

LII

EEEEEEEEEE	XX	XX	TTTTTTTTTT	RRRRRRRR	AAAAAA	CCCCCCCC	TTTTTTTTTT	
EEEEEEEEEE	XX	XX	TTTTTTTTTT	RRRRRRRR	AAAAAA	CCCCCCCC	TTTTTTTTTT	
EE	XX	XX	TT	RR	RR	AA	CC	TT
EE	XX	XX	TT	RR	RR	AA	CC	TT
EE			TT	RR	RR	AA	CC	TT
EE	XX	XX	TT	RR	RR	AA	CC	TT
EEEEEEEEEE		XX	TT	RRRRRRRR		AA	CC	TT
EEEEEEEEEE		XX	TT	RRRRRRRR		AA	CC	TT
EE		XX	TT	RR	RR	AAAAAAAAAA	CC	TT
EE		XX	TT	RR	RR	AAAAAAAAAA	CC	TT
EE			TT	RR	RR	AA	CC	TT
EE	XX	XX	TT	RR	RR	AA	CC	TT
EEEEEEEEEEEE	XX	XX	TT	RR	RR	AA	CC	TT
EEEEEEEEEEEE	XX	XX	TT	RR	RR	AA	CCCCCCCC	TT
							CCCCCCCC	

```

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_extract ( ! Extract modules from library
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: Benn Schreiber, CREATION DATE: 26-June-1979
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V02-005 RPG0035 Bob Grosso 03-Mar-1982
53 0053 1 Close /OUT file if nothing extracted and continue
54 0054 1 extracting if one of a list of modules to be extracted
55 0055 1 is not found.
56 0056 1
57 0057 1 V02-004 RPG0034 Bob Grosso 7-Aug-1981
```


LIB_EXTRACT
V04=000

B 16
16-Sep-1984 01:51:18
14-Sep-1984 12:38:01

VAX-11 BLISS-32 V4.0-742
[LIBRAR.SRC]EXTRACT.B32;1

Page 2
(1)

: 58
: 59
: 60
: 61
: 62
: 63
: 64
: 65
: 66
: 67
: 68
: 69

0058 1
0059 1
0060 1
0061 1
0062 1
0063 1
0064 1
0065 1
0066 1
0067 1
0068 1
0069 1

! !
! !
! !
! !
! !
! !
! !
! !
! !
! !
! !
! !

