


```
IIIIII  NN  NN  PPPPPPP  UU  UU  TTTTTTTTTT  TTTTTTTTTT  XX  XX  TTTTTTTTTT
IIIIII  NN  NN  PPPPPPP  UU  UU  TTTTTTTTTT  TTTTTTTTTT  XX  XX  TTTTTTTTTT
II  NN  NN  PP  PP  UU  UU  TT  TT  XX  XX  TT
II  NN  NN  PP  PP  UU  UU  TT  TT  XX  XX  TT
II  NNNN  NN  PP  PP  UU  UU  TT  TT  XX  XX  TT
II  NNNN  NN  PP  PP  UU  UU  TT  TT  XX  XX  TT
II  NN  NN  PPPPPPP  UU  UU  TT  TT  XX  XX  TT
II  NN  NN  PPPPPPP  UU  UU  TT  TT  XX  XX  TT
II  NN  NNNN  PP  UU  UU  TT  TT  XX  XX  TT
II  NN  NNNN  PP  UU  UU  TT  TT  XX  XX  TT
II  NN  NN  PP  UU  UU  TT  TT  XX  XX  TT
IIIIII  NN  NN  PP  UU  UU  TT  TT  XX  XX  TT
IIIIII  NN  NN  PP  UU  UU  TT  TT  XX  XX  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 MODULE lib_inputtxt ( ! Get next TEXT input line
2 0002 0 LANGUAGE (BLISS32),
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: Library command processor
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 The VAX/VMS librarian is invoked by DCL to process the LIBRARY
38 0038 1 command. It utilizes the librarian procedure set to perform
39 0039 1 the actual modifications to the library.
40 0040 1
41 0041 1 ENVIRONMENT:
42 0042 1
43 0043 1 VAX native, user mode.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Marty Jack, Benn Schreiber CREATION DATE: 23-Aug-1979
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V02-007 RPG0047 Bob Grosso 17-Nov-1981
53 0053 1 Fix entering of empty text modules to not pad other
54 0054 1 modules with blank line.
55 0055 1
56 0056 1 V02-006 RPG0046 Bob Grosso 7-Aug-1981
57 0057 1 lib$gl_ctlmsk now a quadword
```

LIB INPUTTXT
V04=000

H 15
16-Sep-1984 02:00:06
14-Sep-1984 12:38:05

VAX-11 Bliss-32 V4.0-742
[LIBRAR.SRC]INPUTTXT.B32;1

Page 2
(1)

58 0058 1
59 0059 1
60 0060 1
61 0061 1
62 0062 1
63 0063 1
64 0064 1
65 0065 1
66 0066 1
67 0067 1
68 0068 1
69 0069 1
70 0070 1
71 0071 1
72 0072 1

