



```

CCCCCCCC  RRRRRRR  000000  SSSSSSSS  SSSSSSSS
CCCCCCCC  RRRRRRR  000000  SSSSSSSS  SSSSSSSS
CC         RR      RR  00      00  SS         SS
CC         RR      RR  00      00  SS         SS
CC         RR      RR  00      00  SS         SS
CC         RR      RR  00      00  SS         SS
CC         RRRRRRR  00      00  SSSSSS      SSSSSS
CC         RRRRRRR  00      00  SSSSSS      SSSSSS
CC         RR  RR   00      00          SS      SS
CC         RR  RR   00      00          SS      SS
CC         RR  RR   00      00          SS      SS
CC         RR  RR   00      00          SS      SS
CCCCCCCC  RR      RR  000000  SSSSSSSS  SSSSSSSS
CCCCCCCC  RR      RR  000000  SSSSSSSS  SSSSSSSS

```

```

....
....
....
....

```

```

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_cross ( !Cross reference of an object library
2 0002 0
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
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: 09-Oct-1979
50 0050 1
51 0051 1 MODIFIED BY:
52 0052 1
53 0053 1 V03-001 JWT0090 Jim Teague 19-Jan-1983
54 0054 1 Teach Librarian how to cross reference object
55 0055 1 libraries which have the new object records.
56 0056 1 --

```

000

000

000

```
58 0057 1 LIBRARY
59 0058 1 'SYSSLIBRARY:STARLET.L32';           !System macro definitions
60 0059 1 REQUIRE
61 0060 1 'PREFIX';                           !SET OF GENERAL MACROS ETC
62 0244 1 REQUIRE
63 0245 1 'LIBDEF';                           !Librarian structure defs.
64 0533 1 REQUIRE
65 0534 1 'LBRDEF';                           !Library processor defs.
66 1125 1 REQUIRE
67 1126 1 'CRFMDL';                           !Cross ref. internal defines.
68 1195 1
69 1196 1 EXTERNAL
70 1197 1   lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL), !RMS STV from Librarian
71 1198 1   lib$gl_creflags : BLOCK,                !Cross reference indicator flags
72 1199 1   lib$gl_modnamix,                       !Index for module names
73 1200 1   lib$gl_modlist : VECTOR,              !List head for /ONLY
74 1201 1   lib$gl_outfdb : REF BBLOCK,          !Pointer to output FDB
75 1202 1   lib$gl_ctlmsk : BLOCK [1],           !Control flags
76 1203 1   lib$gl_libfdb : REF BBLOCK,          !Pointer to library fdb
77 1204 1   lib$gl_inpfdb : REF BBLOCK,         !Pointer to input file fdb
78 1205 1   lib$gl_rab : BBLOCK,                !RAB for output
79 1206 1   lib$gl_type,                          !Type of library
80 1207 1   lib$gl_libctl;                       !Library control index
81 1208 1
82 1209 1 FORWARD ROUTINE
83 1210 1   lib_crfout,                            !Output one cref line
84 1211 1   newpage,                              !New page for listing
85 1212 1   outputsubhdr,                          !Output cref sub header
86 1213 1   crferror,                              !Error handler for cross reference errors
87 1214 1   cross_one_module,                     !Cross reference one module
88 1215 1   prohdr,                               !Routine to process module headers
89 1216 1   progsd,                               !Routine to process gsd records
90 1217 1   pronul,                               !Routine to process ignored records
91 1218 1   propsectdef,                          !Process p-section definitions
92 1219 1   symbols,                              !Process symbol definitions and references
93 1220 1   entpnts,                              !Process entry point definitions
94 1221 1   procedef,                             !Process procedure declarations
95 1222 1   pro_epmw,
96 1223 1   pro_idc,
97 1224 1   pro_env,
98 1225 1   pro_lsy,
99 1226 1   pro_lepm,
100 1227 1   pro_lpro,
101 1228 1   pro_spsc,
102 1229 1   pro$ymbol;                          !Do all the work of symbol resolution
103 1230 1
104 1231 1 EXTERNAL ROUTINE
105 1232 1   lib_open_out,                          !Open output file
106 1233 1   lib_close_out,                        !Close output file
107 1234 1   lib_get_mem,                          !Allocate virtual memory
108 1235 1   lib_get_zmem,                          !Allocate zeroed virtual memory
109 1236 1   lib_free_mem,                          !and give it back
110 1237 1   lib_log_op,                            !Log operation on console
111 1238 1   lib_log_op,                            !Log operation on console
112 1239 1   SYSSFAO : ADDRESSING_MODE (GENERAL), !Formatted ASCII output
113 1240 1   crf$insrtkey : ADDRESSING_MODE (GENERAL),
114 1241 1   crf$insrtref : ADDRESSING_MODE (GENERAL),
```

```
115 1242 1 crf$out : ADDRESSING MODE (GENERAL),
116 1243 1 lib$crf_ins_key : ADDRESSING MODE (GENERAL), !Insert key name definition into cross reference
117 1244 1 lib$crf_ins_ref : ADDRESSING MODE (GENERAL), !Insert key reference into cross reference
118 1245 1 lib$crf_output : ADDRESSING MODE (GENERAL), !Output cross reference
119 1246 1 lib$lp_lines : ADDRESSING MODE (GENERAL), !Figure number of lines per page
120 1247 1 lbr$find : ADDRESSING MODE (GENERAL), !Find module by RFA
121 1248 1 lbr$get_record : ADDRESSING MODE (GENERAL), !Read a record
122 1249 1 lbr$get_index : ADDRESSING MODE (GENERAL); !Return contents of index
123 1250 1
124 1251 1 EXTERNAL LITERAL
125 1252 1 lbr$_nomtchfou, !No match found
126 1253 1 lib$_notobjlib, !Not an object library
127 1254 1 lib$_nomtchfou, !No match found
128 1255 1 lib$_indexerr, !Index error
129 1256 1 lib$_lookuperr; !Lookup error in library
130 1257 1
131 1258 1 OWN
132 1259 1 gsdoffset, !Offset into concatenated gsd record
133 1260 1 symbolstring : REF VECTOR [BYTE], !Pointer to current symbol
134 1261 1 recdesc : BBLOCK [dsc$_s_bln], !String descriptor for record
135 1262 1 found1, !True when one found
136 1263 1 curmod : REF BBLOCK, !Pointer to current module descriptor
137 1264 1 symbolval, !Value of current symbol
138 1265 1 linecount, !Number of lines on current page
139 1266 1 linesperpage, !Number of lines per page
140 1267 1 pagenum, !Current page number
141 1268 1 current_time : VECTOR [2], !Current time
142 1269 1 listingwidth : INITIAL (lib$_lisreclng), !Width of listing line
143 1270 1 cursubhdr : REF BBLOCK, !Pointer to current sub header
144 1271 1 curheader : REF BBLOCK, !Pointer to current header descriptor
145 1272 1 maxsymlng, !Maximum length of symbol seen
146 1273 1 maxmodlng, !Maximum length of module seen
147 1274 1 a_zero : INITIAL (0), !A long zero
148 1275 1 listdefext : descriptor ('SYSSDISK:[] .LIS'), !New default filename for listing file
149 1276 1
150 1277 1 Headers for top of page
151 1278 1
152 1279 1 crfsymheader : descriptor ('Cross reference by symbol'),
153 1280 1 crfvalheader : descriptor ('Cross reference by value'),
154 1281 1 crfmdfheader : descriptor ('Cross reference of definitions in module'),
155 1282 1 crfmrheader : descriptor ('Cross reference of references in module'),
156 1283 1
157 1284 1 Sub headers for cross reference
158 1285 1
159 1286 1 crfsymsubhdr : BBLOCK [dsc$_s_bln*2], !Final string descriptor stored here
160 P 1287 1 crfsymsubproto1 : BBLOCK [dsc$_s_bln] INITIAL (stringdesc ( !Subheader for cref by symbol
161 1288 1 '<Symbol!>!T!<Value!>!')),
162 P 1289 1 crfsymsubproto2 : BBLOCK [dsc$_s_bln] INITIAL (stringdesc (
163 1290 1 '<Defined By!>Referenced By ...')),
164 P 1291 1 crfsymsubproto3 : BBLOCK [dsc$_s_bln] INITIAL (stringdesc (
165 1292 1 '<!6*-!>!112!5*-!>!')),
166 P 1293 1 crfsymsubproto4 : BBLOCK [dsc$_s_bln] INITIAL (stringdesc (
167 1294 1 '<!10*-!>!17*-!>!')),
168 P 1295 1 crfvalsubhdr : BBLOCK [dsc$_s_bln*2] INITIAL (stringdesc ( !Subheader for cref by value
169 1296 1 '!<Value!>Symbols...'),
170 1297 1 stringdesc ('!10<!5*-!>!10*-!')),
171 1298 1 crfmdfsubhdr : BBLOCK [dsc$_s_bln*2], !Final string descriptor stored here
```

```

: 172 P 1299 1 crfmdfsubproto1 : BBLOCK [dsc$c_s_bln] INITIAL (stringdesc (      !Subheader for cref of module defs
: 173 1300 1      '<Module!>Symbol definitions...')),
: 174 P 1301 1 crfmdfsubproto2 : BBLOCK [dsc$c_s_bln] INITIAL (stringdesc(
: 175 1302 1      '<!6*-!>!21*-!')),
: 176 1303 1 crfmrfsubhdr : BBLOCK [dsc$c_s_bln*2]      !Final string descriptor stored here
: 177 P 1304 1 crfmrfsubproto1 : BBLOCK [dsc$c_s_bln] INITIAL (stringdesc (      !Subheader for cref of module refs
: 178 1305 1      '<Module!>Symbol references...')),
: 179 P 1306 1 crfmrfsubproto2 : BBLOCK [dsc$c_s_bln] INITIAL (stringdesc (
: 180 1307 1      '<!6*-!>!20*-!')),
: 181 1308 1 fao_newpage : descriptor ('!^'),      !Output a form feed
: 182 1309 1 fao_pageheader : descriptor ('!40AS !60AS !17XD Page !UL'),      !Top of page header
: 183 1310 1 fao_crefao : descriptor ('!AC!UL!AC'),      !Fao control string to create fao co
: 184 1311 1 fao_crefao2 : descriptor ('!AC!UL!AS!UL!AS'),      !Fao control string to create fao co
: 185 1312 1 fao_crefao3 : descriptor ('!AC!UL!AS'),      !Fao control string to create fao co
: 186 1313 1 fao_exclam : countedstring ('!'),
: 187 1314 1 fao_ac : countedstring ('AC'),
: 188 1315 1 fao_sym : BBLOCK [8],      !Fao control string created here
: 189 1316 1 fao_symdesc : BBLOCK [dsc$c_s_bln] INITIAL (8, fao_sym),      !String descriptor for fao_sym
: 190 1317 1 fao_mod : BBLOCK [8],      !Fao control string created here
: 191 1318 1 fao_modesc : BBLOCK [dsc$c_s_bln] INITIAL (8, fao_mod),      !String descriptor for fao_mod
: 192 1319 1
: 193 1320 1 : Cross reference data base
: 194 1321 1
: 195 P 1322 1 symbolfield : $crffield (fao_string = '!3!AC',      ! for long names
: 196 1323 1      field_width = 3),
: 197 1324 1 symbolend : $crffieldend,
: 198 1325 1
: 199 P 1326 1 spacefield : $crffield (fao_string = ' ',      ! spaces between symbol and vale
: 200 1327 1      field_width = 1),
: 201 1328 1 spaceend : $crffieldend,
: 202 1329 1
: 203 P 1330 1 nullfield : $crffield (fao_string = ' ',      ! field of zero width
: 204 1331 1      field_width = 0),
: 205 1332 1 nullend : $crffieldend,
: 206 1333 1
: 207 P 1334 1 valuefield : $crffield (fao_string = '!XL',      ! hex longword value
: 208 1335 1      field_width = 8),
: 209 1336 1 valueend : $crffieldend,
: 210 1337 1
: 211 P 1338 1 valueflags : $crffield (fao_string = '!3* ',      ! default suffix
: 212 1339 1      field_width = 3),      ! when none of following applies
: 213 P 1340 1      reloc_suf : $crffield (bit_mask = gsy$m_rel,      ! relocatable only
: 214 1341 1      fao_string = '-R',      ! suffix
: 215 1342 1      field_width = 3),
: 216 P 1343 1      undef_suf : $crffield (bit_mask = gsy$m_def,      ! suffix on undefined
: 217 1344 1      set_clear = 0,      ! symbol values
: 218 1345 1      fao_string = '!-+',      !
: 219 1346 1      field_width = 3),
: 220 1347 1 valflgsend : $crffieldend,
: 221 1348 1
: 222 P 1349 1 refnceflags : $crffield (fao_string = '!3* ',      ! default spaces when none of
: 223 1350 1      field_width = 3),      ! the following references
: 224 P 1351 1      weak_pre : $crffield (bit_mask = gsy$m_weak,      ! prefix for weak reference
: 225 1352 1      fao_string = '!WK-',
: 226 1353 1      field_width = 3),
: 227 1354 1 reflagsend : $crffieldend,
: 228 1355 1

```

```

: 229 P 1356 1 referencer : Scrffield (fao_string = '!31AC', ! for long names
: 230 1357 1 field_width = 32),
: 231 1358 1 reflngend : Scrffieldend,
: 232 1359 1
: 233 P 1360 1 relsymval : Scrffield (fao_string = '!4*',
: 234 1361 1 field_width = 4),
: 235 P 1362 1 reloc_pre : Scrffield (bit_mask = gsy$m_rel,
: 236 P 1363 1 fao_string = 'R-',
: 237 1364 1 field_width = 4),
: 238 1365 1 relsymvalend : Scrffieldend,
: 239 1366 1
: 240 P 1367 1 lib_crfsymval : Scrffctltable (keytype = ascic, ! define the table that drives
: 241 P 1368 1 error = crferror,
: 242 P 1369 1 output = lib_crfout,
: 243 P 1370 1 key1table = symbolfield,
: 244 P 1371 1 key2table = spacefield, ! the cross reference to produce
: 245 P 1372 1 val1table = valuefield, ! the formatted symbol table
: 246 P 1373 1 val2table = valueflags, ! listing lines
: 247 P 1374 1 ref2table = referencer, ! each entry here is a
: 248 1375 1 refltable = refnceflags), ! pointer to the formats
: 249 P 1376 1 lib_crfbyval : Scrffctltable (keytype = bin_u32, ! define the table that drives
: 250 P 1377 1 error = crferror,
: 251 P 1378 1 output = lib_crfout,
: 252 P 1379 1 key1table = valuefield, ! cref to produce the cross
: 253 P 1380 1 key2table = spacefield, ! reference of symbols
: 254 P 1381 1 val1table = 0, ! sorted by value
: 255 P 1382 1 val2table = 0,
: 256 P 1383 1 ref2table = symbolfield,
: 257 1384 1 refltable = spacefield),
: 258 P 1385 1 lib_crfmoddef : Scrffctltable (keytype = ascic, ! Cross ref of syms defined in module
: 259 P 1386 1 error = crferror,
: 260 P 1387 1 output = lib_crfout,
: 261 P 1388 1 key1table = symbolfield,
: 262 P 1389 1 key2table = spacefield,
: 263 P 1390 1 val1table = 0,
: 264 P 1391 1 val2table = 0,
: 265 P 1392 1 ref2table = referencer,
: 266 1393 1 refltable = 0),
: 267 P 1394 1 lib_crfmodref : Scrffctltable (keytype = ascic, ! Cross ref of syms referenced in module
: 268 P 1395 1 error = crferror,
: 269 P 1396 1 output = lib_crfout,
: 270 P 1397 1 key1table = symbolfield,
: 271 P 1398 1 key2table = spacefield,
: 272 P 1399 1 val1table = 0,
: 273 P 1400 1 val2table = 0,
: 274 P 1401 1 ref2table = referencer,
: 275 1402 1 refltable = 0);
: 276 1403 1
: 277 1404 1
: 278 1405 1 BIND
: 279 1406 1 reclng = recdesc [dsc$w_length] : WORD, !Name the length of the record
: 280 1407 1 objrec = recdesc [dsc$a_pointer] : REF BBLOCK, !and the pointer
: 281 1408 1 objvec = recdesc [dsc$a_pointer] : REF VECTOR [,BYTE], ! name pointer as a bytevector pointer
: 282 1409 1 recdispatch = PLIT ( !Set up maximum allowed record type
: 283 1410 1 prohdr, !0 - module header
: 284 1411 1 progsd, !1 - gsd records
: 285 1412 1 pronul, !2 - tir