lib\$gl_ctlmsk now a quadword

V02-003 BLS0029 Benn Schreiber 23-Dec-1980
Convert messages to message compiler

V02-002 RPG0003 Bob Grosso 3-Sep-1980
Exit read loop and continue extracting modules
if read error encountered during extract.


```

71 0070 1 LIBRARY
72 0071 1 'SYSS$LIBRARY:STARLET.L32';
73 0072 1 REQUIRE
74 0073 1 'PREFIX';
75 0257 1 REQUIRE
76 0258 1 'LIBDEF';
77 0546 1 REQUIRE
78 0547 1 'LBRDEF';
79 1138 1
80 1139 1 EXTERNAL
81 1140 1   lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL), !RMS STV from librarian I/O ops
82 1141 1   lib$gl_modxtrl : VECTOR [2], !Listhead for modules to extract
83 1142 1   lib$gl_ctlmsk : BLOCK [2], !Librarian control mask
84 1143 1   lib$gl_libctl, !Library control index
85 1144 1   lib$gl_type, !Type of library
86 1145 1   lib$gl_modnamix, !Index number for mod name index
87 1146 1   lib$al_rab : BBLOCK, !RAB
88 1147 1   lib$gl_libfdb : REF BBLOCK, !Pointer to library FDB
89 1148 1   lib$gl_outfdb : REF BBLOCK; !Output file FDB
90 1149 1
91 1150 1 EXTERNAL ROUTINE
92 1151 1   lbr$find : ADDRESSING_MODE (GENERAL), !Find known text records
93 1152 1   lbr$lookup_key : ADDRESSING_MODE (GENERAL), !Lookup key in library
94 1153 1   lbr$get_record : ADDRESSING_MODE (GENERAL), !Read record from library
95 1154 1   lbr$get_index : ADDRESSING_MODE (GENERAL), !Return all elements of an index
96 1155 1   lib_open_out, !Open output file
97 1156 1   lib_close_out, !Close output file
98 1157 1   lib_log_op, !Log extraction
99 1158 1   lib_get_zmem, !Allocate zeroed virtual memory
100 1159 1   lib_get_mem, !Allocate virtual memory
101 1160 1   lib_free_mem; !Deallocate virtual memory
102 1161 1
103 1162 1 EXTERNAL LITERAL
104 1163 1   lbr$_nomtchfou, !No match found
105 1164 1   lib$_romtchfou, !No match found
106 1165 1   lib$_indexerr, !Index error
107 1166 1   lib$_lookuperr, !Lookup error in library
108 1167 1   lib$_extracted; !Successfully extracted module
109 1168 1
110 1169 1 FORWARD ROUTINE
111 1170 1   extractmodule; !Extract one module
112 1171 1
113 1172 1 OWN
114 1173 1   extracted1; !Flag true if a module was extracted
```



```
116 1174 1 GLOBAL ROUTINE lib_extrct_mods =
117 1175 2 BEGIN
118 1176 2 ++
119 1177 2 Extract modules from the library
120 1178 2
121 1179 2 Inputs:
122 1180 2
123 1181 2 NONE
124 1182 2
125 1183 2 Implicit inputs:
126 1184 2
127 1185 2 lib$gl_modxtrl is the queue listhead containing the names of the
128 1186 2 modules to extract.
129 1187 2
130 1188 2 Outputs:
131 1189 2
132 1190 2 the specified modules are extracted and written to the output file.
133 1191 2
134 1192 2 --
135 1193 2
136 1194 2 LOCAL
137 1195 2 nmbk : REF BBLOCK,
138 1196 2 keydesc : BBLOCK [dsc$c_s_bln]; !Descriptor for key returned
139 1197 2
140 1198 2 BIND
141 1199 2 outdesc = lib$gl_outfdb [fdb$l_namdesc] : BBLOCK,
142 1200 2 libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK;
143 1201 2
144 1202 2 BUILTIN
145 1203 2 REMQUE;
146 1204 2
147 1205 2 Open the output file
148 1206 2
149 1207 2 extracted1 = false;
150 P 1208 2 perform (lib_open_out (.lib$gl_outfdb, !Open the file
151 P 1209 2 lib$gl_libfdb [fdb$l_nam], using the library NAM block as the related NAM blo
152 P 1210 2 (IF .lib$gl_type EQL lbr$c_typ_mlb with carriage control
153 P 1211 2 OR .lib$gl_type EQL lbr$c_typ_hlp if macro or help
154 P 1212 2 OR .lib$gl_type EQL lbr$c_typ_txt or text library
155 P 1213 2 THEN true
156 1214 2 ELSE false));
157 1215 2
158 1216 2 Now lookup each module on the extract list, and if found
159 1217 2 copy it to the output file. Issue warning message if
160 1218 2 module is not found
161 1219 2
162 1220 2 WHILE NOT REMQUE (.lib$gl_modxtrl, nmbk) !Get next one off the list
163 1221 2 DO BEGIN
164 1222 2 LOCAL
165 1223 2 status;
166 1224 2 keydesc [dsc$w_length] = .nmbk [lnb$b_namlng]; !Set descriptor for module name
167 1225 2 keydesc [dsc$a_pointer] = nmbk [lnb$t_name];
168 1226 2 IF NOT CH$FAIL (CH$FIND_CH (.keydesc [dsc$w_length], !Check if any wild cards
169 1227 2 .keydesc [dsc$a_pointer], %ASCII '%'))
170 1228 2 OR NOT CH$FAIL (CH$FIND_CH (.keydesc [dsc$w_length],
171 1229 2 .keydesc [dsc$a_pointer], %ASCII '*f'))
172 1230 2 THEN BEGIN
```



```

: 173      1231  5      IF (status = lbr$get_index (lib$gl_libctl, lib$gl_modnamix,
: 174      1232  4      extractmodule, keydesc)) EQL (br$nomtchfou
: 175      1233  4      THEN SIGNAL (lib$nomtchfou, 1, keydesc)
: 176      1234  4      ELSE IF NOT .status
: 177      1235  4      THEN SIGNAL (lib$indexerr, 1,
: 178      1236  4      lib$gl_libfdb [fdb$l_namdesc], .status,
: 179      1237  4      .lbr$gl_rmsstv);
: 180      1238  4      END
: 181      1239  3      ELSE
: 182      1240  3      extractmodule ( keydesc );      !Extract the module
: 183      1241  3
: 184      1242  3      lib_free_mem (lnb$fixedsize+.nmblk [lnb$b_namlng], .nmblk); !Deallocate the module name block
: 185      1243  2      END;      !WHILE loop
: 186      1244  2
: 187      1245  2      ! Close the output file
: 188      1246  2
: 189      1247  2      lib_close_out (.lib$gl_outfdb, NOT .extracted1);
: 190      1248  2
: 191      1249  2      RETURN true
: 192      1250  1      END;      ! Of lib_extrct_mods
```

```

.TITLE LIB_EXTRACT
.IDENT \V04-000\
.PSECT $OWNS,NOEXE,2
```