V02-005 RPG0037 Bob Grosso 24-Jun-1981
Permit <> on module name directory specification.
V02-004 RPG0036 Bob Grosso 18-Jun-1981
Permit entering of empty text modules.
V02-003 RPG0035 Bob Grosso 22-Apr-1981
Record module names for Library Update History
V02-002 BLS0029 Benn Schreiber 23-Dec-1980
Convert messages to message compiler


```
: 74      0073 1 LIBRARY
: 75      0074 1
: 76      0075 1 REQUIRE 'SYSS$LIBRARY:STARLET.L32';
: 77      0076 1
: 78      0260 1 REQUIRE 'PREFIX';
: 79      0261 1
: 80      0549 1 REQUIRE 'LIBDEF';
: 81      0550 1
: 82      1141 1
: 83      1142 1 EXTERNAL ROUTINE
: 84      1143 1     get_record,           !Read next input record
: 85      1144 1     lib_log_op,         !Log insert operation
: 86      1145 1     lib_log_upd,        !Record module names for LUH
: 87      1146 1     lbr$put_record : ADDRESSING_MODE (GENERAL), !Write record to library
: 88      1147 1     lbr$replace_key : ADDRESSING_MODE (GENERAL), !Replace or insert key
: 89      1148 1     lbr$lookup_key : ADDRESSING_MODE (GENERAL), !Lookup key
: 90      1149 1     lbr$insert_key : ADDRESSING_MODE (GENERAL), !Insert key
: 91      1150 1     lbr$delete_key : ADDRESSING_MODE (GENERAL), !Delete key
: 92      1151 1     lbr$delete_data : ADDRESSING_MODE (GENERAL), !Delete data
: 93      1152 1     lbr$put_end : ADDRESSING_MODE (GENERAL); !Finish writing to library
: 94      1153 1
: 95      1154 1 EXTERNAL
: 96      1155 1     lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL), !RMS STV from librarian
: 97      1156 1     lib$gl_keysize,      !Max length of keys
: 98      1157 1     lib$gl_ctlmsk : BLOCK [2],
: 99      1158 1     lib$gl_libfdb : REF BBLOCK,
100      1159 1     lib$gl_inpfdb : REF BBLOCK,
101      1160 1     lib$gl_libctl;
102      1161 1
103      1162 1 EXTERNAL LITERAL
104      1163 1     lib$_inserterr,       !Error inserting
105      1164 1     lib$_deldataerr,     !Delete data error
106      1165 1     lib$_replaced,       !Module replaced
107      1166 1     lib$_inserted,       !Module inserted
108      1167 1     lib$_nomodnam,       !No module name
109      1168 1     lib$_modnamlng,      !Module name length bad
110      1169 1     lib$_dupmod;         !Duplicate module
111      1170 1
112      1171 1 FORWARD ROUTINE
113      1172 1     setmodname;
114      1173 1
115      1174 1 OWN
116      1175 1     bufdesc : BBLOCK [dsc$c_s_bln],           ! Descriptor for current line
117      1176 1     txtrfa : BBLOCK [rfa$c_length],
118      1177 1     modnamdesc : BBLOCK [dsc$c_s_bln];        ! Descriptor for module name
119      1178 1
120      1179 1 BIND
121      1180 1     linelen = bufdesc [dsc$w_length] : WORD,
122      1181 1     lineaddr = bufdesc [dsc$a_pointer];
```

```
124 1182 1 GLOBAL ROUTINE lib_input_txt =
125 1183 2 BEGIN
126 1184 2
127 1185 2 This routine reads TEXT source files and inserts them into the library.
128 1186 2
129 1187 2 LOCAL
130 1188 2   deltxtrfa : BBLOCK [rfa$length],
131 1189 2   replacing,
132 1190 2   status,
133 1191 2   get_status;
134 1192 2
135 1193 2 BIND
136 1194 2   libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK,
137 1195 2   inpdesc = lib$gl_inpfdb [fdb$l_namdesc] : BBLOCK;
138 1196 2
139 1197 2 perform (setmodname ());
140 1198 2 txtrfa = 0;
141 1199 2
142 1200 2 Get first record. If it's blank, then call lbr$put_record to
143 1201 2 put a module header. Otherwise loop reading whole input file
144 1202 2 until end of file.
145 1203 2
146 1204 2 get_status = get_record (bufdesc);
147 1205 2 IF NOT (status = lbr$put_record (lib$gl_libctl, bufdesc, txtrfa))
148 1206 2 THEN SIGNAL (lib$writeerr, 1, libdesc, status, lbr$gl_rmsstv);
149 1207 2
150 1208 2 IF .get_status NEQ rms$_eof
151 1209 2 THEN
152 1210 2   BEGIN
153 1211 2     WHILE (get_status = get_record (bufdesc)) NEQ rms$_eof DO
154 1212 2       IF NOT (status = lbr$put_record (lib$gl_libctl, bufdesc, txtrfa))
155 1213 2       THEN SIGNAL (lib$writeerr, 1, libdesc, status, lbr$gl_rmsstv);
156 1214 2     END;
157 1215 2
158 1216 2 Write end of module record
159 1217 2
160 1218 2 IF NOT (status = lbr$put_end (lib$gl_libctl))
161 1219 2 THEN SIGNAL (lib$writeerr, 1, libdesc, status, lbr$gl_rmsstv);
162 1220 2
163 1221 2
164 1222 2 If replacing, determine if module already exists in library. If it does,
165 1223 2 then replace the module and delete the old data. If it doesn't just
166 1224 2 insert the module by calling lbr$replace_key. If inserting the module,
167 1225 2 then call lbr$insert_key.
168 1226 2
169 1227 2
170 1228 2 replacing = false;
171 1229 2 IF .lib$gl_ctlmsk [lib$v_replace]
172 1230 2 THEN BEGIN
173 1231 2   replacing = lbr$lookup_key (lib$gl_libctl, modnamdesc, deltxtrfa);
174 1232 2   status = lbr$replace_key (lib$gl_libctl, modnamdesc, deltxtrfa, txtrfa);
175 1233 2   IF NOT .status
176 1234 2   THEN BEGIN
177 1235 2     SIGNAL (lib$inserterr, 2, modnamdesc, libdesc, status, lbr$gl_rmsstv);
178 1236 2     RETURN lib$inserterr;
179 1237 2   END;
180 1238 2   IF .replacing
```