```

```
: 286      1413  1  
: 287      1414  1  
: 288      1415  1  
: 289      1416  1  
: 290      1417  1
```

```
pronul,  
pronul,  
pronul,  
pronul,  
pronul) : VECTOR;
```

```
!3 - end of module  
!4 - dbg - check sequence and copy  
!5 - tbt - check sequence and copy  
!6 - lnk - check sequence and copy  
!7 - eomw
```



```
292 1418 1 GLOBAL ROUTINE lib_cross_obj = ,
293 1419 2 BEGIN
294 1420 2 !++
295 1421 2
296 1422 2 Read all modules in object library and create cross reference.
297 1423 2 Then print it
298 1424 2
299 1425 2 Implicit inputs:
300 1426 2
301 1427 2 Library is open
302 1428 2
303 1429 2 Outputs:
304 1430 2
305 1431 2 The cross reference listing file is written and closed.
306 1432 2
307 1433 2 --
308 1434 2
309 1435 2 LOCAL
310 1436 2 symsubhdr1 : BBLOCK [lib$c_lisreclng],
311 1437 2 symsubhdr2 : BBLOCK [lib$c_lisreclng],
312 1438 2 mdfsubhdr1 : BBLOCK [lib$c_lisreclng],
313 1439 2 mdfsubhdr2 : BBLOCK [lib$c_lisreclng],
314 1440 2 mrfsubhdr1 : BBLOCK [lib$c_lisreclng],
315 1441 2 mrfsubhdr2 : BBLOCK [lib$c_lisreclng],
316 1442 2 desc : REF BBLOCK,
317 1443 2 lnblk : REF BBLOCK,
318 1444 2 status,
319 1445 2 keydesc : BBLOCK [dsc$c_s_bln];
320 1446 2
321 1447 2 BIND
322 1448 2 symbolfaodesc = symbolfield [fld$b_faodsc] : BBLOCK, !FAO descriptor for cref
323 1449 2 refncfaodesc = referencer [fld$b_faodsc] : BBLOCK,
324 1450 2 outdefext = lib$gl_outfdb [fdb$l_defext] : BBLOCK;
325 1451 2
326 1452 2 BUILTIN
327 1453 2 INSQUE,
328 1454 2 REMQUE;
329 1455 2
330 1456 2 IF .lib$gl_type NEQ lbr$c_typ_obj
331 1457 3 THEN BEGIN
332 1458 3 SIGNAL (lib$_notobjlib, 1, lib$gl_libfdb [fdb$l_namdesc]);
333 1459 3 RETURN lib$_notobjlib;
334 1460 3 END;
335 1461 2
336 1462 2 found1 = false;
337 1463 2 maxsymlng = 6; !Min symbol length is 6
338 1464 2 maxmodlng = 6; ! as is module
339 1465 2 $GETTIM (TIMADR = current time); !Get time for listing
340 1466 2 CH$MOVE (dsc$c_s_bln, listdefext, outdefext); !Set default extn.
341 1467 2 IF NOT .lib$gl_cflmsk [lib$v_only] !If not /ONLY
342 1468 3 THEN BEGIN
343 1469 3 perform (lib_get_zmem (lnb$c_fixedsz + 1, lnblk)); !Then make up one
344 1470 3 INSQUE (.lnblk, .lib$gl_modlisl [1]); !Insert into queue
345 1471 3 lnblk [lnb$b_namlng] = T;
346 1472 3 CH$FILL (%ASCII '*', 1, lnblk [lnb$t_name]);
347 1473 3 END;
348 1474 2 !
```

```
349 1475 2 ! Figure # lines per page
350 1476 2
351 1477 2 linesperpage = lib$lp_lines () - 6;
352 1478 2 pagenum = 0; ! and start with page 1
353 1479 2
354 1480 2 ! Look at each block set up for /ONLY, and cross reference each module
355 1481 2 ! that matches.
356 1482 2
357 1483 2 WHILE NOT REMQUE (.lib$gl_modlist, lnblk)
358 1484 3 DO BEGIN
359 1485 3     keydesc [dsc$w_length] = .lnblk [lnb$b_namlng];
360 1486 3     keydesc [dsc$a_pointer] = lnblk [lnb$t_name];
361 1487 4     IF (status = lbr$get_index (lib$gl_libctl, lib$gl_modnamix, !Search index for matches
362 1488 3         cross_one_module, keydesc)) EQL lbr$_nomtchfou
363 1489 3         THEN SIGNAL (lib$_nomtchfou, 1, keydesc)
364 1490 3         ELSE IF NOT .status
365 1491 3             THEN SIGNAL (lib$_indexerr, 1, lib$gl_libfdb [fdb$l_namdesc],
366 1492 3                 .status, .lbr$gl_rmsstv);
367 1493 3     lib_free_mem (lnb$c_fixedsize + .lnblk [lnb$b_namlng], .lnblk); !Deallocate the block
368 1494 2     END;
369 1495 2
370 1496 2 ! Call CRF to output the listing, close files and we are done
371 1497 2
372 1498 2 IF NOT .found1
373 1499 2 THEN RETURN lbr$_nomtchfou;
374 1500 2 P perform (lib_open_out (.lib$gl_outfdb, lib$gl_libfdb [fdb$t_nam], !Open output file with carriage control
375 1501 2     true));
376 1502 2
377 1503 2 ! Create fao strings for symbols and modules
378 1504 2
379 1505 2 SY$FAO (fao_crefao, fao_syndesc, fao_syndesc, fao_exclam, !Create fao control string for symbols
380 1506 2     .maxsymlng, fao_ac);
381 1507 2 SY$FAO (fao_crefao, fao_modesc, fao_modesc, fao_exclam, !Create fao control string for modules
382 1508 2     .maxmodlng, fao_ac);
383 1509 2
384 1510 2 symbolfield [fld$b_maxlng] = .maxsymlng;
385 1511 2 referencer [fld$b_maxlng] = .maxmodlng + 1;
386 1512 2
387 1513 2 IF .lib$gl_creflags [lib$v_crfbysym]
388 1514 3 THEN BEGIN
389 1515 3     CH$MOVE (dsc$c_s_bln, fao_syndesc, symbolfaodesc); !Set up string descriptors
390 1516 3     CH$MOVE (dsc$c_s_bln, fao_modesc, refncfaodesc);
391 1517 3     crfsymsubhdr [dsc$w_length] = lib$c_lisreclng; !Set up sub header fao control strings
392 1518 3     crfsymsubhdr [dsc$a_pointer] = symsubhdr1;
393 1519 3     SY$FAO (fao_crefao2, crfsymsubhdr, crfsymsubhdr, fao_exclam,
394 1520 3         .maxsymlng + 1, crfsymsubproto1,
395 1521 3         .maxmodlng + 7, crfsymsubproto2);
396 1522 3     desc = crfsymsubhdr + dsc$c_s_bln;
397 1523 3     desc [dsc$w_length] = lib$c_lisreclng;
398 1524 3     desc [dsc$a_pointer] = symsubhdr2;
399 1525 3     SY$FAO (fao_crefao2, .desc, .desc, fao_exclam,
400 1526 3         .maxsymlng + 1, crfsymsubproto3,
401 1527 3         .maxmodlng + 7, crfsymsubproto4);
402 1528 3     curheader = crfsymheader;
403 1529 3     cursubhdr = crfsymsubhdr;
404 1530 3     linecount = .linesperpage; !force new page
405 1531 3     crf$out (lib_crf$ymval, .listingwidth, .linesperpage - 4,
```

```
406 1532 3 .linesperpage, crf$k_defs_refs, crf$k_delete);
407 1533 3 END;
408 1534 2
409 1535 2 IF .lib$gl_creflags [lib$v_crfbyval]
410 1536 2 THEN BEGIN
411 1537 2 curheader = crfvalheader;
412 1538 2 cursubhdr = crfvalsubhdr;
413 1539 2 linecount = .linesperpage; !Force new page
414 1540 2 crf$out (lib_crfbyval, .listingwidth, .linesperpage - 4,
415 1541 2 .linesperpage, crf$k_vals_refs, crf$k_delete);
416 1542 2 END;
417 1543 2
418 1544 2 IF .lib$gl_creflags [lib$v_crfbymod]
419 1545 2 THEN BEGIN
420 1546 2 CH$MOVE (dsc$c_s_bln, fao_modesc, symbol aodesc); !Set up string descriptors
421 1547 2 CH$MOVE (dsc$c_s_bln, fao_syndesc, refncfaodesc);
422 1548 2 symbolfield [fld$b_maxlng] = .maxmodlng;
423 1549 2 referencer [fld$b_maxlng] = .maxsymlng + 1;
424 1550 2 crfmdfsubhdr [dsc$w_length] = lib$c_lisreclng; !Create the fao control string for sub header
425 1551 2 crfmdfsubhdr [dsc$a_pointer] = mdfsubhdr1;
426 1552 2 SY$FAO (fao_crefao3, crfmdfsubhdr, crfmdfsubhdr, fao_exclam,
427 1553 2 .maxmodlng+1, crfmdfsubproto1);
428 1554 2 desc = crfmdfsubhdr + dsc$c_s_bln;
429 1555 2 desc [dsc$w_length] = lib$c_lisreclng;
430 1556 2 desc [dsc$a_pointer] = mdfsubhdr2;
431 1557 2 SY$FAO (fao_crefao3, .desc, .desc, fao_exclam,
432 1558 2 .maxmodlng+1, crfmdfsubproto2);
433 1559 2 curheader = crfmdfheader;
434 1560 2 cursubhdr = crfmdfsubhdr;
435 1561 2 linecount = .linesperpage; !Force new page
436 1562 2 crf$out (lib_crfmoddef, .listingwidth, .linesperpage - 4,
437 1563 2 .linesperpage, crf$k_vals_refs, crf$k_delete);
438 1564 2
439 1565 2 crfmrsubhdr [dsc$w_length] = lib$c_lisreclng; !Create the fao control string for sub header
440 1566 2 crfmrsubhdr [dsc$a_pointer] = mrsubhdr1;
441 1567 2 SY$FAO (fao_crefao3, crfmrsubhdr, crfmrsubhdr, fao_exclam,
442 1568 2 .maxmodlng+1, crfmrsubproto1);
443 1569 2 desc = crfmrsubhdr + dsc$c_s_bln;
444 1570 2 desc [dsc$w_length] = lib$c_lisreclng;
445 1571 2 desc [dsc$a_pointer] = mrsubhdr2;
446 1572 2 SY$FAO (fao_crefao3, .desc, .desc, fao_exclam,
447 1573 2 .maxmodlng+1, crfmrsubproto2);
448 1574 2 curheader = crfmrheader;
449 1575 2 cursubhdr = crfmrsubhdr;
450 1576 2 linecount = .linesperpage; !Force new page
451 1577 2 crf$out (lib_crfmodref, .listingwidth, .linesperpage - .linecount,
452 1578 2 .linesperpage, crf$k_vals_refs, crf$k_delete);
453 1579 2 END;
454 1580 2 lib_close_out (.lib$gl_outfdb, false);
455 1581 2 RETURN true
456 1582 1 END; !Of lib_cross_lib
```

```
.TITLE LIB_CROSS
.IDENT \V04-000\
.PSECT $PLITS,NOWRT,NOEXE,2
```

```

53 49 4C 2E 5D 5B 3A 4B 53 49 44 24 53 59 53 00000 P.AAA: .ASCII \SYS$DISK:[].LIS\<0>
00 0000F
65 63 6E 65 72 65 66 65 72 20 73 73 6F 72 43 00010 P.AAB: .ASCII \Cross reference by symbol\<0><0><0>
00 00 00 6C 6F 62 6D 79 73 20 79 62 20 0001F
65 63 6E 65 72 65 66 65 72 20 73 73 6F 72 43 0002C P.AAC: .ASCII \Cross reference by value\
00 65 75 6C 61 76 20 79 62 20 0003B
65 63 6E 65 72 65 66 65 72 20 73 73 6F 72 43 00044 P.AAD: .ASCII \Cross reference of definitions in module\
73 6E 6F 69 74 69 6E 69 66 65 64 20 66 6F 20 00053
00 65 6C 75 64 6F 6D 20 6E 69 20 00062
65 63 6E 65 72 65 66 65 72 20 73 73 6F 72 43 0006C P.AAE: .ASCII \Cross reference of references in module-
20 73 65 63 6E 65 72 65 66 65 72 20 66 6F 20 0007B
00 65 6C 75 64 6F 6D 20 6E 69 20 0008A
61 56 3C 31 31 21 3E 21 6C 6F 62 6D 79 53 3C 00094 P.AAF: .ASCII \<Symbol!>!11<Value!>!<0><0><0>
00 00 00 21 3E 21 65 75 6C 000A3
65 52 3E 21 79 42 20 64 65 6E 69 66 65 44 3C 000AC P.AAG: .ASCII \<Defined By!>Referenced By ...<0><0>
2E 2E 2E 20 79 42 20 64 65 63 6E 65 72 65 66 000BB
00 00 00 00 00 000CA
2D 2A 35 21 3C 31 31 21 3E 21 2D 2A 36 21 3C 000CC P.AAH: .ASCII \<!6*-!>!11<!5*-!>!<0><0>
00 00 21 3E 21 000DB
00 00 2D 2A 37 31 21 3E 21 2D 2A 30 31 21 3C 000E0 P.AAI: .ASCII \<!10*-!>!17*-!<0><0><0>
000EF
62 6D 79 53 3E 21 65 75 6C 61 56 3C 30 31 21 000F0 P.AAJ: .ASCII \!10<Value!>Symbols...<0><0><0>
00 00 00 2E 2E 2E 73 6C 6F 000FF
2D 2A 30 31 21 3E 21 2D 2A 35 21 3C 30 31 21 00108 P.AAK: .ASCII \!10<!5*-!>!10*-!<0>
00117
6C 6F 62 6D 79 53 3E 21 65 6C 75 64 6F 4D 3C 00118 P.AAL: .ASCII \<Module!>Symbol definitions...<0><0>
2E 2E 2E 73 6E 6F 69 74 69 6E 69 66 65 64 20 00127
00 00 00 00 000136
6C 6F 62 6D 2A 31 32 21 3E 21 2D 2A 36 21 3C 00138 P.AAM: .ASCII \<!6*-!>!21*-!
00 2E 2E 2E 73 53 3E 21 65 6C 75 64 6F 4D 3C 00144 P.AAN: .ASCII \<Module!>Symbol references...<0><0>
00 2E 2E 2E 73 65 63 6E 65 72 65 66 65 72 20 00153
00 00 00162
00 00 00163
00 00 00164 P.AAO: .ASCII <0>
00 00 5E 21 00170 P.AAP: .ASCII \<!6*-!>!20*-!
37 31 21 20 53 41 30 36 21 20 53 41 30 34 21 00174 P.AAQ: .ASCII \!^!<0><0>
00 4C 55 21 20 65 67 61 50 20 20 44 25 00183
00 00 00 43 41 21 4C 55 21 43 41 21 00190 P.AAR: .ASCII \!AC!UL!AC!<0><0><0>
53 41 21 4C 55 21 53 41 21 4C 55 21 43 41 21 0019C P.AAS: .ASCII \!AC!UL AS!UL!AS!<0>
00 001AB
00 00 00 53 41 21 4C 55 21 43 41 21 001AC P.AAT: .ASCII \!AC!UL!AS!<0><0><0>
00 00 00 43 41 31 33 21 001B8 P.AAU: .ASCII \!31AC!<0><0><0>
00 00 00 00 20 001C0 P.AAV: .ASCII \ \<0><0><0>
00 00 00 00 20 001C4 P.AAW: .ASCII \ \<0><0><0>
00 4C 58 21 001C8 P.AAX: .ASCII \!XL!<0>
20 2A 33 21 001CC P.AAY: .ASCII \!3* \
00 20 52 2D 001D0 P.AAZ: .ASCII \-R \<0>
00 20 2A 2D 001D4 P.ABA: .ASCII \-* \<0>
20 2A 33 21 001D8 P.ABB: .ASCII \!3* \
00 2D 48 57 001DC P.ABC: .ASCII \WK-\<0>
00 00 00 43 41 31 33 21 001E0 P.ABD: .ASCII \!31AC!<0><0><0>
20 2A 34 21 001E8 P.ABE: .ASCII \!4* \
2D 52 20 20 001EC P.ABF: .ASCII \ R-\
00000008 001F0
0000000V 0000000V 0000C000V 00000000V 00000000V 00000000V 001F4 P.ABG: .ADDRESS PROHDR, PROGSD, PRONUL, PRONUL, PRONUL, -
00000000V 00000000V 0020C PRONUL, PRONUL, PRONUL