00000 EXTRACTED1:

```

.BLKB 4
.EXTRN LBR$GL_RMSSTV, LIB$GL_MODXTRL
.EXTRN LIB$GL_CTLMSK, LIB$GL_LIBCTL
.EXTRN LIB$GL_TYPE, LIB$GL_MODNAMIX
.EXTRN LIB$AL_RAB, LIB$GL_LIBFDB
.EXTRN LIB$GL_OUTFDB, LBR$FIND
.EXTRN LBR$LOOKUP_KEY, LBR$GET_RECORD
.EXTRN LBR$GET_INDEX, LIB_OPEN_OUT
.EXTRN LIB_CLOSE_OUT, LIB_LOG_OP
.EXTRN LIB_GET_ZMEM, LIB_GET_MEM
.EXTRN LIB_FREE_MEM, LBR$NOMTCHFOU
.EXTRN LIB$NOMTCHFOU, LIB$INDEXERR
.EXTRN LIB$LOOKUPERR, LIB$EXTRACTED
.PSECT $CODE$,NOWRT,2
```

```

54 00000000G 00 001C 00000
5E 0000' 08 9E 00002
0000' CF D4 0000C
50 0000G CF D0 00010
02 50 D1 00015
0A 13 00018
03 50 D1 0001A
05 13 0001D
04 50 D1 0001F
04 12 00022
01 DD 00024 1$:
```

```

.ENTRY LIB_EXTRCT_MODS, Save R2,R3,R4
MOVAB LIB$SIGNAL, R4
SUBL2 #8, SP
CLRL EXTRACTED1
MOVL LIB$GL_TYPE, R0
CMPL R0, #2
BEQL 1$
CMPL R0, #3
BEQL 1$
CMPL R0, #4
BNEQ 2$
PUSHL #1
```

```

: 1174
:
: 1207
: 1214
:
```