LIB_INPUTTXT
V04=000

K 15
16-Sep-1984 02:00:06 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:38:05 [LIBRAR.SRC]INPUTTXT.B32;1

Page 5
(3)

```

181      1239  4      THEN IF NOT (status = lbr$delete_data (lib$gl_libctl, deltxtrfa))
182      1240  4      THEN BEGIN
183      1241  4      SIGNAL (lib$_delaterr, 1, libdesc, .status, .lbr$gl_rmsstv);
184      1242  4      RETURN lib$_delaterr;
185      1243  3      END;
186      1244  3      END
187      1245  3      ELSE BEGIN
188      1246  3      status = lbr$insert_key (lib$gl_libctl, modnamdesc, txtrfa);
189      1247  3      IF NOT .status
190      1248  3      THEN BEGIN
191      1249  4      SIGNAL (lib$_inserterr, 2, modnamdesc, libdesc, .status, .lbr$gl_rmsstv);
192      1250  4      RETURN lib$_inserterr;
193      1251  4      END;
194      1252  3      END;
195      1253  2      lib_log_upd ((IF .replacing THEN lbr$_replaced
196      1254  2      ELSE lbr$_inserted), modnamdesc );
197      1255  3      lib_log_op ((IF .replacing THEN lib$_replaced
198      1256  3      ELSE lib$_inserted), modnamdesc, .lib$gl_libfdb);
199      1257  3      RETURN true
200      1258  2      !Of lib_input_txt
201      1259  2      END;
202      1260  1
```

.TITLE LIB_INPUTTXT
.IDENT \V04-000\

.PSECT \$OWNS\$,NOEXE,2

00000 BUFDESC: .BLKB 8
00008 TXTRFA: .BLKB 6
0000E .BLKB 2
00010 MODNAMDESC:
.BLKB 8

LINELEN= BUFDESC
LINEADDR= BUFDESC+4
.EXTRN GET_RECORD, LIB_LOG_OP
.EXTRN LIB_LOG_UPD, LBR\$PUT_RECORD
.EXTRN LBR\$REPLACE_KEY
.EXTRN LBR\$LOOKUP_KEY, LBR\$INSERT_KEY
.EXTRN LBR\$DELETE_KEY, LBR\$DELETE_DATA
.EXTRN LBR\$PUT_END, LBR\$GL_RMSSTV
.EXTRN LIB\$GL_REYSIZE, LIB\$GL_CTLMSK
.EXTRN LIB\$GL_LIBFDB, LIB\$GL_INPFDB
.EXTRN LIB\$GL_LIBCTL, LIB\$INSERTERR
.EXTRN LIB\$_DELATERR, LIB\$_REPLACED
.EXTRN LIB\$_INSERTED, LIB\$_MODNAM
.EXTRN LIB\$_MODNAMLNG, LIB\$_DUPMOD

.PSECT \$CODE\$,NOWRT,2

OFFC 00000

.ENTRY LIB_INPUT_TXT, Save R2,R3,R4,R5,R6,R7,R8,- ; 1182
R9,R10,R11
MOVL #LIB\$_INSERTERR, R11
MOVL #LIB\$_DELATERR, R10
MOVAB LBR\$PUT_RECORD, R9