```

.PSECT \$OWNS,NOEXE,2

00000	GSDOFFSET:			
		.BLKB	4	
00004	SYMBOLSTRING:			
		.BLKB	4	
00008	RECDESC:	.BLKB	8	
00010	FOUND1:	.BLKB	4	
00014	CURMOD:	.BLKB	4	
00018	SYMBOLVAL:			
		.BLKB	4	
0001C	LINECOUNT:			
		.BLKB	4	
00020	LINESPERPAGE:			
		.BLKB	4	
00024	PAGENUM:	.BLKB	4	
00028	CURRENT_TIME:			
		.BLKB	8	
00000084	00030 LISTINGWIDTH:			
		.LONG	132	:
00034	CURSUBHDR:			
		.BLKB	4	
00038	CURHEADER:			
		.BLKB	4	
0003C	MAXSYMLNG:			
		.BLKB	4	
00040	MAXMODLNG:			
		.BLKB	4	
00000000	00044 A_ZERO:	.LONG	0	:
0000000F	00048 LISTDEFEXT:			
		.LONG	15	:
00000000'	0004C	.ADDRESS	P.AAA	:
00000019	00050 CRFSYMHEADER:			:
		.LONG	25	:
00000000'	00054	.ADDRESS	P.AAB	:
00000018	00058 CRFVALHEADER:			:
		.LONG	24	:
00000000'	0005C	.ADDRESS	P.AAC	:
00000028	00060 CRFMDFHEADER:			:
		.LONG	40	:
00000000'	00064	.ADDRESS	P.AAD	:
00000027	00068 CRFMRFHEADER:			:
		.LONG	39	:
00000000'	0006C	.ADDRESS	P.AAF	:
	00070 CRFSYMSUBHDR:			:
		.BLKB	16	:
00000015	00080 CRFSYMSUBPROTO1:			:
		.LONG	21	:
00000000'	00084	.ADDRESS	P.AAF	:
0000001E	00088 CRFSYMSUBPROTO2:			:
		.LONG	30	:
00000000'	0008C	.ADDRESS	P.AAG	:
00000012	00090 CRFSYMSUBPROTO3:			:
		.LONG	18	:
00000000'	00094	.ADDRESS	P.AAH	:
0000000D	00098 CRFSYMSUBPROTO4:			:

: R

00000000'	0009C		.LONG	13	:	
00000015	000A0	CRFVALSUBHDR:	.ADDRESS	P.AAI	:	
			.LONG	21	:	
00000000'	000A4		.ADDRESS	P.AAJ	:	
0000000F	000A8		.LONG	15	:	
00000000'	000AC		.ADDRESS	P.AAK	:	
	000B0	CRFMDFSUBHDR:			:	
			.BLKB	16	:	
0000001E	000C0	CRFMDFSUBPROTO1:			:	
			.LONG	30	:	
00000000'	000C4		.ADDRESS	P.AAL	:	
0000000C	000C8	CRFMDFSUBPROTO2:			:	
			.LONG	12	:	
00000000'	000CC		.ADDRESS	P.AAM	:	
	000D0	CRFMRFSUBHDR:			:	
			.BLKB	16	:	
0000001D	000E0	CRFMRFSUBPROTO1:			:	
			.LONG	29	:	
00000000'	000E4		.ADDRESS	P.AAN	:	
0000000C	000E8	CRFMRFSUBPROTO2:			:	
			.LONG	12	:	
00000000'	000EC		.ADDRESS	P.AAO	:	
00000002	000F0	FAO_NEWPAGE:			:	
			.LONG	2	:	
00000000'	000F4		.ADDRESS	P.AAP	:	
0000001B	000F8	FAO_PAGEHEADER:			:	
			.LONG	27	:	
00000000'	000FC		.ADDRESS	P.AAQ	:	
00000009	00100	FAO_CREFAO:			:	
			.LONG	9	:	
00000000'	00104		.ADDRESS	P.AAR	:	
0000000F	00108	FAO_CREFAO2:			:	
			.LONG	15	:	
00000000'	0010C		.ADDRESS	P.AAS	:	
00000009	00110	FAO_CREFAO3:			:	
			.LONG	9	:	
00000000'	00114		.ADDRESS	P.AAT	:	
	01	00118	FAO_EXCLAM:		:	
			.BYTE	1	:	
	21	00119	.ASCII	!\	:	
		0011A	.BLKB	2	:	
	02	0011C	FAO_AC:	.BYTE	2	:
43	41	0011D	.ASCII	\AC\	:	
		0011F	.BLKB	1	:	
		00120	FAO_SYM:	.BLKB	8	:
00000008	00128	FAO_SYMDESC:			:	
			.LONG	8	:	
00000000'	0012C		.ADDRESS	FAO_SYM	:	
	00130	FAO_MOD:	.BLKB	8	:	
00000008	00138	FAO_MODESC:			:	
			.LONG	8	:	
00000000'	0013C		.ADDRESS	FAO_MOD	:	
	8000	00140	SYMBOLFIELD:		:	
			.WORD	-32768	:	
	01	00142	.BYTE	1	:	
	1F	00143	.BYTE	31	:	

00000005	00144	.LONG	5	
00000000	00148	.ADDRESS	P.AAU	:
0000	0014C	SYMBOLEND:		:
		.WORD	0	:
	0014E	.BLKB	2	:
8000	00150	SPACEFIELD:		:
		.WORD	-32768	:
01	00152	.BYTE	1	:
01	00153	.BYTE	1	:
00000001	00154	.LONG	1	:
00000000	00158	.ADDRESS	P.AAV	:
0000	0015C	SPACEEND:		:
		.WORD	0	:
	0015E	.BLKB	2	:
8000	00160	NULLFIELD:		:
		.WORD	-32768	:
01	00162	.BYTE	1	:
00	00163	.BYTE	0	:
00000001	00164	.LONG	1	:
00000000	00168	.ADDRESS	P.AAW	:
0000	0016C	NULLEND:		:
		.WORD	0	:
	0016E	.BLKB	2	:
8000	00170	VALUEFIELD:		:
		.WORD	-32768	:
01	00172	.BYTE	1	:
08	00173	.BYTE	8	:
00000003	00174	.LONG	3	:
00000000	00178	.ADDRESS	P.AAX	:
0000	0017C	VALUEEND:		:
		.WORD	0	:
	0017E	.BLKB	2	:
8000	00180	VALUEFLAGS:		:
		.WORD	-32768	:
01	00182	.BYTE	1	:
03	00183	.BYTE	3	:
00000004	00184	.LONG	4	:
00000000	00188	.ADDRESS	P.AAY	:
8008	0018C	RELOC_SUF:		:
		.WORD	-32760	:
01	0018E	.BYTE	1	:
03	0018F	.BYTE	3	:
00000003	00190	.LONG	3	:
00000000	00194	.ADDRESS	P.AAZ	:
8002	00198	UNDEF_SUF:		:
		.WORD	-32766	:
00	0019A	.BYTE	0	:
03	0019B	.BYTE	3	:
00000003	0019C	.LONG	3	:
00000000	001A0	.ADDRESS	P.ABA	:
0000	001A4	VALFLGSEND:		:
		.WORD	0	:
	001A6	.BLKB	2	:
8000	001A8	REFNCEFLAGS:		:
		.WORD	-32768	:
01	001AA	.BYTE	1	:
03	001AB	.BYTE	3	:
00000004	001AC	.LONG	4	:

```

00000000' 001B0 .ADDRESS P.ABB
      8001 001B4 WEAK_PRE:
          .WORD -32767
          01 001B6 .BYTE 1
          03 001B7 .BYTE 3
00000003' 001B8 .LONG 3
00000000' 001BC .ADDRESS P.ABC
      0000 001C0 REFLAGSEND:
          .WORD 0
          8000 001C2 .BLKB 2
          001C4 REFERENCER:
          01 001C6 .WORD -32768
          20 001C7 .BYTE 1
          00000005' 001C8 .BYTE 32
          00000000' 001C8 .LONG 5
      0000 001CC .ADDRESS P.ABD
          0000 001D0 REFLNGEND:
          .WORD 0
          8000 001D2 .BLKB 2
          001D4 RELSYMVAL:
          01 001D6 .WORD -32768
          04 001D7 .BYTE 1
          00000004' 001D8 .BYTE 4
          00000000' 001DC .LONG 4
      8008 001E0 .ADDRESS P.ABE
          001E0 RELOC_PRE:
          01 001E2 .WORD -32760
          04 001E3 .BYTE 1
          00000004' 001E4 .BYTE 4
          00000000' 001E8 .LONG 4
      0000 001EC .ADDRESS P.ABF
          001EC RELSYMVALEND:
          .WORD 0
          00000000 001EE LIB_CRFSYMVAL:
          .LONG 0
          00000000 001F2 .LONG 0
          00 00 00 00 001F6 .BYTE 0, 0, 0, 0
          00000000 001FA .LONG 0
          00000000 001FE .LONG 0, 0
          00006400 00206 .LONG 25600
          00000032 0020A .LONG 50
00000000 00000000 00000000 00000000 00000000 00000000 0020E .LONG 0, 0, 0, 0, 0, 0, 0, 0
00000000' 00000000' 00000000' 00000000' 00000000V 00000000V 00226
00000000' 00000000' 00000000' 00000000' 00000000V 00000000V 0022E .ADDRESS CRFERROR, LIB_CRFOUT, SYMBOLFIELD, -
00000000' 00000000' 00246 .SPACEFIELD, VALUEFIELD, VALUEFLAGS, -
          .REFNCEFLAGS, REFERENCER
          00000000 0024E LIB_CRFBYVAL:
          .LONG 0
          00000000 00252 .LONG 0
          00 00 00 01 00256 .BYTE 1, 0, 0, 0
          00000000 0025A .LONG 0
          00000000 0025E .LONG 0, 0
          00006400 00266 .LONG 25600
          00000032 0026A .LONG 50
00000000 00000000 00000000 00000000 00000000 00000000 0026E .LONG 0, 0, 0, 0, 0, 0, 0, 0
          00000000 00000000 00286
          00000000' 00000000' 00000000V 00000000V 0028E .ADDRESS CRFERROR, LIB_CRFOUT, VALUEFIELD, -

```



				00000000	00000000	0029E	.LONG	0, 0	SPACEFIELD	:
				00000000'	00000000'	002A6	.ADDRESS	SPACEFIELD, SYMBOLFIELD		:
					00000000	002AE	LIB_CRFMODDEF:			:
					00000000	002B2	.LONG	0		:
				00 00	00 00	002B6	.BYTE	0, 0, 0, 0		:
					00000000	002BA	.LONG	0		:
				00000000	00000000	002BE	.LONG	0, 0		:
					00006400	002C6	.LONG	25600		:
					00000032	002CA	.LONG	50		:
00000000	00000000	00000000	00000000	00000000	00000000	002CE	.LONG	0, 0, 0, 0, 0, 0, 0, 0		:
				00000000	00000000	002E6				:
				00000000'	00000000'	002EE	.ADDRESS	CRFERROR, LIB_CRFOUT, SYMBOLFIELD, -		:
					00000000	002FE	SPACEFIELD			:
				00000000	00000000	0030A	.LONG	0, 0, 0		:
					00000000'	0030A	.ADDRESS	REFERENCER		:
					00000000	0030E	LIB_CRFMODREF:			:
					00000000	00312	.LONG	0		:
				00 00	00 00	00316	.BYTE	0, 0, 0, 0		:
					00000000	0031A	.LONG	0		:
				00000000	00000000	0031E	.LONG	0, 0		:
					00006400	00326	.LONG	25600		:
					00000032	0032A	.LONG	50		:
00000000	00000000	00000000	00000000	00000000	00000000	0032E	.LONG	0, 0, 0, 0, 0, 0, 0, 0		:
				00000000	00000000	00346				:
				00000000'	00000000'	0034E	.ADDRESS	CRFERROR, LIB_CRFOUT, SYMBOLFIELD, -		:
					00000000	0035E	SPACEFIELD			:
				00000000	00000000	0036A	.LONG	0, 0, 0		:
					00000000'	0036A	.ADDRESS	REFERENCER		:

RECLNG=	RECDESC
OBJREC=	RECDESC+4
OBJVEC=	RECDESC+4
RECDISPATCH=	P.ABG
SYMBOLFAODESC=	SYMBOLFIELD+4
REFNCFAODESC=	REFERENCER+4
.EXTRN	LBR\$GL_RMSSTV, LIB\$GL_CREFLAGS
.EXTRN	LIB\$GL_MODNAMIX
.EXTRN	LIB\$GL_MODLISL, LIB\$GL_OUTFDB
.EXTRN	LIB\$GL_CTLMSK, LIB\$GL_CIBFDB
.EXTRN	LIB\$GL_INPFDB, LIB\$AL_RAB
.EXTRN	LIB\$GL_TYPE, LIB\$GL_LIBCTL
.EXTRN	LIB_OPEN_OUT, LIB_CLOSE_OUT
.EXTRN	LIB_GET_MEM, LIB_GET_ZMEM
.EXTRN	LIB_FREE_MEM, LIB_LOG_OP
.EXTRN	SYSS\$FAO, CRF\$INSRTKEY
.EXTRN	CRF\$INSRTREF, CRF\$OUT
.EXTRN	LIB\$CRF_INS_KEY
.EXTRN	LIB\$CRF_INS_REF
.EXTRN	LIB\$CRF_OUTPUT, LIB\$LP_LINES
.EXTRN	LBR\$FIND, LBR\$GET_RECORD
.EXTRN	LBR\$GET_INDEX, LBR\$NOMTCHFOU
.EXTRN	LIB\$NOTOBLIB, LIB\$NOMTCHFOU
.EXTRN	LIB\$INDEXERR, LIB\$_COOKUPERR
.EXTRN	SYSS\$GETTIM

```

.PSECT $CODE$,NOWRT,2
OFFC 00000
.ENTRY LIB_CROSS_OBJ, Save R2,R3,R4,R5,R6,R7,R8,- 1418
          R9,R10,R11
          #LIB$ NOTOBJLIB, R11
          LIB$SIGNAL, R10
          CRF$OUT, R9
          SYSS$FAO, R8
          MAXMODLNG, R7
          -804(SP), SP
52      0000G CF      08 C1 00028 ADDL3 #8, LIB$GL_OUTFDB, R2 1450
          01      0000G CF D1 0002E Cmpl LIB$GL_TYPE, #1 1456
          11 13 00033 BEQL 1$
7E      0000G CF      10 C1 00035 ADDL3 #16, LIB$GL_LIBFDB, -(SP) 1458
          01 DD 0003B PUSHL #1
          5B DD 0003D PUSHL R11
          6A      03 FB 0003F CALLS #3, LIB$SIGNAL
          50      5B DO 00042 MOVL R11, R0 1459
          04 00045 RET
          DO A7 D4 00046 1$: CLRL FOUND1 1462
          FC A7 06 DO 00049 MOVL #6, MAXSYMLNG 1463
          67 06 DO 0004D MOVL #6, MAXMODLNG 1464
          E8 A7 9F 00050 PUSHAB CURRENT TIME 1465
          00 01 FB 00053 CALLS #1, SYSS$GETTIM
62      08 A7 08 2B 0005A MOV3 #8, LISTDEFEXT, (R2) 1466
1C      0000G CF      02 E0 0005F BBS #2, LIB$GL_CTLMSK+3, 3$ 1467
          5E DD 00065 PUSHL SP 1469
          0B DD 00067 PUSHL #11
          0000G CF      02 FB 00069 CALLS #2, LIB_GET_ZMEM
          01      50 EB 0006E BLBS STATUS, 2$
          04 00071 RET
          0000G DF      00 BE 0E 00072 2$: INSQUE @LNBLK, @LIB$GL_MODLISL+4 1470
          50 6E DO 00078 MOVL LNBLK, R0 1471
          09 A0 2A01 8F 80 0007B MOVW #10753, 9(R0)
          00000000G 00 00 FB 00081 3$: CALLS #0, LIB$LP_LINES 1477
          E0 A7 FA A0 9E 00088 MOVAB -6(R0), LINESPERPAGE
          E4 A7 D4 0008D CLRL PAGENUM 1478
          6E 0000G DF 0F 00090 4$: REMQUE @LIB$GL_MODLISL, LNBLK 1483
          6B 1D 00095 BVS 7$
          52 6E DO 00097 MOVL LNBLK, R2 1485
          04 AE 09 A2 9B 0009A MOVZBW 9(R2), KEYDESC
          08 AE 0A A2 9E 0009F MOVAB 10(R2), KEYDESC+4 1486
          04 AE 9F 000A4 PUSHAB KEYDESC 1487
          0000V CF 9F 000A7 PUSHAB CROSS ONE MODULE
          0000G CF 9F 000AB PUSHAB LIB$GL_MODNAMIX
          0000G CF 9F 000AF PUSHAB LIB$GL_LIBCTL
          00000000G 00 04 FB 000B3 CALLS #4, LBR$GET_INDEX
          53 50 DO 000BA MOVL R0, STATUS
          00000000G 8F 53 D1 000BD Cmpl STATUS, #LBR$_NOMTCHFOU 1488
          10 12 000C4 BNEQ 5$
          04 AE 9F 000C6 PUSHAB KEYDESC 1489
          01 DD 000C9 PUSHL #1
          00000000G 8F DD 000CB PUSHL #LIB$ NOMTCHFOU
          6A 03 FB 000D1 CALLS #3, LIB$SIGNAL
          1C 11 000D4 BRB 6$
          19 53 EB 000D6 5$: BLBS STATUS, 6$ 1490
```

			00000000G	00	DD	000D9	PUSHL	LBR\$GL_RMSSTV	1492
				53	DD	000DF	PUSHL	STATUS	
7E	0000G	CF		10	C1	000E1	ADDL3	#16, LIB\$GL_LIBFDB, -(SP)	1491
			00000000G	01	DD	000E7	PUSHL	#1	
		6A		8F	DD	000E9	PUSHL	#LIB\$ INDEXERR	
				05	FB	000EF	CALLS	#5, LIB\$SIGNAL	
		7E	09	52	DD	000F2	PUSHL	R2	1493
		6E		A2	9A	000F4	MOVZBL	9(R2), -(SP)	
	0000G	CF		0A	CO	000F8	ADDL2	#10, (SP)	
				02	FB	000FB	CALLS	#2, LIB_FREE_MEM	
		08	DO	8E	11	00100	BRB	4\$	1483
		50	00000000G	A7	E8	00102	BLBS	FOUND1, 8\$	1498
				8F	DO	00106	MOVL	#LBR\$_NOMTCHFOU, RC	1499
				04	0010D		RET		
7E	0000G	CF	00000040	01	DD	0010E	PUSHL	#1	1501
			0000G	8F	C1	00110	ADDL3	#64, LIB\$GL_LIBFDB, -(SP)	
	0000G	CF		CF	DD	0011A	PUSHL	LIB\$GL_OUTFDB	
		01		03	FB	0011E	CALLS	#3, LIB_OPEN_OUT	
				50	E8	00123	BLBS	STATUS, 9\$	
				04	00126		RET		
			00DC	C7	9F	00127	PUSHAB	FAO AC	1505
			FC	A7	DD	0012B	PUSHL	MAXSYMLNG	1506
			00D8	C7	9F	0012E	PUSHAB	FAO_EXCLAM	1505
			00E8	C7	9F	00132	PUSHAB	FAO_SYMDESC	
			00E8	C7	9F	00136	PUSHAB	FAO_SYMDESC	
			00C0	C7	9F	0013A	PUSHAB	FAO_CREFAO	
		68		06	FB	0013E	CALLS	#6, SYSSFAO	
			00DC	C7	9F	00141	PUSHAB	FAO AC	1507
				67	DD	00145	PUSHL	MAXMODLNG	1508
			00D8	C7	9F	00147	PUSHAB	FAO_EXCLAM	1507
			00F8	C7	9F	0014B	PUSHAB	FAO_MODESC	
			00F8	C7	9F	0014F	PUSHAB	FAO_MODESC	
			00C0	C7	9F	00153	PUSHAB	FAO_CREFAO	
		68		06	FB	00157	CALLS	#6, SYSSFAO	
0187	C7	0103	FC	A7	90	0015A	MOVB	MAXSYMLNG, SYMBOLFIELD+3	1510
				01	81	00160	ADDB3	#1, MAXMODLNG, REFERENCER+3	1511
			0000G	CF	E8	00166	BLBS	LIB\$GL_CREFLAGS, 10\$	1513
				008B	31	0016B	BRW	11\$	
0104	C7	00E8	C7	08	28	0016E	MOV3	#8, FAO_SYMDESC, SYMBOLFAODESC	1515
0188	C7	00F8	C7	08	28	00176	MOV3	#8, FAO_MODESC, REFNCFAODESC	1516
		30	A7	84	8F	9B	MOVZBW	#132, CRFSYMSUBHDR	1517
		34	A7	FF7C	CD	9E	MOVAB	SYMSUBHDR1, CRFSYMSUBHDR+4	1518
				48	A7	9F	PUSHAB	CRFSYMSUBPROTO2	1519
	7E		67	07	C1	0018C	ADDL3	#7, MAXMODLNG, -(SP)	1521
				40	A7	9F	PUSHAB	CRFSYMSUBPROTO1	1519
	7E	FC	A7	01	C1	00193	ADDL3	#1, MAXSYMLNG, -(SP)	1520
				00D8	C7	9F	PUSHAB	FAO_EXCLAM	1519
			30	A7	9F	0019C	PUSHAB	CRFSYMSUBHDR	
			30	A7	9F	0019F	PUSHAB	CRFSYMSUBHDR	
			00C8	C7	9F	001A2	PUSHAB	FAO_CREFAO2	
		68		08	FB	001A6	CALLS	#8, SYSSFAO	
		56	38	A7	9E	001A9	MOVAB	CRFSYMSUBHDR+8, DESC	1522
		66	84	8F	9B	001AD	MOVZBW	#132, (DESC)	1523
		04	A6	FEF8	CD	9E	MOVAB	SYMSUBHDR2, 4(DESC)	1524
				58	A7	9F	PUSHAB	CRFSYMSUBPROTO4	1525
	7E		67	07	C1	001BA	ADDL3	#7, MAXMODLNG, -(SP)	1527
				50	A7	9F	PUSHAB	CRFSYMSUBPROTO3	1525

7E	FC	A7	00D8	01 C1 001C1	ADDL3	#1, MAXSYMLNG, -(SP)	1526
				C7 9F 001C6	PUSHAB	FAO_EXCLAM	1525
				56 DD 001CA	PUSHL	DESC	
				56 DD 001CC	PUSHL	DESC	
			00C8	C7 9F 001CE	PUSHAB	FAO_CREFAO2	
		68		08 FB 001D2	CALLS	#8, -SYSS\$FAO	
	F8	A7	10	A7 9E 001D5	MOVAB	CRFSYMHEADER, CURHEADER	1528
	F4	A7	30	A7 9E 001DA	MOVAB	CRFSYMSUBHDR, CURSUBHDR	1529
		50	E0	A7 D0 001DF	MOVL	LINESPERPAGE, RO	1530
	DC	A7		50 D0 001E3	MOVL	RO, LINECOUNT	
		7E		02 7D 001E7	MOVQ	#2, -(SP)	1531
				50 DD 001EA	PUSHL	RO	1532
			FC	A0 9F 001EC	PUSHAB	-4(RO)	1531
			F0	A7 DD 001EF	PUSHL	LISTINGWIDTH	
			01AE	C7 9F 001F2	PUSHAB	LIB_CRFSYMVAL	
		69		06 FB 001F6	CALLS	#6, -CRFSOUT	
24	0000G	CF		01 E1 001F9	BBC	#1, LIB\$GL_CREFLAGS, 12\$	1535
	F8	A7	18	A7 9E 001FF	MOVAB	CRFVALHEADER, CURHEADER	1537
	F4	A7	60	A7 9E 00204	MOVAB	CRFVALSUBHDR, CURSUBHDR	1538
		50	E0	A7 D0 00209	MOVL	LINESPERPAGE, RO	1539
	DC	A7		50 D0 0020D	MOVL	RO, LINECOUNT	
		7E		01 7D 00211	MOVQ	#1, -(SP)	1540
				50 DD 00214	PUSHL	RO	1541
			FC	A0 9F 00216	PUSHAB	-4(RO)	1540
			F0	A7 DD 00219	PUSHL	LISTINGWIDTH	
			020E	C7 9F 0021C	PUSHAB	LIB_CRFBYVAL	
		69		06 FB 00220	CALLS	#6, -CRFSOUT	
03	0000G	CF		02 E0 00223	BBS	#2, LIB\$GL_CREFLAGS, 13\$	1544
			00FD	31 00229	BRW	14\$	
0104	C7	00F8	C7	08 28 0022C	MOV3	#8, FAO_MODESC, SYMBOLFAODESC	1546
0188	C7	00E8	C7	08 28 00234	MOV3	#8, FAO_SYMDESC, REFNCFAODESC	1547
		0103	C7	67 90 0023C	MOV3	MAXMODLNG, SYMBOLFIELD+3	1548
0187	C7	FC	A7	01 81 00241	ADDB3	#1, MAXSYMLNG, REFERENCER+3	1549
		70	A7	84 8F 9B 00248	MOVZBW	#132, CRFMDFSUBHDR	1550
		74	A7	FE74 CD 9E 0024D	MOVAB	MDFSUBHDR1, CRFMDFSUBHDR+4	1551
			0080	C7 9F 00253	PUSHAB	CRFMDFSUBPROTO1	1552
7E		67		01 C1 00257	ADDL3	#1, MAXMODLNG, -(SP)	1553
			00D8	C7 9F 00258	PUSHAB	FAO_EXCLAM	1552
			70	A7 9F 0025F	PUSHAB	CRFMDFSUBHDR	
			70	A7 9F 00262	PUSHAB	CRFMDFSUBHDR	
			00D0	C7 9F 00265	PUSHAB	FAO_CREFAO3	
		68		06 FB 00269	CALLS	#6, -SYSS\$FAO	
		56	78	A7 9E 0026C	MOVAB	CRFMDFSUBHDR+8, DESC	1554
		66	84	8F 9B 00270	MOVZBW	#132, (DESC)	1555
	04	A6	0114	CE 9E 00274	MOVAB	MDFSUBHDR2, 4(DESC)	1556
			0088	C7 9F 0027A	PUSHAB	CRFMDFSUBPROTO2	1557
7E		67		01 C1 0027E	ADDL3	#1, MAXMODLNG, -(SP)	1558
			00D8	C7 9F 00282	PUSHAB	FAO_EXCLAM	1557
				56 DD 00286	PUSHL	DESC	
				56 DD 00288	PUSHL	DESC	
			00D0	C7 9F 0028A	PUSHAB	FAO_CREFAO3	
		68		06 FB 0028E	CALLS	#6, -SYSS\$FAO	
	F8	A7	20	A7 9E 00291	MOVAB	CRFMDFHEADER, CURHEADER	1559
	F4	A7	70	A7 9E 00296	MOVAB	CRFMDFSUBHDR, CURSUBHDR	1560
		50	E0	A7 D0 0029B	MOVL	LINESPERPAGE, RO	1561
	DC	A7		50 D0 0029F	MOVL	RO, LINECOUNT	
		7E		01 7D 002A3	MOVQ	#1, -(SP)	1562