			02	11	00026	BRB	3\$		
			7E	D4	00028	CLRL	-(SP)		
7E	0000G	CF	00000040	8F	C1	ADDL3	#64, LIB\$GL_LIBFDB, -(SP)		
			0000G	CF	DD	PUSHL	LIB\$GL_OUTFDB		
	0000G	CF		03	FB	CALLS	#3, LIB_OPEN_OUT		
		01		50	E8	BLBS	STATUS, 4\$		
				04	00040	RET			
		53	0000G	DF	0F	REMOVE	@LIB\$GL_MODXTRL, NMBLK		1220
				03	1C	BVC	5\$		
				0089	31	BRW	12\$		
		6E	09	A3	9B	MOVZBW	9(NMBLK), KEYDESC		1224
04	BE	04	0A	A3	9E	MOVAB	10(R3), KEYDESC+4		1225
		6E		25	3A	LOCC	#37, KEYDESC, @KEYDESC+4		1226
				02	12	BNEQ	6\$		
				51	D4	CLRL	R1		
				51	D5	TSTL	R1		1227
				0D	12	BNEQ	8\$		
04	BE		6E	2A	3A	LOCC	#42, KEYDESC, @KEYDESC+4		1228
				02	12	BNEQ	7\$		
				51	D4	CLRL	R1		
				51	D5	TSTL	R1		1229
				4E	13	BEQL	10\$		
				5E	DD	PUSHL	SP		1231
			0000V	CF	9F	PUSHAB	EXTRACTMODULE		
			0000G	CF	9F	PUSHAB	LIB\$GL_MODNAMIX		
			0000G	CF	9F	PUSHAB	LIB\$GL_LIBCTL		
				04	FB	CALLS	#4, LBR\$GET_INDEX		
	00000000G	00		50	D0	MOVL	R0, STATUS		
		52		52	D1	CMPL	STATUS, #LBR\$_NOMTCHFOU		1232
	00000000G	8F		0F	12	BNEQ	9\$		
				5E	DD	PUSHL	SP		1233
				01	DD	PUSHL	#1		
			00000000G	8F	DD	PUSHL	#LIB\$_NOMTCHFOU		
				03	FB	CALLS	#3, LIB\$SIGNAL		
			64	25	11	BRB	11\$		
				52	E8	BLBS	STATUS, 11\$		1234
				00	DD	PUSHL	LBR\$GL_RMSSTV		1237
				52	DD	PUSHL	STATUS		1236
7E	0000G	CF		10	C1	ADDL3	#16, LIB\$GL_LIBFDB, -(SP)		
				01	DD	PUSHL	#1		
			00000000G	8F	DD	PUSHL	#LIB\$_INDEXERR		
				05	FB	CALLS	#5, LIB\$SIGNAL		
				07	11	BRB	11\$		1226
				5E	DD	PUSHL	SP		1240
	0000V	CF		01	FB	CALLS	#1, EXTRACTMODULE		
				53	DD	PUSHL	NMBLK		1242
			09	A3	9A	MOVZBL	9(NMBLK), -(SP)		
				0A	C0	ADDL2	#10, (SP)		
	0000G	CF		02	FB	CALLS	#2, LIB_FREE_MEM		
				FF6D	31	BRW	4\$		1220
				CF	D2	MCOML	EXTRACTED1, -(SP)		1247
			0000G	CF	DD	PUSHL	LIB\$GL_OUTFDB		
	0000G	CF		02	FB	CALLS	#2, LIB_CLOSE_OUT		
				50	D0	MOVL	#1, R0		1249
				01	D0	RET			1250
				04	000E5				