5B 00000000G 8F DO 00002
5A 00000000G 8F DO 00009
59 00000000G 00 9E 00010

53	0000G	58	0000G	CF	9E	00017	MOVAB	LIB\$GL_LIBCTL, R8	
	0000V	57	00000000G	00	9E	0001C	MOVAB	LIB\$SIGNAL, R7	
		56	00000000G	00	9E	00023	MOVAB	LBR\$GL_RMSSTV, R6	
		55	0000'	CF	9E	0002A	MOVAB	MODNAMEDESC, R5	
		5E		08	C2	0002F	SUBL2	#8, SP	
		CF		10	C1	00032	ADDL3	#16, LIB\$GL_LIBFDB, R3	1194
		CF		00	FB	00038	CALLS	#0, SETMODNAME	1197
		01		50	E8	0003D	BLBS	STATUS, 1\$	
					04	00040	RET		
			F8	A5	D4	00041	CLRL	TXTRFA	1198
			F0	A5	9F	00044	PUSHAB	BUFDESC	1204
	0000G	CF		01	FB	00047	CALLS	#1, GET_RECORD	
		54		50	D0	0004C	MOVL	R0, GET_STATUS	
			F8	A5	9F	0004F	PUSHAB	TXTRFA	1205
			F0	A5	9F	00052	PUSHAB	BUFDESC	
		69		58	DD	00055	PUSHL	R8	
		52		03	FB	00057	CALLS	#3, LBR\$PUT_RECORD	
		11		50	D0	0005A	MOVL	R0, STATUS	
				52	E8	0005D	BLBS	STATUS, 2\$	
				66	DD	00060	PUSHL	LBR\$GL_RMSSTV	1206
				52	DD	00062	PUSHL	STATUS	
				53	DD	00064	PUSHL	R3	
				01	DD	00066	PUSHL	#1	
			008610D2	8F	DD	00068	PUSHL	#8786130	
	0001827A	67		05	FB	0006E	CALLS	#5, LIB\$SIGNAL	
		8F		54	D1	00071	CMPL	GET_STATUS, #98938	1208
				38	13	00078	BEQL	4\$	
			F0	A5	9F	0007A	PUSHAB	BUFDESC	1211
				01	FB	0007D	CALLS	#1, GET_RECORD	
	0000G	CF		50	D0	00082	MOVL	R0, GET_STATUS	
		54		54	D1	00085	CMPL	GET_STATUS, #98938	
	0001827A	8F		24	13	0008C	BEQL	4\$	
			F8	A5	9F	0008E	PUSHAB	TXTRFA	1212
			F0	A5	9F	00091	PUSHAB	BUFDESC	
		69		58	DD	00094	PUSHL	R8	
		52		03	FB	00096	CALLS	#3, LBR\$PUT_RECORD	
		DB		50	D0	00099	MOVL	R0, STATUS	
				52	E8	0009C	BLBS	STATUS, 3\$	
				66	DD	0009F	PUSHL	LBR\$GL_RMSSTV	1213
				52	DD	000A1	PUSHL	STATUS	
				53	DD	000A3	PUSHL	R3	
				01	DD	000A5	PUSHL	#1	
			008610D2	8F	DD	000A7	PUSHL	#8786130	
		67		05	FB	000AD	CALLS	#5, LIB\$SIGNAL	
				C8	11	000B0	BRB	3\$	1212
				58	DD	000B2	PUSHL	R8	1219
	00000000G	00		01	FB	000B4	CALLS	#1, LBR\$PUT_END	
		52		50	D0	000BB	MOVL	R0, STATUS	
		11		52	E8	000BE	BLBS	STATUS, 5\$	
				66	DD	000C1	PUSHL	LBR\$GL_RMSSTV	1220
				52	DD	000C3	PUSHL	STATUS	
				53	DD	000C5	PUSHL	R3	
				01	DD	000C7	PUSHL	#1	
			008610D2	8F	DD	000C9	PUSHL	#8786130	
		67		05	FB	000CF	CALLS	#5, LIB\$SIGNAL	
				54	D4	000D2	CLRL	REPLACING	1228
4C	0000G	CF		05	E1	000D4	BBC	#5, LIB\$GL_CTLMSK+1, 6\$	1229