			50	DD	002A6	PUSHL	R0	:	1563
		FC	A0	9F	002A8	PUSHAB	-4(R0)	:	1562
		F0	A7	DD	002AB	PUSHL	LISTINGWIDTH	:	
		026E	C7	9F	002AE	PUSHAB	LIB_CRFMODDEF	:	
	0090	69	06	FB	002B2	CALLS	#6, CRFSOUT	:	
	0094	C7	84	8F	9B	002B5	MOVZBW	#132, CRFMRF SUBHDR	1565
			0090	CE	9E	002BB	MOVAB	MRFSUBHDR1, CRFMRF SUBHDR+4	1566
7E		67	00A0	C7	9F	002C2	PUSHAB	CRFMRF SUBPROTO1	1567
				01	C1	002C6	ADDL3	#1, MAXMODLNG, -(SP)	1568
			00D8	C7	9F	002CA	PUSHAB	FAO_EXCLAM	1567
			0090	C7	9F	002CE	PUSHAB	CRFMRF SUBHDR	
			0090	C7	9F	002D2	PUSHAB	CRFMRF SUBHDR	
			00D0	C7	9F	002D6	PUSHAB	FAO_CREFA03	
		68	06	FB	002DA	CALLS	#6, -SYSSFAO	:	
		56	0098	C7	9E	002DD	MOVAB	CRFMRF SUBHDR+8, DESC	1569
		66	84	8F	9B	002E2	MOVZBW	#132, (DESC)	1570
	04	A6	0C	AE	9E	002E6	MOVAB	MRFSUBHDR2, 4(DESC)	1571
			00A8	C7	9F	002EB	PUSHAB	CRFMRF SUBPROTO2	1572
7E		67		01	C1	002EF	ADDL3	#1, MAXMODLNG, -(SP)	1573
			00D8	C7	9F	002F3	PUSHAB	FAO_EXCLAM	1572
				56	DD	002F7	PUSHL	DESC	
				56	DD	002F9	PUSHL	DESC	
		68	00D0	C7	9F	002FB	PUSHAB	FAO_CREFA03	
			06	FB	002FF	CALLS	#6, -SYSSFAO	:	
	F8	A7	28	A7	9E	00302	MOVAB	CRFMRF HEADER, CURHEADER	1574
	F4	A7	0090	C7	9E	00307	MOVAB	CRFMRF SUBHDR, CURSUBHDR	1575
		50	E0	A7	D0	0030D	MOVL	LINESPERPAGE, R0	1576
	DC	A7		50	D0	00311	MOVL	R0, LINECOUNT	
		7E		01	7D	00315	MOVQ	#1, -(SP)	1577
				50	DD	00318	PUSHL	R0	1578
7E		50	DC	A7	C3	0031A	SUBL3	LINECOUNT, R0, -(SP)	1577
			F0	A7	DD	0031F	PUSHL	LISTINGWIDTH	
			02CE	C7	9F	00322	PUSHAB	LIB_CRFMODREF	
		69		06	FB	00326	CALLS	#6, -CRFSOUT	
				7E	D4	00329	CLRL	-(SP)	1580
			0000G	CF	DD	0032B	PUSHL	LIB\$GL_OUTFDB	
	0000G	CF		02	FB	0032F	CALLS	#2, LIB_CLOSE_OUT	
		50		01	D0	00334	MOVL	#1, R0	1581
				04	00337	RET		:	1582

: Routine Size: 824 bytes,      Routine Base: \$CODE\$ + 0000

LIB-  
V04=  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6  
6

: R  
:  
:

```

458 1583 1 ROUTINE cross_one_module (keydesc, keyrfa) =
459 1584 2 BEGIN
460 1585 2 ++
461 1586 2 |
462 1587 2 | This routine is called by the library procedures for each module
463 1588 2 | in the index that matches the current /only module name. The
464 1589 2 | module is read in its entirety and a cross reference is added to
465 1590 2 | the cross reference in progress.
466 1591 2 |
467 1592 2 | Inputs:
468 1593 2 |
469 1594 2 |     keydesc      pointer to string descriptor for module name
470 1595 2 |     keyrfa       pointer to rfa of module
471 1596 2 |
472 1597 2 | --
473 1598 2 |
474 1599 2 MAP
475 1600 2     keydesc : REF BBLOCK,
476 1601 2     keyrfa  : REF BBLOCK;
477 1602 2
478 1603 2 LOCAL
479 1604 2     status,
480 1605 2     bufdesc : BBLOCK [dsc$c_s_bln],
481 1606 2     recbuf  : BBLOCK [lbr$c_maxrecsiz];
482 1607 2
483 P 1608 2 perform (lib_get_zmem (mnb$c_fixedsize + .keydesc [dsc$w_length],
484 1609 2     curmod)); !Allocate module name block
485 1610 2 curmod [mnb$b_namlng] = .keydesc [dsc$w_length];
486 1611 2 IF .keydesc [dsc$w_length] GTR .maxmodlng !Check for new larger module name
487 1612 2 THEN maxmodlng = .keydesc [dsc$w_length];
488 1613 2 CHSMOVE (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer], !Copy module name
489 1614 2     curmod [mnb$t_name]);
490 1615 2 |
491 1616 2 | FIND the module so we can read the text
492 1617 2 |
493 P 1618 2 rms_perform (lbr$find (lib$gl_libctl, .keyrfa), !Point to the module
494 1619 2     lib$lookupperr, .lbr$gl_rmsstvt, 1, lib$gl_libfdb [fdb$l_namdesc]);
495 1620 2 bufdesc [dsc$w_length] = lbr$c_maxrecsiz;
496 1621 2 bufdesc [dsc$a_pointer] = recbuf;
497 1622 2 recdesc [dsc$a_pointer] = recbuf; !**temp
498 1623 2 |
499 1624 2 | Read the module and dispatch on each record
500 1625 2 |
501 1626 2 WHILE (status = lbr$get_record (lib$gl_libctl, bufdesc, recdesc)) NEQ rms$_eof
502 1627 2 DO perform ((.recdispatch [.objrec [obj$b_rectyp]]) ());
503 1628 2 found1 - true;
504 1629 2 RETURN true
505 1630 1 END; !Of cross_one_module

```

007C 0000 CROSS\_ONE MODULE:

56	0000'	CF	9E	0002	WORD	Save R2,R3,R4,R5,R6	:	1583
5E	F7F8	CE	9E	0007	MOVAB	CURMOD, R6	:	
					MOVAB	-2056(SP), SP	:	

				56	DD	0000C		PUSHL	R6		1609
		52		AC	DO	0000E		MOVL	KEYDESC, R2		
		7E	04	62	3C	00012		MOVZWL	(R2), -(SP)		
		6E		09	CO	00015		ADDL2	#9, (SP)		
		0000G		02	FB	00018		CALLS	#2, LIB_GET_ZMEM		
		CF		50	E8	0001D		BLBS	STATUS, -1\$		
		01			04	00020		RET			
		50		66	DO	00021	1\$:	MOVL	CURMOD, R0		1610
		08		62	90	00024		MOVW	(R2), 8(R0)		
2C	A6	62		00	ED	00028		CMPZV	#0, #16, (R2), MAXMODLNG		1611
		10		04	15	0002E		BLEQ	2\$		
		2C	A6	62	3C	00030		MOVZWL	(R2), MAXMODLNG		1612
		04	B2	62	28	00034	2\$:	MOVW	(R2), @4(R2), 9(R0)		1614
09	A0	08		AC	DD	0003A		PUSHL	KEYRFA		1619
		00000000G		CF	9F	0003D		PUSHAB	LIB\$GL_LIBCTL		
		1D		02	FB	00041		CALLS	#2, LBR\$FIND		
		00000000G		50	E8	00048		BLBS	STATUS, 3\$		
				00	DD	0004B		PUSHL	LBR\$GL_RMSSTV		
		7E		50	DD	00051		PUSHL	STATUS		
		0000G	CF	10	C1	00053		ADDL3	#16, LIB\$GL_LIBFDB, -(SP)		
				01	DD	00059		PUSHL	#1		
		00000000G		8F	DD	0005B		PUSHL	#LIB\$ LOOKUPERR		
		F8	AD	05	FB	00061		CALLS	#5, LIB\$SIGNAL		
		FC	AD	8F	B0	00068	3\$:	MOVW	#2048, BUFDESC		1620
		F8	A6	6E	9E	0006E		MOVAB	RECBUF, BUFDESC+4		1621
				6E	9E	00072		MOVAB	RECBUF, RECDESC+4		1622
		F4	A6	9F	00076	4\$:		PUSHAB	RECDESC		1626
		F8	AD	9F	00079			PUSHAB	BUFDESC		
		00000000G		CF	9F	0007C		PUSHAB	LIB\$GL_LIBCTL		
		0001827A	8F	03	FB	00080		CALLS	#3, LBR\$GET_RECORD		
				50	DO	00087		MOVL	R0, STATUS		
				52	D1	0008A		CPL	STATUS, #98938		
				11	13	00091		BEQ	5\$		
				50	F8	B6	9A	00093	MOVZBL	@OBJREC, R0	1627
				50	0000	CF40	DO	00097	MOVL	RECDISPATCH[R0], R0	
				60	00	FB	0009D	CALLS	#0, (R0)		
				D3	50	E8	000A0	BLBS	STATUS, 4\$		
					04	000A3		RET			
		FC	A6	01	DO	000A4	5\$:	MOVL	#1, FOUND1		1628
			50	01	DO	000A8		MOVL	#1, R0		1629
				04	000AB			RET			1630

; Routine Size: 172 bytes, Routine Base: \$CODE\$ + 0338

```

: 507      1631 1 ROUTINE prohdr =
: 508      1632 2 BEGIN
: 509      1633 2 !
: 510      1634 2 BIND
: 511      1635 2      modidstring = objrec [mhd$t_name] + .objrec [mhd$b_namlng] : VECTOR [,BYTE];
: 512      1636 2 !
: 513      1637 2
: 514      1638 2 RETURN true
: 515      1639 1 END;          ! OF prohdr

```

```

          50          0000 00000 PROHDR: .WORD  Save nothing      : 1631
          01  D0 00002      MOVL  #1, R0      : 1638
          04 00005      RET                    : 1639

```

: Routine Size: 6 bytes, Routine Base: \$CODE\$ + 03E4



```

1640 1 ROUTINE progsd =
1641 2 BEGIN
1642 2
1643 2 ++
1644 2 Verify GSD records and dispatch on the four sub-types:
1645 2 (0) P-SECTION definition
1646 2 (1) Symbol definition/reference
1647 2 (2) Entry point definition
1648 2 (3) Procedure declaration
1649 2 --
1650 2 BIND
1651 2 gsddispatch = PLIT (
1652 2     propsectdef,
1653 2     symbols,
1654 2     entpnts,
1655 2     procedef,
1656 2     symbols,
1657 2     pro_epmw,
1658 2     procedef,
1659 2     pro_idc,
1660 2     pro_env,
1661 2     pro_lsy,
1662 2     pro_lepm,
1663 2     pro_lpro,
1664 2     pro_spsc) : VECTOR;
1665 2
1666 2 gsdoffset = obj$c_subtyp;
1667 2
1668 2 WHILE .gsdoffset LSSU .reclng DO
1669 2     perform (( .gsddispatch [.objvec [.gsdoffset]]) ());
1670 2 RETURN true
1671 1 END;          ! Of progsd

```

```

.PSECT $PLITS,NOWRT,NOEXE,2
.LONG 13
P.ABH: .ADDRESS PROPSECTDEF, SYMBOLS, ENTPNTS, PROCEDEF, -
        SYMBOLS, PRO_EPMW, PROCEDEF, PRO_IDC, -
        PRO_ENV, PRO_LSY, PRO_LEPM, PRO_LPRO, -
        PRO_SPSC

```

GSDDISPATCH= P.ABH

```

.PSECT $CODE$,NOWRT,2
PROGSD: .WORD Save R2          : 1640
        MOVAB GSDOFFSET, R2   :
        MOVL #1, GSDOFFSET    : 1666
        CMPZV #0, #16, RECLNG, GSDOFFSET : 1668
        BLEQU 2$              :
        ADDL3 GSDOFFSET, OBJVEC, R0 : 1669
        MOVZBL (R0), R0
        MOVL GSDDISPATCH(R0), R0
        CALLS #0, (R0)

```

LIB\_CROSS  
V04-000

F 12  
16-Sep-1984 01:48:05  
14-Sep-1984 12:37:58

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[LIBRAR.SRC]CROSS.B32;1  
Page 24  
(8)

E4	50	E8 00023	BLBS	STATUS, 1\$
		04 00026	RET	
50	01	D0 00027 2\$:	MOVL	#1, R0
		04 0002A	RET	

.....  
: 1670  
: 1671

: Routine Size: 43 bytes, Routine Base: \$CODE\$ + 03EA

LIB  
V04

: R

: (

```

: 550      1672 1 ROUTINE pronul =
: 551      1673 2 BEGIN
: 552      1674 2 |
: 553      1675 2 | Null routine
: 554      1676 2 |
: 555      1677 2 RETURN true
: 556      1678 1 END;

```

```

                    50          0000 00000 PRONUL: .WORD Save nothing      : 1672
                                01  D0 00002      MOVL #1, R0                : 1677
                                04 00005      RET                          : 1678

```

: Routine Size: 6 bytes, Routine Base: \$CODES + 0415

: 557 1679 1 %SBTTL 'propsectdef';

propsectdef

```

: 559 1680 1 ROUTINE propsectdef =
: 560 1681 2 BEGIN
: 561 1682 3
: 562 1683 4 ++
: 563 1684 5 process P-section definitions
: 564 1685 6
: 565 1686 7 --
: 566 1687 8
: 567 1688 9 BIND
: 568 1689 10 psctdef = objvec [.gsdoffset] : BBLOCK;
: 569 1690 11 LOCAL
: 570 1691 12 length;
: 571 1692 13
: 572 1693 14 length = $BYTEOFFSET(gps$t_name) - $BYTEOFFSET(gps$t_start) + ! Compute the offset of next GSD
: 573 1694 15 .psctdef [gps$b_namlng]; ! From length of this
: 574 1695 16 gsdoffset = .gsdoffset + .length;
: 575 1696 17 RETURN true
: 576 1697 18 END; ! Of propsectdef

```

0000 0000 PROPSECTDEF:

							.WORD	Save nothing	: 1680
50	0000'	CF	0000'	CF	C1	00002	ADDL3	GSDOFFSET, OBJVEC, R0	: 1689
		50	08	A0	9A	0000A	MOVZBL	8(R0), LENGTH	: 1693
		50		09	C0	0000E	ADDL2	#9, LENGTH	: 1695
	0000'	CF		50	C0	00011	ADDL2	LENGTH, GSDOFFSET	: 1696
		50		01	D0	00016	MOVL	#1, R0	: 1697
				04	00019		RET		

; Routine Size: 26 bytes, Routine Base: \$CODE\$ + 041B

symbols

```

578 1698 1 %SBTTL 'symbols';
579 1699 1
580 1700 1 ROUTINE symbols =
581 1701 2 BEGIN
582 1702 2
583 1703 2 LOCAL
584 1704 2 length;
585 1705 2 BIND
586 1706 2 symbolrec = objvec [.gsdoffset] : BBLOCK;
587 1707 2
588 1708 2
589 1709 2 IF NOT .symbolrec [gsy$v_def]
590 1710 2 THEN BEGIN
591 1711 2 length = $BYTEOFFSET(srf$t_name) - $BYTEOFFSET(srf$t_start) +
592 1712 2 .symbolrec [srf$b_namlng];
593 1713 2 symbolstring = symbolrec [srf$b_namlng]; ! Point to the symbol string
594 1714 2 symbolval = a_zero;
595 1715 2 END
596 1716 2
597 1717 2 ELSE
598 1718 2 BEGIN
599 1719 2 IF .objvec [.gsdoffset] EQL obj$c_gsd_symw ! If word psect
600 1720 2 THEN
601 1721 2 BEGIN
602 1722 2 length = $BYTEOFFSET(sdfw$t_name) - $BYTEOFFSET(sdfw$t_start) +
603 1723 2 .symbolrec [sdfw$b_namlng];
604 1724 2 symbolstring = symbolrec [sdfw$b_namlng]; ! Point to the symbol
605 1725 2 symbolval = symbolrec[sdfw$t_value];
606 1726 2 END
607 1727 2 ELSE
608 1728 2 BEGIN
609 1729 2 length = $BYTEOFFSET(sdf$t_name) - $BYTEOFFSET(sdf$t_start) +
610 1730 2 .symbolrec [sdf$b_namlng];
611 1731 2 symbolstring = symbolrec [sdf$b_namlng]; ! Point to the symbol
612 1732 2 symbolval = symbolrec[sdf$t_value];
613 1733 2 END;
614 1734 2 END;
615 1735 2 perform (prosymbol ());
616 1736 2 gsdoffset = .gsdoffset + .length; ! Update the gsd offset for next
617 1737 2 RETURN true
618 1738 1 END; !Of symbols

```

				000C	00000	SYMBOLS:	.WORD	Save R2,R3	1700	
		53	0000'	CF	9E	00002	MOVAB	SYMBOLSTRING, R3	1706	
50	08	A3	FC	A3	C1	00007	ADDL3	GSDOFFSET, OBJVEC, R0	1709	
12	02	A0		01	E0	0000D	BBS	#1, 2(R0), 1\$	1711	
		52	04	A0	9A	00012	MOVZBL	4(R0), LENGTH	1713	
		52		05	C0	00016	ADDL2	#5, LENGTH	1714	
		63	04	A0	9E	00019	MOVAB	4(R0), SYMBOLSTRING	1709	
	14	A3	40	A3	9E	0001D	MOVAB	A ZERO, SYMBOLVAL	1719	
		04		27	11	00022	BRB	3\$		
				60	91	00024	1\$:	CMPB	(R0), #4	
				12	12	00027	BNEQ	2\$		

symbols

	52	0A	A0	9A	00029	MOVZBL	10(R0), LENGTH	:	1722	
	52		0B	C0	0002D	ADDL2	#11, LENGTH	:		
	63	0A	A0	9E	00030	MOVAB	10(R0), SYMBOLSTRING	:	1724	
14	A3	06	A0	9E	00034	MOVAB	6(R0), SYMBOLVAL	:	1725	
			10	11	00039	BRB	3\$	:	1719	
	52	09	A0	9A	0003B	2\$:	MOVZBL	9(R0), LENGTH	:	1729
	52		0A	C0	0003F	ADDL2	#10, LENGTH	:		
	63	09	A0	9E	00042	MOVAB	9(R0), SYMBOLSTRING	:	1731	
14	A3	05	A0	9E	00046	MOVAB	5(R0), SYMBOLVAL	:	1732	
0000V	CF		00	FB	0004B	3\$:	CALLS	#0, PROSYMBOL	:	1735
	07		50	E9	00050	BLBC	STATUS, 4\$	:		
FC	A3		52	C0	00053	ADDL2	LENGTH, GSDOFFSET	:	1736	
	50		01	D0	00057	MOVL	#1, R0	:	1737	
			04	0005A	4\$:	RET		:	1738	

; Routine Size: 91 bytes, Routine Base: \$CODE\$ + 0435

entpnts

```

: 620 1739 1 %SBTTL 'entpnts';
: 621 1740 1
: 622 1741 1 ROUTINE entpnts =
: 623 1742 2 BEGIN
: 624 1743 2 !
: 625 1744 2 LOCAL
: 626 1745 2 length;
: 627 1746 2 BIND
: 628 1747 2 symbolrec = objvec [.gsdoffset] : BBLOCK;
: 629 1748 2
: 630 1749 2 length = $BYTEOFFSET(epm$t_name) - $BYTEOFFSET(epm$t_start) +
: 631 1750 2 .symbolrec [epm$b_namlng];
: 632 1751 2 symbolstring = symbolrec [epm$b_namlng]; ! Point to the symbol
: 633 1752 2 symbolval = symbolrec[epm$l_addr];
: 634 1753 2 perform (prosymbol ());
: 635 1754 2 gsdoffset = .gsdoffset + .length; ! Else update the offset for next
: 636 1755 2 RETURN true
: 637 1756 1 END; ! Of entpnts

```

				000C	0000	ENTPNTS:	.WORD	Save R2,R3	: 1741
		53	0000'	CF	9E	00002	MOVAB	GSDOFFSET, R3	: 1747
50	0C	A3		63	C1	00007	ADDL3	GSDOFFSET, OBJVEC, R0	: 1749
		52	0B	A0	9A	0000C	MOVZBL	11(R0), LENGTH	: 1751
		52	0C	C0	00010		ADDL2	#12, LENGTH	: 1752
	04	A3	0B	A0	9E	00013	MOVAB	11(R0), SYMBOLSTRING	: 1753
	18	A3	05	A0	9E	00018	MOVAB	5(R0), SYMBOLVAL	: 1754
	C000V	CF		00	FB	0001D	CALLS	#0, PROSYMBOL	: 1755
		06		50	E9	00022	BLBC	STATUS, 1\$	: 1756
		63		52	C0	00025	ADDL2	LENGTH, GSDOFFSET	: 1755
		50		01	D0	00028	MOVL	#1, R0	: 1756
				04	0002B	1\$:	RET		

; Routine Size: 44 bytes, Routine Base: \$CODE\$ + 0490

procedef

```

: 639 1757 1 %SBTTL 'procedef';
: 640 1758 1
: 641 1759 1 ROUTINE procedef =
: 642 1760 2 BEGIN
: 643 1761 2
: 644 1762 2 A procedure definition is an extended entry point definition, carrying with
: 645 1763 2 it a description of the procedure's formal arguments. processing these consists
: 646 1764 2 in normal symbol definition processing followed by:-
: 647 1765 2 (1) Validation of the format of formal description (i.e. just check
: 648 1766 2 that minimum number of arguments specified is less than
: 649 1767 2 or equal to the maximum.
: 650 1768 2
: 651 1769 2
: 652 1770 2 LOCAL
: 653 1771 2 argcount;
: 654 1772 2
: 655 1773 2 IF .objvec [.gsdoffset] EQL obj$c_gsd_prow
: 656 1774 2 THEN
: 657 1775 3 perform (pro_epmw ())
: 658 1776 2 ELSE
: 659 1777 2 perform (entpnts ());
: 660 1778 2
: 661 1779 3 BEGIN
: 662 1780 3 BIND
: 663 1781 3 formals = objvec [.gsdoffset] : BBLOCK;
: 664 1782 3 gsdoffset = .gsdoffset + fml$c_size;
: 665 1783 3 IF (argcount = .formals [fml$b_maxargs]) NEQ 0 ! Update record pointer
: 666 1784 3 THEN INCRU i FROM 1 TO .argcount ! If args
: 667 1785 4 DO BEGIN ! then skip them
: 668 1786 4 BIND
: 669 1787 4 argdesc = objvec [.gsdoffset] : BBLOCK;
: 670 1788 4
: 671 1789 4 gsdoffset = .gsdoffset + .argdesc [arg$b_bytecnt] + arg$c_size;
: 672 1790 3 END;
: 673 1791 3 RETURN true
: 674 1792 2 END;
: 675 1793 1 END;

```

! Of procedef

000C 0000 PROCEDEF:

		53	0000'	CF 9E 00002	.WORD	Save R2,R3	: 1759
50	0C	A3		63 C1 00007	MOVAB	GSDOFFSET, R3	: 1773
		06		60 91 0000C	ADDL3	GSDOFFSET, OBJVEC, R0	
				07 12 0000F	CMPB	(R0), #6	
	0000V	CF		00 FB 00011	BNEQ	1\$	: 1775
				04 11 00016	CALLS	#0, PRO_EPMW	
	B8	AF		00 FB 00018	BRB	2\$	: 1777
		2D		50 E9 0001C	CALLS	#0, ENTPNTS	
50	0C	A3		63 C1 0001F	BLBC	STATUS, 6\$	: 1781
		63		02 C0 00024	ADDL3	GSDOFFSET, OBJVEC, R0	: 1782
		52	01	A0 9A 00027	ADDL2	#2, GSDOFFSET	: 1783
				1C 13 0002B	MOVZBL	1(R0), ARGCOUNT	
		51		01 D0 0002D	BEQL	5\$	: 1784
					MOVL	#1, I	



procedef

50	0C	A3	12	11	00030		BRB	4\$		
		50	63	C1	00032	3\$:	ADDL3	GSDOFFSET, OBJVEC, R0	:	1787
		50		A0	9A	00037	MOVZBL	1(R0), R0	:	1789
		63	63	C0	0003B		ADDL2	GSDOFFSET, R0	:	
				A0	9E	0003E	MOVAB	2(R0), GSDOFFSET	:	
		52	51	D6	00042		INCL	1	:	1784
			51	D1	00044	4\$:	CMPL	1, ARGCOUNT	:	
		50	E9	1B	00047		BLEQU	3\$	:	
			01	D0	00049	5\$:	MOVL	#1, R0	:	1791
			04	0004C	6\$:		RET		:	1793

: Routine Size: 77 bytes, Routine Base: \$CODE\$ + 04BC

.....

pro\_epmw

```

: 677      1794 1 %SBTTL 'pro_epmw';
: 678      1795 1
: 679      1796 1 ROUTINE pro_epmw =
: 680      1797 2 BEGIN
: 681      1798 2
: 682      1799 2          Process entry points with word psect
: 683      1800 2
: 684      1801 2 LOCAL
: 685      1802 2          length;
: 686      1803 2 BIND
: 687      1804 2          symbolrec = objvec [.gsdoffset] : BBLOCK;
: 688      1805 2
: 689      1806 2
: 690      1807 2          length = $BYTEOFFSET(epmw$t_name) - $BYTEOFFSET(epmw$t_start) +
: 691      1808 2                      .symbolrec [epmw$b_namlng];
: 692      1809 2          symbolstring = symbolrec [epmw$b_namlng];          ! Point to the symbol
: 693      1810 2          symbolval = symbolrec[epmw$l_addr$];
: 694      1811 2          perform (prosymbol ());
: 695      1812 2          gsdoffset = .gsdoffset + .length;          ! Else update the offset for next
: 696      1813 2          RETURN true
: 697      1814 1          END;          ! Of pro_epmw

```

000C 0000 PRO\_EPMW:

		53	0000'	CF	9E	00002	.WORD	Save R2,R3	: 1796
50	0C	A3		63	C1	00007	MOVAB	GSDOFFSET, R3	: 1804
		52	0C	A0	9A	0000C	ADDL3	GSDOFFSET, OBJVEC, R0	: 1807
		52		0D	C0	00010	MOVZBL	12(R0), LENGTH	
	04	A3	C	A0	9E	00013	ADDL2	#13, LENGTH	: 1809
	18	A3	06	A0	9E	00018	MOVAB	12(R0), SYMBOLSTRING	: 1810
	0000V	CF		00	FB	0001D	MOVAB	6(R0), SYMBOLVAL	: 1811
		06		50	E9	00022	CALLS	#0, PROSYMBOL	
		63		52	C0	00025	BLBC	STATUS, 1\$	: 1812
		50		01	D0	00028	ADDL2	LENGTH, GSDOFFSET	: 1813
				04	0002B	1\$:	MOVL	#1, R0	: 1814
							RET		

: Routine Size: 44 bytes, Routine Base: \$CODE\$ + 0509

: 698 1815 1

```

: 700      1816 1 %SBTTL 'pro_idc';
: 701      1817 1
: 702      1818 1 ROUTINE pro_idc =
: 703      1819 2 BEGIN
: 704      1820 2
: 705      1821 2 |
: 706      1822 2 |         Process random entity check
: 707      1823 2 |         by skipping it.
: 708      1824 2 |
: 709      1825 2 LOCAL
: 710      1826 2     identstring : REF VECTOR [,BYTE],    ! pointer to ident string
: 711      1827 2     objectname : REF VECTOR [,BYTE],   ! pointer to object name string
: 712      1828 2     length;
: 713      1829 2 BIND
: 714      1830 2     idc_rec = objvec [.gsdoffset] : BBLOCK;
: 715      1831 2     identstring = idc_rec [idc$b_namlng] + 1 + .idc_rec [idc$b_namlng];
: 716      1832 2     objectname = identstring [1] + .identstring [0];
: 717      1833 2     length = objectname [1] + .objectname [0] - idc_rec;
: 718      1834 2     gsdoffset = .gsdoffset + .length;
: 719      1835 2 RETURN true
: 720      1836 1 END;

```

! Of pro\_idc

52	0000'	CF	0000'	CF	C1	00002	PRO_IDC: .WORD	Save R2	: 1818
		50	03	A2	9A	0000A	ADDL3	GSDOFFSET, OBJVEC, R2	: 1829
		50	04	A042	9E	0000E	MOVZBL	3(R2), R0	: 1831
		51		60	9A	00013	MOVAB	4(R0)[R2], IDENTSTRING	
		50	01	A140	9E	00016	MOVZBL	(IDENTSTRING), R1	: 1832
		51		60	9A	0001B	MOVAB	1(R1)[IDENTSTRING], OBJECTNAME	
		50		51	C0	0001E	MOVZBL	(OBJECTNAME), R1	: 1833
		50		52	C2	00021	ADDL2	R1, OBJECTNAME	
				50	D6	00024	SUBL2	R2, R0	
	0000'	CF		50	C0	00026	INCL	LENGTH	: 1834
		50		01	D0	0002B	ADDL2	LENGTH, GSDOFFSET	: 1835
					04	0002E	MOVL	#1, R0	: 1836
							RET		

: Routine Size: 47 bytes, Routine Base: \$CODE\$ + 0535

: 721 1837 1

pro\_env

```

: 723 1838 1 %SBTTL 'pro_env';
: 724 1839 1
: 725 1840 1 ROUTINE pro_env =
: 726 1841 2 BEGIN
: 727 1842 2
: 728 1843 2 | Process environment definition
: 729 1844 2 | by skipping it.
: 730 1845 2 |
: 731 1846 2 LOCAL
: 732 1847 2 length;
: 733 1848 2 BIND
: 734 1849 2 env_rec = objvec [.gsdoffset] : BBLOCK;
: 735 1850 2
: 736 1851 2
: 737 1852 2 length = env_rec [env$t_name] - objvec [.gsdoffset] +
: 738 1853 2 | .env_rec [env$b_namlng];
: 739 1854 2 gsdoffset = .gsdoffset + .length;
: 740 1855 2 RETURN true
: 741 1856 1 END;
! Of pro_env

```

				0004	0000	PRO_ENV: .WORD	Save R2		: 1840
50	0000'	CF	0000'	CF	C1 00002	ADDL3	GSDOFFSET, OBJVEC, R0		: 1849
51		50		50	C3 0000A	SUBL3	R0, R0, R1		: 1852
		52	05	A0	9A 0000E	MOVZBL	5(R0), R2		: 1853
		51		52	C0 00012	ADDL2	R2, R1		: 1852
		50	06	A1	9E 00015	MOVAB	6(R1), LENGTH		: 1854
	0000'	CF		50	C0 00019	ADDL2	LENGTH, GSDOFFSET		: 1855
		50		01	D0 0001E	MOVL	#1, R0		: 1856
				04	00021	RET			

; Routine Size: 34 bytes, Routine Base: \$CODE\$ + 0564

; 742 1857 1

S  
R  
L  
M  
C

pro\_lsy

```

: 744 1858 1 %SBTTL 'pro_lsy':
: 745 1859 1
: 746 1860 1 ROUTINE pro_lsy =
: 747 1861 2 BEGIN
: 748 1862 2
: 749 1863 2 | Process local symbol definition/reference
: 750 1864 2 | by skipping it.
: 751 1865 2 |
: 752 1866 2 LOCAL
: 753 1867 2 length;
: 754 1868 2 BIND
: 755 1869 2 lsy_rec = objvec [.gsdoffset] : BBLOCK;
: 756 1870 2
: 757 1871 2 IF NOT .lsy_rec [lsy$v_def]
: 758 1872 2 THEN
: 759 1873 2 length = $BYTEOFFSET(lsrfs_name) - $BYTEOFFSET(lsrfs_start) +
: 760 1874 2 .lsy_rec [lsrfs_b_namlng]
: 761 1875 2 ELSE
: 762 1876 2 length = $BYTEOFFSET(lsdofs_name) - $BYTEOFFSET(lsdofs_start) +
: 763 1877 2 .lsy_rec [lsdofs_b_namlng];
: 764 1878 2 gsdoffset = .gsdoffset + .length;
: 765 1879 2 RETURN true
: 766 1880 1 END;

```

! Of pro\_lsy

				0000	00000	PRO_LSY: .WORD	Save nothing		: 1860
50	0000'	CF	0000'	CF	C1 00002	ADDL3	GSDOFFSET, OBJVEC, R0		: 1869
09	02	A0		01	E0 0000A	BBS	#1, 2(R0), 1\$		: 1871
		50	06	A0	9A 0000F	MOVZBL	6(R0), LENGTH		: 1873
		50		07	C0 00013	ADDL2	#7, LENGTH		: 1874
				07	11 00016	BRB	2\$		: 1875
		50	0C	A0	9A 00018	1\$: MOVZBL	12(R0), LENGTH		: 1876
		50		0D	C0 0001C	ADDL2	#13, LENGTH		: 1877
	0000'	CF		50	C0 0001F	2\$: ADDL2	LENGTH, GSDOFFSET		: 1878
		50		01	D0 00024	MOVL	#1, R0		: 1879
				04	00027	RET			: 1880

; Routine Size: 40 bytes, Routine Base: \$CODE\$ + 0586

; 767 1881 1

pro\_lepm

```

: 769      1882 1 %SBTTL 'pro_lepm';
: 770      1883 1
: 771      1884 1 ROUTINE pro_lepm =
: 772      1885 2 BEGIN
: 773      1886 2
: 774      1887 2 |
: 775      1888 2 | Process local symbol entry point definition
: 776      1889 2 | by skipping it.
: 777      1890 2 LOCAL
: 778      1891 2 length;
: 779      1892 2 BIND
: 780      1893 2 lepm_rec = objvec [.gsdoffset] : BBLOCK;
: 781      1894 2
: 782      1895 2
: 783      1896 2 length = $BYTEOFFSET(lepm$t_name) - $BYTEOFFSET(lepm$t_start) +
: 784      1897 2 | .lepm_rec [lepm$b_namlng];
: 785      1898 2 gsdoffset = .gsdoffset + .length; ! Else update the offset for next
: 786      1899 2 RETURN true
: 787      1900 1 END; ! Of pro_lepm

```

0000 0000 PRO\_LEPM:

50	0000'	CF	0000'	CF	C1	00002	.WORD	Save nothing	: 1884
		50	OE	A0	9A	0000A	ADDL3	GSDOFFSET, OBJVEC, R0	: 1893
		50		0F	C0	0000E	MOVZBL	14(R0), LENGTH	: 1896
	0000'	CF		50	C0	00011	ADDL2	#15, LENGTH	: 1898
		50		01	D0	00016	ADDL2	LENGTH, GSDOFFSET	: 1899
				04	00	0019	MOVL	#1, R0	: 1900
							RET		

: Routine Size: 26 bytes, Routine Base: \$CODE\$ + 05AE

: 788 1901 1

pro\_lpro

```

: 790      1902 1 %SBTTL 'pro_lpro';
: 791      1903 1
: 792      1904 1 ROUTINE pro_lpro =
: 793      1905 2 BEGIN
: 794      1906 2
: 795      1907 2 |
: 796      1908 2 |           Process local symbol procedure definition
: 797      1909 2 |           by skipping it.
: 798      1910 2 |
: 799      1911 2 LOCAL
: 800      1912 2 length;
: 801      1913 2 BIND
: 802      1914 2 lpro_rec = objvec [.gsdoffset] : BBLOCK;
: 803      1915 2
: 804      1916 2 length = $BYTEOFFSET(lpro$t_name) - $BYTEOFFSET(lpro$t_start) +
: 805      1917 2 |           .lpro_rec [lpro$b_namlng];
: 806      1918 2 gsdoffset = .gsdoffset + ..length;           ! Else update the offset for next
: 807      1919 2 RETURN true
: 808      1920 1 END;

```

! Of pro\_lpro

0000 00000 PRO\_LPRO:

50	0000'	CF	0000'	CF	C1	00002	.WORD	Save nothing	:	1904
		50	0E	A0	9A	0000A	ADDL3	GSDOFFSET, OBJVEC, R0	:	1913
		50		0F	C0	0000E	MOVZBL	14(R0), LENGTH	:	1916
	0000'	CF		50	C0	00011	ADDL2	#15, LENGTH	:	1918
		50		01	D0	00016	ADDL2	LENGTH, GSDOFFSET	:	1919
				04	00019		MOVL	#1, R0	:	1920
							RET		:	

: Routine Size: 26 bytes, Routine Base: \$CODE\$ + 05C8

: 809 1921 1

pro\_spsc

```

: 811 1922 1 %SBTTL 'pro_spsc';
: 812 1923 1
: 813 1924 1 ROUTINE pro_spsc =
: 814 1925 2 BEGIN
: 815 1926 2
: 816 1927 2 Process shareable image psect definition
: 817 1928 2 by ignoring it.
: 818 1929 2
: 819 1930 2 LOCAL
: 820 1931 2 length;
: 821 1932 2 BIND
: 822 1933 2 spsct_def = objvec [.gsdoffset] : BBLOCK;
: 823 1934 2
: 824 1935 2 length = $BYTEOFFSET(sgps$t_name) - $BYTEOFFSET(sgps$t_start) +
: 825 1936 2 .spsct_def [sgps$b_namlng];
: 826 1937 2 gsdoffset = .gsdoffset + .length;
: 827 1938 2 RETURN true
: 828 1939 1 END;

```

! Of pro\_spsc

0000 0000 PRO\_SPSC:

50	0000'	CF	0000'	CF	C1	00002	.WORD	Save nothing	:	1924
		50	0C	A0	9A	0000A	ADDL3	GSDOFFSET, OBJVEC, R0	:	1933
		50		0D	C0	0000E	MOVZBL	12(R0), LENGTH	:	1935
	0000'	CF		50	C0	00011	ADDL2	#13, LENGTH	:	1937
		50		01	D0	00016	ADDL2	LENGTH, GSDOFFSET	:	1938
				04	00019		MOVL	#1, R0	:	1939
							RET		:	

: Routine Size: 26 bytes. Routine Base: \$CODE\$ + 05E2

: 829 1940 1



pro\_spssc

```
831 1941 1 ROUTINE prosymbol =
832 1942 2 BEGIN
833 1943 2
834 1944 2 !++
835 1945 2 ! This routine does all the work of processing a symbol
836 1946 2 !--
837 1947 2 BIND
838 1948 2     symbolrec = objvec [.gsdoffset] : BBLOCK;
839 1949 2
840 1950 2 LOCAL
841 1951 2     symdesc : REF BBLOCK,
842 1952 2     weakflag,
843 1953 2     definition;
844 1954 2
845 1955 2 IF .symbolstring [0] GTR .maxsymlng
846 1956 2     THEN maxsymlng = .symbolstring [0];
847 1957 2 perform (lib_get_zmem (snb$c fixedsize + .symbolstring [0], symdesc));
848 1958 2 symdesc [snb$w_flags] = .symbolrec [gsy$w_flags];
849 1959 2 symdesc [snb$b_namlng] = .symbolstring [0];
850 1960 2 CH$MOVE (.symbolstring [0], symbolstring [1], symdesc [snb$t_name]);
851 1961 2 symdesc [snb$l_value] = ..symbolval;
852 1962 2 weakflag = .symbolrec [gsy$v_weak];
853 1963 2 definition = false;
854 1964 2 IF .symbolrec [gsy$v_def]
855 1965 2     THEN definition = true;
856 1966 2 IF .lib$gl_creflags [lib$v_crfbysym]           !If cross ref by symbols
857 1967 2 THEN BEGIN
858 1968 2     IF .definition
859 1969 3 P THEN perform (lib$crf_ins_key (lib_crf$symval, symdesc[snb$b_namlng], !Insert key definition if a definit
860 1970 3     symdesc [snb$l_value], symdesc [snb$w_flags]));
861 1971 3     THEN perform (crf$insrtkey (lib_crf$symval, symdesc [snb$b_namlng],
862 1972 3     symdesc [snb$l_value], .symdesc [snb$w_flags]));
863 1973 3
864 1974 3 ! Insert reference
865 1975 3
866 1976 3 P perform (lib$crf_ins_ref (lib_crf$symval, symdesc [snb$b_namlng],
867 1977 3     curmod [mnb$b_namlng], weakflag, definition));
868 1978 3 !perform (crf$insrtref (lib_crf$symval, symdesc [snb$b_namlng],
869 1979 3     curmod [mnb$b_namlng], .weakflag, .definition));
870 1980 2 END;           !Of cref by symbols
871 1981 2
872 1982 2 ! If absolute definition, insert reference to value into cross reference by value table
873 1983 2
874 1984 2 IF .definition AND NOT .symbolrec [gsy$v_rel]
875 1985 2     AND .lib$gl_creflags [lib$v_crfbyval]
876 1986 2 P THEN perform (lib$crf_ins_ref (lib_crf$byval, symdesc [snb$l_value], .symbolstring,
877 1987 2     symdesc [snb$w_flags], a_zero));
878 1988 2     THEN perform (crf$insrtref (lib_crf$byval, symdesc [snb$l_value], .symbolstring,
879 1989 2     .symdesc [snb$w_flags], .a_zero));
880 1990 2
881 1991 2 ! If cross referencing by modules, insert into the proper table
882 1992 2
883 1993 2 IF .lib$gl_creflags [lib$v_crfbymod]
884 1994 2 P THEN perform (lib$crf_ins_ref ((IF .definition THEN lib_crf$moddef
885 1995 2     ELSE lib_crf$modref),
886 1996 2     -urmod [mnb$b_namlng], symdesc [snb$b_namlng],
887 1997 2     symdesc [snb$w_flags], a_zero));
```

pro\_spsc

```

: 888 1998 2 ! THEN perform (crf$insrtref ((IF .definition THEN lib_crfmoddef
: 889 1999 2 ! ; ELSE lib_crfmodref),
: 890 2000 2 ! ; curmod [mnb$b_namlng], symdesc [snb$b_namlng],
: 891 2001 2 ! ; .symdesc [snb$w_flags], .a_zero));
: 892 2002 2 !
: 893 2003 2 RETURN true
: 894 2004 1 END;

```

! Of symbol

07FC 0000 PROSYMBOL:

Address	OpCode	Operand	Symbol	Year
	WORD		Save R2,R3,R4,R5,R6,R7,R8,R9,R10	1941
	MOVAB		LIB\$GL_CREFLAGS, R10	
	MOVAB		LIB\$CRF_INS_REF, R9	
	MOVAB		SYMBOLSTRING, R8	
	SUBL2		#12, SP	
57 08	ADDL3	FC A8 C1	GSDOFFSET, OBJVEC, R7	1948
	MOVZBL	00 B8 9A	@SYMBOLSTRING, R0	1955
38 A8	CMPL	50 D1	R0, MAXSYMLNG	
	BLEQ	04 15	1\$	
38 A8	MOVL	50 D0	R0, MAXSYMLNG	1956
	PUSHL	5E DD	SP	1957
	PUSHAB	0B A0 9F	11(R0)	
	CALLS	0000G CF 02	#2, LIB_GET_ZMEM	
	BLBC	63 50 E9	STATUS, 4\$	
08 A6	MOVL	56 6E D0	SYMDESC, R6	1958
	MOVW	08 A7 B0	2(R7), 8(R6)	
	MOVL	50 68 D0	SYMBOLSTRING, R0	1959
0A A6	MOVB	0A A6 90	(R0), 10(R6)	
	MOVZBL	51 60 9A	(R0), R1	1960
08 A6	MOVC3	01 A0 51	R1, 1(R0), 11(R6)	
	MOVL	04 A6 14	@SYMBOLVAL, 4(R6)	1961
08 AE	EXTZV	02 A7 01	#0, #1, 2(R7), WEAKFLAG	1962
	CLRL	04 AE D4	DEFINITION	1963
	BBC	04 A7 01	#1, 2(R7), 2\$	1964
	MOVL	04 AE D0	#1, DEFINITION	1965
	BLBC	33 6A E9	LIB\$GL_CREFLAGS, 5\$	1966
	BLBC	17 04 AE	DEFINITION, 3\$	1968
	PUSHAB	08 A6 9F	8(R6)	1970
	PUSHAB	04 A6 9F	4(R6)	
	PUSHAB	0A A6 9F	10(R6)	
	PUSHAB	01EA C8 9F	LIB_CRFSYMVAL	
	CALLS	00000000G 00 04	#4, LIB\$CRF_INS_KEY	
	BLBC	67 50 E9	STATUS, 10\$	
	PUSHAB	04 AE 9F	DEFINITION	1977
	PUSHAB	0C AE 9F	WEAKFLAG	
7E 10	ADDL3	A8 08 C1	#8, CURMOD, -(SP)	
	PUSHAB	0A A6 9F	10(R6)	
	PUSHAB	01EA C8 9F	LIB_CRFSYMVAL	
	CALLS	69 05 FB	#5, LIB\$CRF_INS_REF	
	BLBC	4F 50 E9	STATUS, 10\$	
	BLBC	1E 04 AE	DEFINITION, 6\$	1984
19 02	BBS	A7 03 E0	#3, 2(R7), 6\$	
15	BBC	6A 01 E1	#1, LIB\$GL_CREFLAGS, 6\$	1985
	PUSHAB	40 A8 9F	A_ZERO	1987

			08	A6	9F	000AD		PUSHAB	8(R6)	:	
				68	DD	000B0		PUSHL	SYMBOLSTRING	:	
			04	A6	9F	000B2		PUSHAB	4(R6)	:	
			024A	C8	9F	000B5		PUSHAB	LIB_CRFBYVAL	:	
	69			05	FB	000B9		CALLS	#5, LIB\$CRF_INS_REF	:	
	2D			50	E9	000BC		BLBC	STATUS, 10\$	:	
26	6A			02	E1	000BF	6\$:	BBC	#2, LIB\$GL_CREFLAGS, 9\$	:	1993
				40	A8	9F	000C3	PUSHAB	A_ZERO	:	1997
				08	A6	9F	000C6	PUSHAB	8(R6)	:	
				0A	A6	9F	000C9	PUSHAB	10(R6)	:	
7E	10	A8		08	C1	000CC		ADDL3	#8, CURMOD, -(SP)	:	
		07		14	AE	E9	000D1	BLBC	DEFINITION, 7\$	:	
		50		02AA	C8	9E	000D5	MOVAB	LIB_CRFMODDEF, R0	:	
					05	11	000DA	BRB	8\$	:	
		50		030A	C8	9E	000DC	7\$:	MOVAB	LIB_CRFMODREF, R0	
					50	DD	000E1	8\$:	PUSHL	R0	
	69				05	FB	000E3		CALLS	#5, LIB\$CRF_INS_REF	
	03				50	E9	000E6		BLBC	STATUS, 10\$	
	50				01	D0	000E9	9\$:	MOVL	#1, R0	2003
					04	000EC	10\$:		RET	:	2004

: Routine Size: 237 bytes, Routine Base: \$CODE\$ + 05FC

```

896 2005 1 ROUTINE lib_crfout (linedesc) =
897 2006 2 BEGIN
898 2007 2
899 2008 2 This routine outputs a line to the listing file
900 2009 2
901 2010 2 Inputs:
902 2011 2
903 2012 2 linedesc address of string descriptor for line
904 2013 2
905 2014 2
906 2015 2 MAP
907 2016 2 linedesc : REF BBLOCK;
908 2017 2
909 2018 2 LOCAL
910 2019 2 ptr;
911 2020 2
912 2021 2 linecount = .linecount + 1; !Count the line
913 2022 2 IF .linecount GTR .linesperpage !If its time for a new page
914 2023 2 THEN perform (newpage ()); ! then do it now
915 2024 2 lib$al_rab [rab$w_rsz] = .linedesc [dsc$w_length];
916 2025 2 lib$al_rab [rab$l_rbf] = .linedesc [dsc$a_pointer];
917 2026 2
918 2027 2 Eliminate extra blanks from the end of the line
919 2028 2
920 2029 2 IF .lib$al_rab [rab$w_rsz] NEQ 0
921 2030 3 THEN BEGIN
922 2031 4 ptr = .lib$al_rab [rab$l_rbf] + .lib$al_rab [rab$w_rsz];
923 2032 4 WHILE (
924 2033 4 ptr = .ptr - 1;
925 2034 4 CH$RCHAR (.ptr) EQL %ASCII ' '
926 2035 4 )
927 2036 3 DO lib$al_rab [rab$w_rsz] = .lib$al_rab [rab$w_rsz] - 1;
928 2037 2 END;
929 P 2038 2 rms_perform ($PUT (RAB = lib$al_rab),
930 P 2039 2 lib$writeerr,
931 2040 2 .lib$al_rab [rab$l_stv], 1, lib$gl_outfdb [fdb$l_namdesc]);
932 2041 2 RETURN true
933 2042 1 END; !Of lib_crfout

```

.EXTRN SYSSPUT

0004 0000 LIB_CRFOUT:						
	52	0000G	CF 9E 00002	.WORD	Save R2	: 2005
		0000'	CF D6 00007	MOVAB	LIB\$AL_RAB+34, R2	: 2021
	0000'	0000'	CF D1 0000B	INCL	LINECOUNT	: 2022
			08 15 00012	CMPL	LINECOUNT, LINESPERPAGE	: 2023
	0000V		00 FB 00014	BLEQ	1\$	: 2024
	48		50 E9 00019	CALLS	#0, NEWPAGE	: 2025
	50	04	AC D0 0001C 1\$:	BLBC	STATUS, 5\$	: 2029
	62		60 B0 00020	MCVL	LINEDESC, R0	: 2031
	06	A2	04 A0 D0 00023	MOVW	(R0), LIB\$AL_RAB+34	
	50		62 3C 00028	MOVL	4(R0), LIB\$AL_RAB+40	
			0D 13 0002B	MOVZWL	LIB\$AL_RAB+34, R0	
			00 13 0002B	BEQL	3\$	
	50	06	A2 C0 0002D	ADDL2	LIB\$AL_RAB+40, PTR	

	20		70 91 00031 2\$:	CMPB	-(PTR), #32	:	2034
			04 12 00034	BNEQ	3\$	:	
			62 B7 00036	DECW	LIB\$AL_RAB+34	:	2036
			F7 11 00038	BRB	2\$	:	
		DE	A2 9F 0003A 3\$:	PUSHAB	LIB\$AL_RAB	:	2040
00000000G	00		01 FB 0003D	CALLS	#1, SYS\$PUT	:	
	1A		50 E8 00044	BLBS	STATUS, 4\$	:	
		EA	A2 DD 00047	PUSHL	LIB\$AL_RAB+12	:	
			50 DD 0004A	PUSHL	STATUS	:	
7E	0000G	CF	10 C1 0004C	ADDL3	#16, LIB\$GL_OUTFDB, -(SP)	:	
			01 DD 00052	PUSHL	#1	:	
		008610D2	8F DD 00054	PUSHL	#8786130	:	
00000000G	00		05 FB 0005A	CALLS	#5, LIB\$SIGNAL	:	
	50		01 D0 00061 4\$:	MOVL	#1, R0	:	2041
			04 00064 5\$:	RET		:	2042

; Routine Size: 101 bytes, Routine Base: \$CODE\$ + 06E9

```

: 935 2043 1 ROUTINE newpage =
: 936 2044 2 BEGIN
: 937 2045 2
: 938 2046 2 : This routine outputs a new page to the listing file
: 939 2047 2
: 940 2048 2 LOCAL
: 941 2049 2     pagebuffer : BBLOCK [lib$c_lisreclng],
: 942 2050 2     pagedesc : BBLOCK [dsc$c_s_bln];
: 943 2051 2
: 944 2052 2 BIND
: 945 2053 2     linelen = pagedesc [dsc$w_length] : WORD;
: 946 2054 2
: 947 2055 2     pagedesc [dsc$w_length] = lib$c_lisreclng;
: 948 2056 2     pagedesc [dsc$a_pointer] = pagebuffer;
: 949 2057 2     pagenum = .pagenum + 1;
: 950 2058 2     linecount = 0;
: 951 2059 2     perform (SYSS$FAO (fao_newpage, linelen, pagedesc));
: 952 2060 2     perform (lib_crfout (pagedesc));
: 953 2061 2     pagedesc [dsc$w_length] = lib$c_lisreclng;
: 954 2062 2     linecount = 0;
: 955 2063 2     perform (SYSS$FAO (fao_pageheader, linelen, pagedesc,
P 2064 2     .curheader, lib$gl_libfdb [fdb$l_namdesc],
P 2065 2     current_time, .pagenum));
: 958 2066 2     perform (lib_crfout (pagedesc));
: 959 2067 2     perform (outputsubhdr ());
: 960 2068 2 RETURN true
: 961 2069 1 END;

```

```

!Next page
!Start count at 0

!Form feed does not count as a line
!Format page header

! and output it
!Output sub headers

!Of newpage

```

			000C	00000	NEWPAGE: .WORD	Save R2,R3		2043	
	53	00000000G	00	9E	00002	MOVAB	SYSS\$FAO, R3		
	52	0000'	CF	9E	00009	MOVAB	PAGENUM, R2		
	5E	FF74	CE	9E	0000E	MOVAB	-140(SP), SP		
	6E	84	8F	9B	00013	MOVZBW	#132, PAGEDESC	2055	
	04	AE	AE	9E	00017	MOVAB	PAGEBUFFER, PAGEDESC+4	2056	
			62	D6	0001C	INCL	PAGENUM	2057	
			F8	A2	D4	0001E	CLRL	LINECOUNT	
			5E	DD	00021	PUSHL	SP	2058	
		04	AE	9F	00023	PUSHAB	LINELEN	2059	
		00CC	C2	9F	00026	PUSHAB	FAO_NEWPAGE		
	63		03	FB	0002A	CALLS	#3, SYSS\$FAO		
	44		50	E9	0002D	BLBC	STATUS, 1\$		
			5E	DD	00030	PUSHL	SP	2060	
	FF64	CF	01	FB	00032	CALLS	#1, LIB_CRFOUT		
		3A	50	E9	00037	BLBC	STATUS, -1\$		
		6E	84	8F	9B	0003A	MOVZBW	#132, PAGEDESC	2061
			F8	A2	D4	0003E	CLRL	LINECOUNT	2062
			62	DD	00041	PUSHL	PAGENUM	2065	
			04	A2	9F	00043	PUSHAB	CURRENT TIME	
	7E	0000G	CF	10	C1	00046	ADDL3	#16, LIB\$GL_LIBFDB, -(SP)	
			14	A2	DD	0004C	PUSHL	CURHEADER	
			10	AE	9F	0004F	PUSHAB	PAGEDESC	
			14	AE	9F	00052	PUSHAB	LINELEN	
		00D4	C2	9F	00055	PUSHAB	FAO_PAGEHEADER		



```

04 AE 9F 00033 PUSHAB LINELEN
52 DD 00036 PUSHL R2
64 03 FB 00038 CALLS #3, SYSS$FAO
24 50 E9 0003B BLBC STATUS, 1$
5E DD 0003E PUSHL SP
65 01 FB 00040 CALLS #1, LIB_CRFOUT
1C 50 E9 00043 BLBC STATUS, -1$
6E 84 8F 9B 00046 MOVZBW #132, LINEDESC
5E DD 0004A PUSHL SP
04 AE 9F 0004C PUSHAB LINELEN
53 DD 0004F PUSHL R3
64 03 FB 00051 CALLS #3, SYSS$FAO
0B 50 E9 00054 BLBC STATUS, 1$
5E DD 00057 PUSHL SP
65 01 FB 00059 CALLS #1, LIB_CRFOUT
03 50 E9 0005C BLBC STATUS, -1$
50 01 D0 0005F MOVL #1, R0
04 00062 1$: RET

```

; Routine Size: 99 bytes, Routine Base: \$CODE\$ + 07C3

```

: 987 2095 1 ROUTINE crferror (errorcode) =
: 988 2096 2 BEGIN
: 989 2097 2 !
: 990 2098 2 ! This routine is called by cross reference if an error is detected
: 991 2099 2 !
: 992 2100 2 SIGNAL (.errorcode);
: 993 2101 2 RETURN true
: 994 2102 1 END;

```

!Of crferror

```

0000 00000 CRFERRGR:
00000000G 00 04 AC DD 00002 .WORD Save nothing
50 01 FB 00005 PUSHL ERRORCODE
01 D0 0000C CALLS #1, LIB$SIGNAL
04 0000F RET

```

; Routine Size: 16 bytes, Routine Base: \$CODE\$ + 0826

```

: 995 2103 1 END
: 996 2104 0 ELUDOM

```

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
------	-------	------------



LIB\_CROSS  
V04=000

pro\_spsc

C 14  
16-Sep-1984 01:48:05  
14-Sep-1984 12:37:58

VAX-11 Bliss-32 V4.0-742  
DISK\$VM\$MASTER:[LIBRAR.SRC]CROSS.B32;1 (21) Page 47

LIB  
V04

:	SOWNS	878	NOVEC, WRT, RD	, NOEXE, NOSHR,	LCL, REL,	CON, NOPIC, ALIGN(2)
:	SPLITS	588	NOVEC, NOWRT, RD	, NOEXE, NOSHR,	LCL, REL,	CON, NOPIC, ALIGN(2)
:	SCODES	2102	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	103	1	581	00:01.0

COMMAND QUALIFIERS

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

: Size: 2102 code + 1466 data bytes  
 : Run Time: 00:50.3  
 : Elapsed Time: 01:37.7  
 : Lines/CPU Min: 2509  
 : Lexemes/CPU-Min: 31507  
 : Memory Used: 354 pages  
 : Compilation Complete

The image displays a grid of approximately 10 columns and 15 rows of terminal window screenshots. Each window shows a different screen from a library system. Several windows are large enough to be legible and are labeled as follows:

- TRANSFER LIS**: Located in the top row, 4th column from the left.
- PUTCACHE LIS**: Located in the second row, 1st column from the left.
- LIBRAR**: Located in the 4th row, 6th column from the left.
- LIBRARIAN MAP**: Located in the 5th row, 6th column from the left.
- PREFIX REQ**: Located in the 4th row, 7th column from the left.
- CROSS LIS**: Located in the 6th row, 7th column from the left.
- SUBS LIS**: Located in the 7th row, 1st column from the left.
- PADL BR LIS**: Located in the 8th row, 1st column from the left.
- COMPRESS LIS**: Located in the 8th row, 6th column from the left.
- LIB MOL**: Located in the 9th row, 6th column from the left.
- DELETE LIS**: Located in the 9th row, 9th column from the left.
- EXTRACT LIS**: Located in the 9th row, 10th column from the left.
- FILETO LIS**: Located in the 10th row, 10th column from the left.
- DATABASE LIS**: Located in the 2nd row, 10th column from the left.

The remaining screenshots in the grid are smaller and less legible, showing various data tables, command prompts, and system messages.