; Routine Size: 230 bytes, Routine Base: \$CODE\$ + 0000

LIB_EXTRACT
V04=000

G 16
16-Sep-1984 01:51:18
14-Sep-1984 12:38:01

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

Page 7
(3)


```
194 1251 1 ROUTINE extractmodule (keydesc, modrfa) =
195 1252 2 BEGIN
196 1253 2 ++
197 1254 2 Routine to extract one module from library
198 1255 2
199 1256 2 Inputs:
200 1257 2
201 1258 2 keydesc Address of string descriptor for module to extract
202 1259 2 modrfa (optional) Address of RFA for module
203 1260 2
204 1261 2 Outputs:
205 1262 2
206 1263 2 The module is extracted from the library
207 1264 2
208 1265 2 --
209 1266 2
210 1267 2 MAP
211 1268 2 keydesc : REF BBLOCK;
212 1269 2
213 1270 2 LOCAL
214 1271 2 recordbuffer : BBLOCK [lbr$c_maxrecsiz], !Record buffer
215 1272 2 recdesc : BBLOCK [dsc$c_s_bln], !String descriptor for recordbuffer
216 1273 2 status,
217 1274 2 txtrfa : BBLOCK [rfa$c_length];
218 1275 2
219 1276 2 BUILTIN
220 1277 2 NULLPARAMETER;
221 1278 2
222 1279 2 recdesc [dsc$w_length] = lbr$c_maxrecsiz; !set up buffer string descriptor
223 1280 2 recdesc [dsc$a_pointer] = recordbuffer;
224 1281 2 IF NOT NULLPARAMETER (2)
225 1282 2 THEN status = lbr$find (lib$gl_libctl, .modrfa)
226 1283 2 ELSE status = lbr$lookup_key (lib$gl_libctl, .keydesc, txtrfa);
227 1284 2 IF NOT .status
228 1285 2 THEN BEGIN
229 1286 2 SIGNAL (lib$lookuperr, 2, .keydesc,
230 1287 2 lib$gl_libfdb [fdb$l_namdesc], .status, .lbr$gl_rmsstv);
231 1288 2 RETURN true; !Continue the search
232 1289 2 END
233 1290 2
234 1291 2 Process module after successful lookup/find
235 1292 2
236 1293 2 ELSE WHILE true !found--copy to output file
237 1294 2 DO IF (
238 1295 2 recdesc [dsc$w_length] = lbr$c_maxrecsiz;
239 1296 2 status = lbr$get_record (lib$gl_libctl, recdesc, recdesc) !Read a record
240 1297 2 )
241 1298 2 THEN BEGIN
242 1299 2 lib$al_rab [rab$w_rsz] = .recdesc [dsc$w_length]; !copy length to RAB
243 1300 2 lib$al_rab [rab$l_rbf] = .recdesc [dsc$a_pointer]; !and its address
244 1301 2 status = $PUT (RAB = lib$al_rab); !Write the record
245 1302 2 IF NOT .status
246 1303 2 THEN BEGIN
247 1304 2 SIGNAL (lib$writeerr,
248 1305 2 1, lib$gl_outfdb [fdb$l_namdesc],
249 1306 2 .status, .lib$al_rab [rab$l_stv]);
250 1307 2 EXITLOOP; ! but continue to process other modules
```



```

: 251      1308      4
: 252      1309      3
: 253      1310      3
: 254      1311      3
: 255      1312      3
: 256      1313      3
: 257      1314      3
: 258      1315      3
: 259      1316      3
: 260      1317      3
: 261      1318      3
: 262      1319      3
: 263      1320      3
: 264      1321      2
: 265      1322      2
: 266      1323      1

      END
      ELSE IF .status EQL rms$_eof ! If end of text
      THEN BEGIN ! Then we are done
        lib_log_op (lib$extracted, ! keydesc, ! so log it if logging
          .lib$gl_libfdb);
        extracted1 = true;
        RETURN true;
      END
      ELSE BEGIN
        SIGNAL (lib$readerr, 1, lib$gl_libfdb [fdb$_namdesc],
          .status, .lbr$gl_rmsstv);
        EXITLOOP; ! but continue to process other modules
      END;

      RETURN true
    END;

! Of extractmodule
```

```

                                .EXTRN SYSSPUT

                                007C 00000 EXTRACTMODULE:
56      0000G CF 9E 00002      .WORD      Save R2,R3,R4,R5,R6      : 1251
55      0000G CF 9E 00007      MOVAB      LIB$GL_LIBCTL, R6
54      00000000G 00 9E 0000C      MOVAB      LIB$GL_LIBFDB, R5
53      00000000G 00 9E 00013      MOVAB      LIB$SIGNAL, R4
5E      F7F0 CE 9E 0001A      MOVAB      LBR$GL_RMSSTV, R3
08      AE 0800 8F B0 0001F      MOVW      -2064(SP), SP
OC      AE 10 AE 9E 00025      MOVW      #2048, RECDISC
02      6C 91 0002A      MOVAB      RECDISBUFFER, RECDISC+4
      13 1F 0002D      CMPB      (AP), #2
      08 AC D5 0002F      BLSSU      1$
      0E 13 00032      TSTL      8(AP)
      08 AC DD 00034      BEQL      1$
      56 DD 00037      PUSHL      MODRFA
00000000G 00 02 FB 00039      PUSHL      R6
      0E 11 00040      CALLS     #2, LBR$FIND
      5E DD 00042 1$:      BRB      2$
      04 AC DD 00044      PUSHL      SP
      56 DD 00047      PUSHL      KEYDESC
00000000G 00 03 FB 00049      PUSHL      R6
      52 D0 00050 2$:      CALLS     #3, LBR$LOOKUP_KEY
      18 52 E8 00053      MOVL      R0, STATUS
      63 DD 00056      BLBS      STATUS, 3$
      52 DD 00058      PUSHL      LBR$GL_RMSSTV
7E      65 10 C1 0005A      PUSHL      STATUS
      04 AC DD 0005E      ADDL3     #16, LIB$GL_LIBFDB, -(SP)
      02 DD 00061      PUSHL      KEYDESC
      00000000G 8F DD 00063      PUSHL      #2
      64 06 FB 00069      PUSHL      #LIB$LOOKUPERR
      6C 11 0006C      CALLS     #6, LIB$SIGNAL
      08 AE 0800 8F B0 0006E 3$:      BRB      5$
      08 AE 9F 00074      MOVW      #2048, RECDISC
      OC AE 9F 00077      PUSHAB    RECDISC
      56 DD 0007A      PUSHAB    RECDISC
00000000G 00 03 FB 0007C      PUSHL      R6
      CALLS     #3, LBR$GET_RECORD
      : 1288
      : 1295
      : 1296
      :
```