		4020	8F BB 000DA	PUSHR	#*M<R5,SP>	1231
			58 DD 000DE	PUSHL	R8	
00000000G	00		03 FB 000E0	CALLS	#3, LBR\$LOOKUP_KEY	
	54		50 D0 000E7	MOVL	R0, REPLACING	
		F8	A5 9F 000EA	PUSHAB	TXTRFA	1232
		04	AE 9F 000ED	PUSHAB	DELTXTFA	
			55 DD 000F0	PUSHL	R5	
			58 DD 000F2	PUSHL	R8	
00000000G	00		04 FB 000F4	CALLS	#4, LBR\$REPLACE_KEY	
	52		50 D0 000FB	MOVL	R0, STATUS	
	39		52 E9 000FE	BLBC	STATUS, 7\$	1233
	49		54 E9 00101	BLBC	REPLACING, 8\$	1238
		4100	8F BB 00104	PUSHR	#*M<R8,SP>	1239
00000000G	00		02 FB 00108	CALLS	#2, LBR\$DELETE_DATA	
	52		50 D0 0010F	MOVL	R0, STATUS	
	38		52 E8 00112	BLBS	STATUS, 8\$	
			66 DD 00115	PUSHL	LBR\$GL_RMSSTV	1241
			52 DD 00117	PUSHL	STATUS	
			53 DD 00119	PUSHL	R3	
			01 DD 0011B	PUSHL	#1	
			5A DD 0011D	PUSHL	R10	
	67		05 FB 0011F	CALLS	#5, LIB\$SIGNAL	
	50		5A D0 00122	MOVL	R10, R0	1242
			04 00125	RET		
		F8	A5 9F 00126 6\$:	PUSHAB	TXTRFA	1247
			55 DD 00129	PUSHL	R5	
			58 DD 0012B	PUSHL	R8	
00000000G	00		03 FB 0012D	CALLS	#3, LBR\$INSERT_KEY	
	52		50 D0 00134	MOVL	R0, STATUS	
	13		52 E8 00137	BLBS	STATUS, 8\$	1248
			66 DD 0013A 7\$:	PUSHL	LBR\$GL_RMSSTV	1250
			52 DD 0013C	PUSHL	STATUS	
			53 DD 0013E	PUSHL	R3	
			55 DD 00140	PUSHL	R5	
			02 DD 00142	PUSHL	#2	
			5B DD 00144	PUSHL	R11	
	67		06 FB 00146	CALLS	#6, LIB\$SIGNAL	
	50		5B D0 00149	MOVL	R11, R0	1251
			04 0014C	RET		
			55 DD 0014D 8\$:	PUSHL	R5	1255
	04		54 E9 0014F	BLBC	REPLACING, 9\$	
			03 DD 00152	PUSHL	#3	
			02 11 00154	BRB	10\$	
			02 DD 00156 9\$:	PUSHL	#2	
0000G	CF		02 FB 00158 10\$:	CALLS	#2, LIB_LOG_UPD	
		0000G	CF DD 0015D	PUSHL	LIB\$GL_CIBFDB	1258
			55 DD 00161	PUSHL	R5	1257
	08		54 E9 00163	BLBC	REPLACING, 11\$	
		00000000G	8F DD 00166	PUSHL	#LIB\$_REPLACED	
			06 11 0016C	BRB	12\$	
		00000000G	8F DD 0016E 11\$:	PUSHL	#LIB\$_INSERTED	
0000G	CF		03 FB 00174 12\$:	CALLS	#3, LIB_LOG_OP	
	50		01 D0 00179	MOVL	#1, R0	1259
			04 0017C	RET		1260

; Routine Size: 381 bytes, Routine Base: \$CODE\$ + 0000

LIB INPUTTXT
V04=000

N 15
16-Sep-1984 02:00:06
14-Sep-1984 12:38:05

```
VAX-11 Bliss-32 V4.0-742
[LIBRAR.SRC]INPUTTXT.B32;1
```

Page 8
(3)