	52		50	D0	00083	MOVL	R0, STATUS	
	33		52	E9	00086	BLBC	STATUS, 4\$	
	0000G	08	AE	B0	00089	MOVW	RECDISC, LIB\$AL_RAB+34	1299
	0000G	0C	AE	D0	0008F	MOVL	RECDISC+4, LIB\$AL_RAB+40	1300
		0000G	CF	9F	00095	PUSHAB	LIB\$AL_RAB	1301
	00000000G	00	01	FB	00099	CALLS	#1, SY\$PUT	
		52	50	D0	000A0	MOVL	R0, STATUS	
		C8	52	E8	000A3	BLBS	STATUS, 3\$	1302
			CF	DD	000A6	PUSHL	LIB\$AL_RAB+12	1306
		0000G	52	DD	000AA	PUSHL	STATUS	
7E	0000G	CF	10	C1	000AC	ADDL3	#16, LIB\$GL_OUTFDB, -(SP)	1305
			01	DD	000B2	PUSHL	#1	
		008610D2	8F	DD	000B4	PUSHL	#8786130	
			30	11	000BA	BRB	7\$	
	0001827A	8F	52	D1	000BC	CMPL	STATUS, #98938	1310
			17	12	000C3	BNEQ	6\$	
			65	DD	000C5	PUSHL	LIB\$GL_LIBFDB	1313
		04	AC	DD	000C7	PUSHL	KEYDESC	1312
		00000000G	8F	DD	000CA	PUSHL	#LIB\$EXTRACTED	
	0000G	CF	03	FB	000D0	CALLS	#3, LIB LOG OP	
	0000	CF	01	D0	000D5	MOVL	#1, EXTRACTED1	1314
			13	11	000DA	BRB	8\$	1315
			63	DD	000DC	PUSHL	LIB\$GL_RMSSTV	1319
			52	DD	000DE	PUSHL	STATUS	
7E		65	10	C1	000E0	ADDL3	#16, LIB\$GL_LIBFDB, -(SP)	1318
			01	DD	000E4	PUSHL	#1	
		008610B2	8F	DD	000E6	PUSHL	#8786098	
		64	05	FB	000EC	CALLS	#5, LIB\$SIGNAL	
		50	01	D0	000EF	MOVL	#1, R0	1322
			04	000F2	RET			1323

; Routine Size: 243 bytes, Routine Base: \$CODE\$ + 00E6

; 267 1324 1 END ! Of module
; 268 1325 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	4	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	473	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		

LIB_EXTRACT
V04=000

K 16
16-Sep-1984 01:51:18
14-Sep-1984 12:38:01

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

Page 11
(4)

; _\$255\$DUA28:[SYSLIB]STARLET.L32;1

9776

35

0

581

00:01.0

COMMAND QUALIFIERS

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

; Size: 473 code + 4 data bytes
; Run Time: 00:18.6
; Elapsed Time: 00:39.9
; Lines/CPU Min: 4269
; Lexemes/CPU-Min: 46685
; Memory Used: 173 pages
; Compilation Complete

0200 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

TRANSFER
LIS

DATABASE
LIS

PUTCACHE
LIS

LIBRAR

PREFIX
REQ

LIBRARIAN
MAP

CROSS
LIS

SUBS
LIS

PADLBR
LIS

COMPRESS
LIS

LIB
MDL

FILEIO
LIS

EXTRACT
LIS

DELETE
LIS