B
C
D
E
F
G
H
I
J
K
L
M
N
B
C
D
E
F
G
H
I
J
K
L
M
N
B
C
D
E
F
G
H
I
J
K
L
M
N
B
C
D
E
F
G
H
I


```
1261 1 ROUTINE setmodname =
1262 2 BEGIN
1263 2
1264 2 This routine extracts the file name of the current input file
1265 2 and saves it for entering the module name into the library.
1266 2
1267 2 BIND
1268 2     modulename = lib$gl_inpfdb [fdb$l_modnam] : BBLOCK,
1269 2     libdesc = lib$gl_lifdb [fdb$l_namdesc] : BBLOCK,
1270 2     inpdesc = lib$gl_inpfdb [fdb$l_namdesc] : BBLOCK,
1271 2     inpstring = .inpdesc [dsc$a_pointer] : VECTOR [,BYTE];
1272 2 LOCAL
1273 2     modascic : VECTOR [lbr$c_pagesize, BYTE],
1274 2     j, k;
1275 2
1276 2 IF .modulename [dsc$w_length] NEQ 0                                !Specified by command
1277 2 THEN BEGIN
1278 2     modnamdesc [dsc$w_length] = .modulename [dsc$w_length];
1279 2     modnamdesc [dsc$a_pointer] = .modulename [dsc$a_pointer];
1280 2 END
1281 2 ELSE BEGIN
1282 2     j = k = -1;
1283 2     DECR i FROM .inpdesc [dsc$w_length]-1 TO 0 DO
1284 2     BEGIN
1285 2         IF .inpstring [.i] EQL %C'.' THEN EXITLOOP j = .i;
1286 2     END;
1287 2     DECR i FROM .j-1 TO 0 DO
1288 2     BEGIN
1289 2         IF (.inpstring [.i] EQL %C']') OR (.inpstring [.i] EQL %C'>')
1290 2         THEN EXITLOOP k = .i + 1;
1291 2     END;
1292 2
1293 2     modnamdesc [dsc$w_length] = .j - .k;
1294 2     modnamdesc [dsc$a_pointer] = .inpstring [.k];
1295 2 END;
1296 2
1297 2 modascic [0] = .modnamdesc [dsc$w_length];
1298 2 CH$MOVE (.modascic [0], .modnamdesc [dsc$a_pointer], modascic [1]);
1299 2 IF .modnamdesc [dsc$w_length] EQL 0
1300 2 THEN BEGIN
1301 2     SIGNAL (lib$_nomodnam, 1, inpdesc);
1302 2     RETURN lib$_nomodnam;
1303 2 END
1304 2 ELSE IF .modnamdesc [dsc$w_length] GTRU .lib$gl_keysize
1305 2 THEN BEGIN
1306 2     SIGNAL (lib$_modnamlng, 3, modascic, .modnamdesc [dsc$w_length], inpdesc);
1307 2     RETURN lib$_modnamlng;
1308 2 END;
1309 2 IF NOT .lib$gl_ctlmsk [lib$v_replace]
1310 2 AND lbr$lookup_key (lib$gl_libctl, modnamdesc, txtrfa)
1311 2 THEN BEGIN
1312 2     SIGNAL (lib$_dupmod, 3, modascic, inpdesc, libdesc);
1313 2     RETURN false;
1314 2 END;
1315 2
1316 2 RETURN true
1317 1 END;
```



```
OFFC 00000 SETMODNAME:
      5B 00000000G 8F D0 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 1261
      5A 00000000G 00 9E 00009 MOVL #LIB$NOMODNAM, R11 :
      59 0000' CF 9E 00010 MOVAB LIB$SIGNAL, R10 :
      5E FE00 CE 9E 00015 MOVAB MODNAMDESC, R9 :
50 0000G CF 18 C1 0001A ADDL3 #24, LIB$GL_INPFDB, R0 : 1268
58 0000G CF 10 C1 00020 ADDL3 #16, LIB$GL_LIBFDB, R8 : 1269
56 0000G CF 10 C1 00026 ADDL3 #16, LIB$GL_INPFDB, R6 : 1270
      60 B5 0002C TSTW (R0) : 1276
      0A 13 0002E BEQL 1$ :
      69 60 B0 00030 MOVW (R0), MODNAMDESC : 1278
      04 A9 04 A0 D0 00033 MOVL 4(R0), MODNAMDESC+4 : 1279
      41 11 00038 BRB 9$ : 1276
      51 01 CE 0003A 1$: MNEGL #1, K : 1282
      52 01 CE 0003D MNEGL #1, J :
      50 66 3C 00040 MOVZWL (R6), I : 1283
      0C 11 00043 BRB 3$ :
      2E 04 B640 91 00045 2$: CMPB @4(R6)[I], #46 : 1285
      05 12 0004A BNEQ 3$ :
      52 50 D0 0004C MOVL I, J :
      03 11 0004F BRB 4$ :
      F1 50 F4 00051 3$: SOBGEQ I, 2$ : 1283
      50 52 D0 00054 4$: MOVL J, I : 1287
      15 11 00057 BRB 7$ :
      5D 8F 04 B640 91 00059 5$: CMPB @4(R6)[I], #93 : 1289
      07 13 0005F BEQL 6$ :
      3E 04 B640 91 00061 CMPB @4(R6)[I], #62 :
      06 12 00066 BNEQ 7$ :
      51 01 A0 9E 00068 6$: MOVAB 1(R0), K : 1290
      03 11 0006C BRB 8$ :
      E8 50 F4 0006E 7$: SOBGEQ I, 5$ : 1287
      69 52 51 A3 00071 8$: SUBW3 K, J, MODNAMDESC : 1293
      04 A9 04 B641 9E 00075 MOVAB @4(R6)[K], MODNAMDESC+4 : 1294
      57 69 3C 0007B 9$: MOVZWL MODNAMDESC, R7 : 1297
      6E 57 90 0007E MOVB R7, MODASCIC :
      50 6E 9A 00081 MOVZBL MODASCIC, R0 : 1298
01 AE 04 B9 50 28 00084 MOVC3 R0, @MODNAMDESC+4, MODASCIC+1 :
      57 D5 0008A TSTL R7 : 1299
      0D 12 0008C BNEQ 10$ :
      56 DD 0008E PUSHL R6 : 1301
      01 DD 00090 PUSHL #1 :
      5B DD 00092 PUSHL R11 :
      6A 03 FB 00094 CALLS #3, LIB$SIGNAL :
      50 5B D0 00097 MOVL R11, R0 : 1302
      04 0009A RET :
      0000G CF 57 D1 0009B 10$: CMPL R7, LIB$GL_KEYSIZ : 1304
      1A 1B 000A0 BLEQU 11$ :
      56 DD 000A2 PUSHL R6 : 1306
      57 DD 000A4 PUSHL R7 :
      08 AE 9F 000A6 PUSHAB MODASCIC :
      03 DD 000A9 PUSHL #3 :
```


		00000000G	8F	DD	000AB	PUSHL	#LIB\$ MODNAMLNG	
	6A		05	FB	000B1	CALLS	#5, LIB\$ SIGNAL	
	50	00000000G	8F	DD	000B4	MOVL	#LIB\$ MODNAMLNG, R0	1307
				04	000BB	RET		
27	0000G	CF	05	E0	000BC	BBS	#5, LIB\$ GL_CTLMSK+1, 12\$	1309
		F8	A9	9F	000C2	PUSHAB	TXTRFA	1310
			59	DD	000C5	PUSHL	R9	
		0000G	CF	9F	000C7	PUSHAB	LIB\$ GL LIBCTL	
	00000000G	00	03	FB	000CB	CALLS	#3, LBR\$ LOOKUP_KEY	
		14	50	E9	000D2	BLBC	R0, 12\$	
		0140	8F	BB	000D5	PUSHR	#M<R6,R8>	1312
		08	AE	9F	000D9	PUSHAB	MODASCIC	
			03	DD	000DC	PUSHL	#3	
		00000000G	8F	DD	000DE	PUSHL	#LIB\$ DUPMOD	
	6A		05	FB	000E4	CALLS	#5, LIB\$ SIGNAL	
			04	11	000E7	BRB	13\$	1313
	50		01	D0	000E9	MOVL	#1, R0	1316
			04	000EC	RET			
			50	D4	000ED	CLRL	R0	1317
			04	000EF	RET			

; Routine Size: 240 bytes, Routine Base: \$CODE\$ + 017D

:	261	1318	1
:	262	1319	1 END
:	263	1320	0 ELUDOM

.EXTRN LIB\$ SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	24	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	621	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	25	0	581	00:01.0

COMMAND QUALIFIERS

LIB_INPUTTXT
V04=000

E 16
16-Sep-1984 02:00:06
14-Sep-1984 12:38:05

VAX-11 Bliss-32 V4.0-742
[LIBRAR.SRC]INPUTTXT.B32;1

Page 12
(4)

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

; Size: 621 code + 24 data bytes
; Run Time: 00:20.5
; Elapsed Time: 00:43.5
; Lines/CPU Min: 3861
; Lexemes/CPU-Min: 43161
; Memory Used: 200 pages
; Compilation Complete

0201 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY