





```

1 0001 0 MODULE lib_inputobj (
2 0002 0
3 0003 0     LANGUAGE (BLISS32),
4 0004 0     IDENT = 'VJ4-000'
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1
9 0009 1 *****
10 0010 1 *
11 0011 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 *  ALL RIGHTS RESERVED.
14 0014 1 *
15 0015 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1 *  TRANSFERRED.
21 0021 1 *
22 0022 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1 *  CORPORATION.
25 0025 1 *
26 0026 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1 *
29 0029 1 *
30 0030 1 *****
31 0031 1
32 0032 1
33 0033 1 ++
34 0034 1
35 0035 1 FACILITY:  Library command processor
36 0036 1
37 0037 1 ABSTRACT:
38 0038 1
39 0039 1     The VAX/VMS librarian is invoked by DCL to process the LIBRARY
40 0040 1     command.  It utilizes the librarian procedure set to perform
41 0041 1     the actual modifications to the library.
42 0042 1
43 0043 1 ENVIRONMENT:
44 0044 1
45 0045 1     VAX native, user mode.
46 0046 1
47 0047 1 --
48 0048 1
49 0049 1
50 0050 1 AUTHOR:  Benn Schreiber,      CREATION DATE:  12-June-1979
51 0051 1
52 0052 1 MODIFIED BY:
53 0053 1
54 0054 1     V02-008      RPG0048      Bob Grosso      11-Mar-1982
55 0055 1     When symbol multiply defined in the same module,
56 0056 1     disregard subsequent references.
57 0057 1     Also fix up several places where $BYTEOFFSET should be used.

```

|    |      |   |         |  |                |             |
|----|------|---|---------|--|----------------|-------------|
| 58 | 0058 | 1 |         |  |                |             |
| 59 | 0059 | 1 | V02-007 | RPG0047  | Bob Grosso     | 02-Feb-1982 |
| 60 | 0060 | 1 |         | Support for logging replace operations in history.   |                |             |
| 61 | 0061 | 1 |         |  |                |             |
| 62 | 0062 | 1 | V02-006 | RPG0046  | Bob Grosso     | 21-Nov-1981 |
| 63 | 0063 | 1 |         | Support new GSD records                              |                |             |
| 64 | 0064 | 1 |         |  |                |             |
| 65 | 0065 | 1 | V02-005 | RPG0045  | Bob Grosso     | 7-Aug-1981  |
| 66 | 0066 | 1 |         | lib\$gl_ctlmsk now a quadword                        |                |             |
| 67 | 0067 | 1 |         |  |                |             |
| 68 | 0068 | 1 | V02-004 | RPG0036  | Bob Grosso     | 25-Jun-1981 |
| 69 | 0069 | 1 |         | Continue after a duplicate module.                   |                |             |
| 70 | 0070 | 1 |         |  |                |             |
| 71 | 0071 | 1 | V02-003 | RPG0035  | Bob Grosso     | 22-Apr-1981 |
| 72 | 0072 | 1 |         | Record module names for update history.              |                |             |
| 73 | 0073 | 1 |         |  |                |             |
| 74 | 0074 | 1 | V02-002 | BLS0029  | Benn Schreiber | 23-Dec-1980 |
| 75 | 0075 | 1 |         | Convert messages to message compiler. Add library of |                |             |
| 76 | 0076 | 1 |         | shareable image symbol tables.                       |                |             |
| 77 | 0077 | 1 | --      |  |                |             |

## Declarations

```
79 0078 1 %SBTTL 'Declarations';
80 0079 1
81 0080 1 LIBRARY
82 0081 1 'SYS$LIBRARY:LIB.L32';           !System macro definitions
83 0082 1 REQUIRE
84 0083 1 'PREFIX';                       !SET OF GENERAL MACROS ETC
85 0267 1 REQUIRE
86 0268 1 'LIBDEF';                       !Librarian structure defs.
87 0556 1 REQUIRE
88 0557 1 'LBRDEF';                       !Library processor defs.
89 1148 1
90 1149 1 EXTERNAL
91 1150 1   lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL), !RMS STV from Librarian
92 1151 1   lib$gl_objmodix, !Index number for module name index
93 1152 1   lib$gl_objgsdix, !index number for gsd symbols
94 1153 1   lib$gl_recount, !Count of records inserted
95 1154 1   lib$al_rab : BBLOCK, !Input file RAB
96 1155 1   lib$gl_type, !Type of library opened
97 1156 1   lib$gl_keysize, !Max size of key
98 1157 1   lib$gl_ctlmsk : BLOCK [2], !Control flags
99 1158 1   lib$gl_libfdb : REF BBLOCK, !Pointer to library fdb
100 1159 1   lib$gl_inpfdb : REF BBLOCK, !Pointer to input file fdb
101 1160 1   lib$gl_libctl; !Library control index
102 1161 1
103 1162 1 FORWARD ROUTINE
104 1163 1   prorec, !check sequence and copy record
105 1164 1   copyrec, !copy record to object library
106 1165 1   prohdr, !Routine to process module headers
107 1166 1   protir, !Routine to process TIR records
108 1167 1   progsd, !Routine to process gsd records
109 1168 1   proeom, !end of module
110 1169 1   seqchk, !
111 1170 1   propsectdef, !Process p-section definitions
112 1171 1   symbols, !Process symbol definitions and references
113 1172 1   entpnts, !Process entry point definitions
114 1173 1   procedef, !Process procedure declarations
115 1174 1   pro_epmw, !Process entry point definition with word psect
116 1175 1   pro_idc, !Process random entity check
117 1176 1   pro_env, !Process environment definition
118 1177 1   pro_lsy, !Process local symbol definition/reference
119 1178 1   pro_lepm, !Process local symbol entry point definition
120 1179 1   pro_lpro, !Process local symbol procedure definition
121 1180 1   pro_spsc, !Process shareable image psect definition
122 1181 1   profile, !Read all records of file
123 1182 1   finish_object, !Do end of module processing
124 1183 1   delsym, !Add symbol to delete symbol list
125 1184 1   prosymbol; !Do all the work of symbol resolution
126 1185 1
127 1186 1 EXTERNAL ROUTINE
128 1187 1   lib_get_mem, !Allocate virtual memory
129 1188 1   lib_get_zmem, !Allocate zeroed virtual memory
130 1189 1   lib_free_mem, !and give it back
131 1190 1   lib_log_op, !Log operation on console
132 1191 1   lib_log_upd, !record module names for LUH
133 1192 1   lbr$search : ADDRESSING_MODE (GENERAL), !Search index for keys with RFA
134 1193 1   lbr$delete_data : ADDRESSING_MODE (GENERAL), !Delete data
135 1194 1   lbr$put_record : ADDRESSING_MODE (GENERAL), !Write record to library
```

```

: 136 1195 1 lbr$put_end : ADDRESSING_MODE (GENERAL), !Terminated writing records
: 137 1196 1 lbr$lookup_key : ADDRESSING_MODE (GENERAL), !Lookup key in library
: 138 1197 1 lbr$set_index : ADDRESSING_MODE (GENERAL), !Set index number
: 139 1198 1 lbr$insert_key : ADDRESSING_MODE (GENERAL), !Insert key
: 140 1199 1 lbr$set_module : ADDRESSING_MODE (GENERAL), !Set module attributes
: 141 1200 1 lbr$replace_key : ADDRESSING_MODE (GENERAL), !Replace key
: 142 1201 1 lbr$delete_key : ADDRESSING_MODE (GENERAL), !Delete key from library
: 143 1202 1 get_record; !Get next input record
: 144 1203 1
: 145 1204 1 EXTERNAL LITERAL
: 146 1205 1 lib$_notshrimg, !File not shareable image
: 147 1206 1 lib$_nosymbols, !No stb in shareable image
: 148 1207 1 lib$_reclng, !Illegal record length
: 149 1208 1 lib$_rectyp, !Illegal record type
: 150 1209 1 lib$_noeom, !No eom record
: 151 1210 1 lib$_strlvl, !Illegal structure level
: 152 1211 1 lib$_modnamlng, !Illegal module name length
: 153 1212 1 lib$_indexerr, !Index error
: 154 1213 1 lib$_inserted, !Module inserted
: 155 1214 1 lib$_replaced, !Module replaced
: 156 1215 1 lib$_dupmodule, !Duplicate module
: 157 1216 1 lib$_gsdtyp, !Illegal gsd type
: 158 1217 1 lib$_spnamlng, !Illegal psect name length
: 159 1218 1 lib$_symnamlng, !Illegal symbol name length
: 160 1219 1 lib$_dupglobal, !Duplicate global
: 161 1220 1 lib$_comcod, !Compilation errors in module
: 162 1221 1 lib$_mhderr, !Module header error
: 163 1222 1 lib$_inserterr, !Insertion error
: 164 1223 1 lib$_delkeyerr, !Delete key error
: 165 1224 1 lib$_deldaterr, !Delete data error
: 166 1225 1 lib$_seqnce; !Record sequence error
: 167 1226 1
: 168 1227 1 OWN
: 169 1228 1 shrgsmatch, !GSMATCH for shareable image
: 170 1229 1 operation,
: 171 1230 1 mhdseen,
: 172 1231 1 lnmseen,
: 173 1232 1 dupseen, ! Record that a duplicate module is being processed
: 174 1233 1 gsdoffset, !Offset into concatenated gsd record
: 175 1234 1 symbolstring : REF VECTOR [,BYTE], !Pointer to current symbol
: 176 1235 1 recdesc : BBLOCK [dsc$_s_bln], !String descriptor for record
: 177 1236 1 lastrctyp, !Type of the previous record
: 178 1237 1 currectyp : INITIAL (obj$_eom), !Type of the current record
: 179 1238 1 maxreclng : INITIAL (obj$_maxreclng), !Maximum record length
: 180 1239 1 mod_name : VECTOR [sym$_maxlng+1, BYTE], !Module name
: 181 1240 1 modulerfa : BBLOCK [rfa$_length], !RFA of module text
: 182 1241 1 oldmodrfa : BBLOCK [rfa$_length], !RFA of old module text
: 183 1242 1 replacing, !Flag if replacing this module
: 184 1243 1 moduledesc : BBLOCK [dsc$_s_bln] INITIAL !String descriptor for module name
: 185 1244 1 (0, mod_name [1])
: 186 1245 1 moduledata : VECTOR [sym$_maxlng + 2, BYTE], !Moduleflags, idlng, moduleid
: 187 1246 1 globlist : VECTOR [2], !Listhead for globals to insert
: 188 1247 1 delist : VECTOR [2], !Listhead for globals to delete
: 189 1248 1 compilecods : BBLOCK [5 * dsc$_s_bln] INITIAL !Name the compilation completion codes
: 190 1249 1 (STRINGDESC ('success'),
: 191 1250 1 STRINGDESC ('warnings'),
: 192 1251 1 STRINGDESC ('errors'),

```

Declarations

```
: 193 1252 1          STRINGDESC ('fatal errors'),  
: 194 1253 1          STRINGDESC ('illegal compilation code'));  
: 195 1254 1  
: 196 1255 1 BIND  
: 197 1256 1      modnamlng = mod name [0] : BYTE,           !Name the module name length  
: 198 1257 1      modulename = mod name [1] : VECTOR [,BYTE], ! and the module name  
: 199 1258 1      moduleflags = moduledata [0] : BYTE,       !Name module flags byte  
: 200 1259 1      idlng = moduledata [1] : BYTE,             !Length of module ident  
: 201 1260 1      moduleid = moduledata [2] : VECTOR [,BYTE], !Name module ident  
: 202 1261 1      reclng = recdesc [dsc$w_length] : WORD,    !Name the length of the record  
: 203 1262 1      objrec = recdesc [dsc$a_pointer] : REF BBLOCK, ! and the pointer  
: 204 1263 1      objvec = recdesc [dsc$a_pointer] : REF VECTOR [,BYTE],  
: 205 1264 1      recdispatch = PLIT(                          !Set up maximum allowed record type  
: 206 1265 1          prohdr,                                !0 - module header  
: 207 1266 1          progsd,                                !1 - gsd records  
: 208 1267 1          protir,                                !2 - tir  
: 209 1268 1          proeom,                                !3 - end of module  
: 210 1269 1          prorec,                                !4 - dbg - check sequence and copy  
: 211 1270 1          prorec,                                !5 - tbt - check sequence and copy  
: 212 1271 1          prorec,                                !6 - lnk - check sequence and copy  
: 213 1272 1          proeom) : VECTOR;                       !7 - eomw  
: 214 1273 1 BUILTIN  
: 215 1274 1     INSQUE,  
: 216 1275 1     REMQUE;
```

```
LIB-INPUT_OBJ
: 218 1276 1 %SBTTL 'LIB-INPUT_OBJ';
: 219 1277 1
: 220 1278 1 GLOBAL ROUTINE lib_input_obj =
: 221 1279 2 BEGIN
: 222 1280 2
: 223 1281 2 | Process an object file
: 224 1282 2 |
: 225 1283 2 LOCAL
: 226 1284 2     hdrblkcnt,
: 227 1285 2     symdsc : REF BBLOCK,
: 228 1286 2     status;
: 229 1287 2
: 230 1288 2 IF .lib$gl_ctlmsk [lib$v_shrstb]           !If processing shareable image stb
: 231 1289 3 THEN BEGIN
: 232 1290 3     lib$al_rab [rab$l_bkt] = 1;           !Set to read block 1
: 233 1291 3     lib$al_rab [rab$w_usz] = 512;       ! and only block 1
: 234 1292 3     rms_perform ($READ (RAB = lib$al_rab), !Read the image header
: 235 1293 3     lib$readerr, ! report any error
: 236 1294 3     .lib$al_rab [rab$l_stv], 1, lib$gl_inpfdb [fdb$l_namdesc]);
: 237 1295 3
: 238 1296 3 IF .lib$al_rab [rab$w_rsz] NEQ 512     ! Image header is 512 bytes long
: 239 1297 4 OR (
: 240 1298 4     BIND
: 241 1299 4     header = .lib$al_rab [rab$l_ubf] : BBLOCK;
: 242 1300 4
: 243 1301 4     IF .header[ihd$b_imgtype] NEQ ihd$k_lim ! type must agree
: 244 1302 4     OR .header[ihd$w_majorid] NEQ ihd$k_majorid ! major header id must match
: 245 1303 4     OR .header[ihd$w_minorid] GTRU ihd$k_minorid ! minor id must not be greater
: 246 1304 5     OR .header[ihd$w_size] GTRU MAXU((.header[ihd$w_patchoff]
: 247 1305 4     + ihp$k_length),ihd$k_length+
: 248 1306 4     ihask_length+ihsk_length+ihik_length) ! Header fixed part must be
: 249 1307 4     and contained in header
: 250 1308 4     OR (hdrblkcnt = .header[ihd$b_hdrblkcnt]-1) LSS 0
: 251 1309 5     OR (symdsc = header + .header[ihd$w_syndbgoff]) ! GST descriptor must be
: 252 1310 5     GEQU (header + .header[ihd$w_size]) ! contained in header
: 253 1311 4     OR (.symdsc[ih$w_gstrecs]) LSSU 3 ! Must be at least 3 blocks
: 254 1312 4     OR (.symdsc[ih$l_gstvbn]) LEQU ! and must be beyond header blocks
: 255 1313 5     (.hdrblkcnt + 2)
: 256 1314 4     THEN true !It's not a shareable image
: 257 1315 5     ELSE (shrgsmatch = .header[ihd$l_ident]; !It's a shareable image, so save the gsmatch
: 258 1316 4     false))
: 259 1317 4     THEN BEGIN
: 260 1318 4     SIGNAL (lib$_notshring, 1, lib$gl_inpfdb [fdb$l_namdesc]);
: 261 1319 4     RETURN lib$_notshring;
: 262 1320 3     END;
: 263 1321 3     lib$al_rab [rab$b_rac] = rab$c_rfa; !Set to point to object file
: 264 1322 3     IF (lib$al_rab [rab$l_rfa0] = .symdsc [ih$l_gstvbn]) NEQ 0 ! which is the symbol table
: 265 1323 4     THEN BEGIN
: 266 1324 4     lib$al_rab [rab$w_rfa4] = 0; ! on a block boundary
: 267 1325 4     rms_perform ($FIND (RAB = lib$al_rab),
: 268 1326 4     lib$readerr, 1, lib$gl_inpfdb [fdb$l_namdesc]);
: 269 1327 4     lib$al_rab [rab$b_rac] = rab$c_seq; !Reset to sequential
: 270 1328 4     END
: 271 1329 4     ELSE BEGIN
: 272 1330 4     SIGNAL (lib$_nosymbols,1,lib$gl_inpfdb [fdb$l_namdesc]);
: 273 1331 4     RETURN true
: 274 1332 3     END;
```



```

: 275      1333 2      END;
: 276      1334 2      status = profile ();
: 277      1335 2      IF NOT .status
: 278      1336 2          THEN finish_object (false);
: 279      1337 2      RETURN .status
: 280      1338 1      END;

```

```

!Clean up if an error
!Of lib_input_obj

```

```

                                .TITLE LIB_INPUTOBJ
                                .IDENT  \V04-000\
                                .PSECT  $SPLITS$,NOWRT,NOEXE,2
                                00 73 73 65 63 63 75 73 00000 P.AAA: .ASCII  \success\<0>
                                73 67 6E 69 6E 72 61 77 00008 P.AAB: .ASCII  \warnings\
                                00 00 73 72 6F 72 72 65 00010 P.AAC: .ASCII  \errors\<0><0>
61 6C 69 73 72 6F 72 72 65 20 6C 61 74 61 66 00018 P.AAD: .ASCII  \fatal errors\
                                63 20 6C 61 67 65 6C 6C 69 00024 P.AAE: .ASCII  \illegal compilation code\
                                65 64 6F 63 20 6E 6F 69 74 00033
                                00000008 0003C
00000000V 00000000V 00000000V 00000000V 00000000V 00000000V 00040 P.AAF: .ADDRESS PROHDR, PROGSD, PROTIR, PROEOM, PROREC, -
                                00000000V 00000000V 00058          PROREC, PROREC, PROEOM
                                .PSECT  $OWNS$,NOEXE,2
                                00000 SHRGSMATCH:
                                .BLKB  4
                                00004 OPERATION:
                                .BLKB  4
                                00008 MHDSEEN: .BLKB  4
                                0000C LNMSEEN: .BLKB  4
                                00010 DUPSEEN: .BLKB  4
                                00014 GSDOFFSET:
                                .BLKB  4
                                00018 SYMBOLSTRING:
                                .BLKB  4
                                0001C RECDESC: .BLKB  8
                                00024 LASTRECTYP:
                                .BLKB  4
                                00000003 00028 CURRECTYP:
                                .LONG  3
                                00000800 0002C MAXRECLNG:
                                .LONG  2048
                                00030 MOD_NAME:
                                .BLKB  32
                                00050 MODULERFA:
                                .BLKB  6
                                00056 .BLKB  2
                                00058 OLDMODRFA:
                                .BLKB  6
                                0005E .BLKB  2
                                00060 REPLACING:
                                .BLKB  4
                                00000000 00064 MODULEDESC:
                                .LONG  0
                                00000000' 00068 .ADDRESS MOD_NAME+1
                                0006C MODULEDATA:

```

```

          .BLKB 33
0008D    .BLKB 3
00090 GLOBLIST:
          .BLKB 8
00098 DELIST: .BLKB 8
00000007 000A0 COMPILECODS:
          .LONG 7
00000000' 000A4 .ADDRESS P.AAA
00000008' 000A8 .LONG 8
00000000' 000AC .ADDRESS P.AAB
00000006' 000B0 .LONG 6
00000000' 000B4 .ADDRESS P.AAC
0000000C' 000B8 .LONG 12
00000000' 000BC .ADDRESS P.AAD
00000018' 000C0 .LONG 24
00000000' 000C4 .ADDRESS P.AAE

```

```

MODNAMLN= MOD_NAME
MODULENAME= MOD_NAME+1
MODULEFLAGS= MODULEDATA
IDLNG= MODULEDATA+1
MODULEID= MODULEDATA+2
RECLNG= RECDISC
OBJREC= RECDISC+4
OBJVEC= RECDISC+4
RECDISPATCH= P.AAF
.EXTRN LBR$GL_RMSSTV, LIB$GL_OBJMODIX
.EXTRN LIB$GL_OBJGSDIX
.EXTRN LIB$GL_RECOUNT, LIB$AL_RAB
.EXTRN LIB$GL_TYPE, LIB$GL_KEYSIZE
.EXTRN LIB$GL_CTLMSK, LIB$GL_LIBFDB
.EXTRN LIB$GL_INPFDB, LIB$GL_LIBCTL
.EXTRN LIB_GET_MEM, LIB_GET_ZMEM
.EXTRN LIB_FREE_MEM, LIB_LOG_OP
.EXTRN LIB_LOG_OPD, LBR$SEARCH
.EXTRN LBR$DELETE_DATA
.EXTRN LBR$PUT_RECORD, LBR$PUT_END
.EXTRN LBR$LOOKUP_KEY, LBR$SET_INDEX
.EXTRN LBR$INSERT_KEY, LBR$SET_MODULE
.EXTRN LBR$REPLACE_KEY
.EXTRN LBR$DELETE_KEY, GET_RECORD
.EXTRN LIB$NOTSHRIMG, LIB$NOSYMBOLS
.EXTRN LIB$RECLNG, LIB$RECTYP
.EXTRN LIB$NOEOM, LIB$STRLVL
.EXTRN LIB$MODNAMLN, LIB$INDEXERR
.EXTRN LIB$INSERTED, LIB$REPLACED
.EXTRN LIB$DUPMODULE, LIB$GSDTYP
.EXTRN LIB$SPNAMLN, LIB$SYMNAMLN
.EXTRN LIB$DUPGLOBAL, LIB$COMCOD
.EXTRN LIB$MHDERR, LIB$INSERTERR
.EXTRN LIB$DELKEYERR, LIB$DELDATEERR
.EXTRN LIB$SEQNCE, SYS$READ
.EXTRN SYS$FIND

```

.PSECT \$CODE\$,NOWRT,2

00FC 0000

.ENTRY LIB\_INPUT\_OBJ, Save R2,R3,R4,R5,R6,R7

: 1278



|           |    |           |    |       |       |            |                            |   |      |
|-----------|----|-----------|----|-------|-------|------------|----------------------------|---|------|
|           |    |           | 25 | 13    | 000D3 | BEQL       | 7\$                        |   |      |
|           |    | 14        | A4 | B4    | 000D5 | CLRW       | LIB\$AL_RAB+20             | : | 1324 |
|           |    |           | 54 | DD    | 000D8 | PUSHL      | R4                         | : | 1326 |
| 00000000G | 00 |           | 01 | FB    | 000DA | CALLS      | #1, SYSS\$FIND             | : |      |
|           | 11 |           | 50 | EB    | 000E1 | BLBS       | STATUS, 6\$                | : |      |
|           |    |           | 01 | DD    | 000E4 | PUSHL      | #1                         | : |      |
|           |    |           | 50 | DD    | 000E6 | PUSHL      | STATUS                     | : |      |
| 7E        | 66 |           | 10 | C1    | 000E8 | ADDL3      | #16, LIB\$GL_INPFDB, -(SP) | : |      |
|           |    | 008610B2  | 8F | DD    | 000EC | PUSHL      | #8786098                   | : |      |
|           | 65 |           | 04 | FB    | 000F2 | CALLS      | #4, LIB\$SIGNAL            | : |      |
|           |    | 1E        | A4 | 94    | 000F5 | 6\$: CLRB  | LIB\$AL_RAB+30             | : | 1327 |
|           |    |           | 13 | 11    | 000F8 | BRB        | 8\$                        | : | 1322 |
| 7E        | 66 |           | 10 | C1    | 000FA | 7\$: ADDL3 | #16, LIB\$GL_INPFDB, -(SP) | : | 1330 |
|           |    |           | 01 | DD    | 000FE | PUSHL      | #1                         | : |      |
|           |    | 00000000G | 8F | DD    | 00100 | PUSHL      | #LIB\$ NOSYMBOLS           | : |      |
|           | 65 |           | 03 | FB    | 00106 | CALLS      | #3, LIB\$SIGNAL            | : |      |
|           | 50 |           | 01 | DD    | 00109 | MOVL       | #1, R0                     | : | 1331 |
|           |    |           | 04 | 0010C |       | RET        |                            | : |      |
| 0000V     | CF |           | 00 | FB    | 0010D | 8\$: CALLS | #0, PROFILE                | : | 1334 |
|           | 52 |           | 50 | DD    | 00112 | MOVL       | R0, STATUS                 | : |      |
|           | 07 |           | 52 | EB    | 00115 | BLBS       | STATUS, 9\$                | : | 1335 |
|           |    |           | 7E | D4    | 00118 | CLRL       | -(SP)                      | : | 1336 |
| 0000V     | CF |           | 01 | FB    | 0011A | CALLS      | #1, FINISH_OBJECT          | : |      |
|           | 50 |           | 52 | DD    | 0011F | 9\$: MOVL  | STATUS, R0                 | : | 1337 |
|           |    |           | 04 | 00122 |       | RET        |                            | : | 1338 |

; Routine Size: 291 bytes, Routine Base: \$CODE\$ + 0000

```
profile
282 1339 1 %SBTTL 'profile';
283 1340 1
284 1341 1 ROUTINE profile =
285 1342 2 BEGIN
286 1343 2
287 1344 2 Read and process all required object module records of the file just opened
288 1345 2 that is, keep reading records to end of file.
289 1346 2
290 1347 2
291 1348 2
292 1349 2 LOCAL
293 1350 2 status:
294 1351 2
295 1352 2 modnamlng = 0; !Zero module name
296 1353 2 modulerfa [rfa$l_vbn] = 0; !Clear VBN
297 1354 2 mhdsen = false;
298 1355 2 lnmseen = false;
299 1356 2 currectyp = obj$c_eom; !Init record to end of module type
300 1357 2 globlist [0] = globlist [0]; !Init globals listhead
301 1358 2 globlist [1] = globlist [0];
302 1359 2 delist [0] = delist [0];
303 1360 2 delist [1] = delist [0];
304 1361 2 moduleflags = 0; ! Zero module flags
305 1362 2 WHILE (status = get_record (recdesc)) NEQ rms$eof ! While there are more records
306 1363 3 DO BEGIN
307 1364 3 lib$gl_recount = .lib$gl_recount + 1; ! Count the record
308 1365 3 IF .reclng GTRU .maxreclng ! And if its length is illegal
309 1366 4 THEN BEGIN
310 1367 4 SIGNAL (lib$_reclng, 3, .reclng, ! then signal the error and give up on this file
311 1368 4 modnamlng, lib$gl_inpfdb [fdb$_namdesc]);
312 1369 4 RETURN lib$_reclng;
313 1370 3 END;
314 1371 3 lastrectyp = .currectyp; ! Copy old current to last type
315 1372 3 currectyp = .objrec [obj$b_rectyp]; ! And get new type
316 1373 3 IF .currectyp LSSU .recdispatch [-1] ! Check it is legal and if
317 1374 3 THEN
318 1375 4 BEGIN
319 1376 4
320 1377 4 If a duplicate module is being processed then ignore record
321 1378 4 unless it is a new module header record.
322 1379 4
323 1380 5 IF (NOT .dupseen)
324 1381 4 THEN
325 1382 4 perform ((.recdispatch [.currectyp]) ()); ! So dispatch to record specific routine
326 1383 5 IF .dupseen AND (.currectyp EQL 3)
327 1384 4 THEN
328 1385 4 dupseen = false;
329 1386 4 END
330 1387 3 ELSE
331 1388 4 BEGIN
332 1389 4 SIGNAL (lib$_rectyp, 3, .currectyp, !If unknown, signal and give up
333 1390 4 modnamlng, lib$gl_inpfdb [fdb$_namdesc]);
334 1391 4 RETURN lib$_rectyp;
335 1392 3 END;
336 1393 3 IF .lib$gl_ctlmsk [lib$v_shrstb]
337 1394 3 AND .currectyp EQL obj$c_eom
338 1395 3 THEN EXITLOOP;
```

```

profile
: 339 1396 2 END;
: 340 1397 2 IF .correctyp NEQ obj$c_eom ! Of records loop
: 341 1398 2 THEN BEGIN ! All done, did we end with eom?
: 342 1399 2 SIGNAL (lib$noeom, 2, modnamlng, lib$gl_inpfdb [fdb$l_namdesc]); !no, signal and return
: 343 1400 2 RETURN lib$noeom;
: 344 1401 2 END;
: 345 1402 2 RETURN true ! Finally return after no more
: 346 1403 1 END; ! Of lib_input_obj

```

|  |    |          |    |                               |                |                            |  |      |
|--|----|----------|----|-------------------------------|----------------|----------------------------|--|------|
|  |    |          |    | 01FC 00000                    | PROFILE: .WORD | Save R2,R3,R4,R5,R6,R7,R8  |  | 1341 |
|  |    |          |    | 58 0000G CF 9E 00002          | MOVAB          | LIB\$GL_INPFDB, R8         |  |      |
|  |    |          |    | 57 00000000G 8F D0 00007      | MOVL           | #LIB\$NOEOM, R7            |  |      |
|  |    |          |    | 56 00000000G 8F D0 0000E      | MOVL           | #LIB\$RECTYP, R6           |  |      |
|  |    |          |    | 55 00000000G 8F D0 00015      | MOVL           | #LIB\$RECLNG, R5           |  |      |
|  |    |          |    | 54 00000000G 00 9E 0001C      | MOVAB          | LIB\$SIGNAL, R4            |  |      |
|  |    |          |    | 53 0000' CF 9E 00023          | MOVAB          | CORRECTYP, R3              |  |      |
|  |    |          |    | 08 A3 94 00028                | CLRB           | MODNAMLNG                  |  | 1352 |
|  |    |          |    | 28 A3 D4 0002B                | CLRL           | MODULERFA                  |  | 1353 |
|  |    |          |    | E0 A3 7C 0002E                | CLRQ           | MHDSEEN                    |  | 1354 |
|  |    |          |    | 63 03 D0 00031                | MOVL           | #3, CORRECTYP              |  | 1356 |
|  | 68 | A3       | 68 | A3 9E 00034                   | MOVAB          | GLOBLIST, GLOBLIST         |  | 1357 |
|  | 6C | A3       | 68 | A3 9E 00039                   | MOVAB          | GLOBLIST, GLOBLIST+4       |  | 1358 |
|  | 70 | A3       | 70 | A3 9E 0003E                   | MOVAB          | DELIST, DELIST             |  | 1359 |
|  | 74 | A3       | 70 | A3 9E 00043                   | MOVAB          | DELIST, DELIST+4           |  | 1360 |
|  |    |          |    | 44 A3 94 00048                | CLRB           | MODULEFLAGS                |  | 1361 |
|  |    |          |    | F4 A3 9F 0004B                | PUSHAB         | RECDESC                    |  | 1362 |
|  |    | 0000G    |    | CF 01 FB 0004E                | CALLS          | #1, GET RECORD             |  |      |
|  |    |          |    | 52 50 D0 00053                | MOVL           | R0, STATUS                 |  |      |
|  |    | 0001827A |    | 8F 52 D1 00056                | CMPL           | STATUS, #98938             |  |      |
|  |    |          |    | 75 13 0005D                   | BEQL           | 8\$                        |  |      |
|  |    |          |    | 04 A3 F4 A3 0000G CF D6 0005F | INCL           | LIB\$GL_RECOUNT            |  | 1364 |
|  |    |          |    | 10 00 00 00063                | CMPZV          | #0, #16, RECLNG, MAXRECLNG |  | 1365 |
|  |    |          |    | 16 1B 0006A                   | BLEQU          | 2\$                        |  |      |
|  |    |          |    | 68 10 C1 0006C                | ADDL3          | #16, LIB\$GL_INPFDB, -(SP) |  | 1368 |
|  |    |          |    | 08 A3 9F 00070                | PUSHAB         | MODNAMLNG                  |  | 1367 |
|  |    |          |    | 7E F4 A3 3C 00073             | MOVZWL         | RECLNG, -(SP)              |  | 1368 |
|  |    |          |    | 03 DD 00077                   | PUSHL          | #3                         |  |      |
|  |    |          |    | 55 DD 00079                   | PUSHL          | R5                         |  |      |
|  |    |          |    | 64 05 FB 0007B                | CALLS          | #5, LIB\$SIGNAL            |  |      |
|  |    |          |    | 50 55 D0 0007E                | MOVL           | R5, R0                     |  | 1369 |
|  |    |          |    | 04 00081                      | RET            |                            |  |      |
|  |    |          |    | FC A3 63 D0 00082             | MOVL           | CORRECTYP, LASTRECTYP      |  | 1371 |
|  |    |          |    | 63 F8 B3 9A 00086             | MOVZBL         | @OBJREC, CORRECTYP         |  | 1372 |
|  |    |          |    | 50 63 D0 0008A                | MOVL           | CORRECTYP, R0              |  | 1373 |
|  |    |          |    | 0000' CF 50 D1 0008D          | CMPL           | R0, RECDISPATCH-4          |  |      |
|  |    |          |    | 1E 1E 00092                   | BGEQU          | 4\$                        |  |      |
|  |    |          |    | 10 E8 A3 E8 00094             | BLBS           | DUPSEEN, 3\$               |  | 1380 |
|  |    |          |    | 50 0000' CF 40 D0 00098       | MOVL           | RECDISPATCH[R0], R0        |  | 1382 |
|  |    |          |    | 60 00 FB 0009E                | CALLS          | #0, (R0)                   |  |      |
|  |    |          |    | 4A 50 E9 000A1                | BLBC           | STATUS, 10\$               |  |      |
|  |    |          |    | 1E E8 A3 E9 000A4             | BLBC           | DUPSEEN, 5\$               |  | 1383 |
|  |    |          |    | 03 63 D1 000AB                | CMPL           | CORRECTYP, #3              |  |      |
|  |    |          |    | 19 12 000AB                   | BNEQ           | 5\$                        |  |      |

|    |       |    |    |      |    |       |       |        |                            |   |      |
|----|-------|----|----|------|----|-------|-------|--------|----------------------------|---|------|
|    |       |    | E8 | A3   | D4 | 000AD |       | CLRL   | DUPSEEN                    | : | 1385 |
|    |       |    |    | 14   | 11 | 000B0 |       | BRB    | 5\$                        | : | 1373 |
| 7E |       | 68 |    | 10   | C1 | 000B2 | 4\$:  | ADDL3  | #16, LIB\$GL_INPFDB, -(SP) | : | 1390 |
|    |       |    | 08 | A3   | 9F | 000B6 |       | PUSHAB | MODNAMLNG                  | : | 1389 |
|    |       |    |    | 50   | DD | 000B9 |       | PUSHL  | R0                         | : | 1390 |
|    |       |    |    | 03   | DD | 000BB |       | PUSHL  | #3                         | : |      |
|    |       |    |    | 56   | DD | 000BD |       | PUSHL  | R6                         | : |      |
|    |       | 64 |    | 05   | FB | 000BF |       | CALLS  | #5, LIB\$SIGNAL            | : |      |
|    |       | 50 |    | 56   | DD | 000C2 |       | MOVL   | R6, R0                     | : | 1391 |
|    |       |    |    |      | 04 | 000C5 |       | RET    |                            | : |      |
| 03 | 0000G | CF |    | 05   | E0 | 000C6 | 5\$:  | BBS    | #5, LIB\$GL_CTLMSK, 7\$    | : | 1393 |
|    |       |    |    | FF7C | 31 | 000CC | 6\$:  | BRW    | 1\$                        | : |      |
|    |       | 03 |    | 63   | D1 | 000CF | 7\$:  | CMPL   | CURRECTYP, #3              | : | 1394 |
|    |       |    |    | F8   | 12 | 000D2 |       | BNEQ   | 6\$                        | : |      |
|    |       | 03 |    | 63   | D1 | 000D4 | 8\$:  | CMPL   | CURRECTYP, #3              | : | 1397 |
|    |       |    |    | 12   | 13 | 000D7 |       | BEQL   | 9\$                        | : |      |
| 7E |       | 68 |    | 10   | C1 | 000D9 |       | ADDL3  | #16, LIB\$GL_INPFDB, -(SP) | : | 1399 |
|    |       |    | 08 | A3   | 9F | 000DD |       | PUSHAB | MODNAMLNG                  | : |      |
|    |       |    |    | 02   | DD | 000E0 |       | PUSHL  | #2                         | : |      |
|    |       |    |    | 57   | DD | 000E2 |       | PUSHL  | R7                         | : |      |
|    |       | 64 |    | 04   | FB | 000E4 |       | CALLS  | #4, LIB\$SIGNAL            | : |      |
|    |       | 50 |    | 57   | DD | 000E7 |       | MOVL   | R7, R0                     | : | 1400 |
|    |       |    |    |      | 04 | 000EA |       | RET    |                            | : |      |
|    |       | 50 |    | 01   | DD | 000EB | 9\$:  | MOVL   | #1, R0                     | : | 1402 |
|    |       |    |    |      | 04 | 000EE | 10\$: | RET    |                            | : | 1403 |

; Routine Size: 239 bytes, Routine Base: \$CODE\$ + 0123





```

: 405      1461      3      CH$MOVE(4,shrgsmatch,moduleid);          !Copy the GSMATCH into module header data
: 406      1462      3      END
: 407      1463      3      ELSE BEGIN
: 408      1464      3      idlng = MINU (sym$cl_maxlng, .modidstring [0]);
: 409      1465      3      CH$MOVE (.modidstring [0], modidstring [1], moduleid);
: 410      1466      3      END;
: 411      1467      3      moduledesc [dsc$w_length] = .modnamlng;
: 412      P 1468      3      perform (lbr$set_index (lib$gl_libctl, lib$gl_objmodix),
: 413      1469      3      [lib$_indexerr,-1, lib$gl_libfdb [fdb$_namdesc]);
: 414      1470      3      replacing = false;
: 415      1471      3      operation = lib$_inserted;
: 416      1472      3
: 417      1473      3      CH$FILL (0, rfa$cl_length, oldmodrfa);          ! initialize rfa
: 418      1474      3      IF lbr$lookup_key (lib$gl_libctl, moduledesc, oldmodrfa)          ! If in library already
: 419      1475      3      THEN IF .lib$gl_ctlmsk [lib$_replace]          ! If replace
: 420      1476      3
: 421      1477      3      ! Key in index, and replacing. Find globals that belong with old
: 422      1478      3      ! module and put on list.
: 423      1479      3
: 424      1480      3      THEN BEGIN
: 425      1481      3      lbr$search (lib$gl_libctl, lib$gl_objgsdix, oldmodrfa, delsym);
: 426      1482      3      replacing = true;
: 427      1483      3      operation = lib$_replaced;          !Set for proeom
: 428      1484      3      END
: 429      1485      3      ELSE BEGIN
: 430      1486      3      SIGNAL (lib$_dupmodule, 3, modnamlng, lib$gl_inpfdb [fdb$_namdesc],
: 431      1487      3      lib$gl_libfdb [fdb$_namdesc]);
: 432      1488      3      dupseen = true;
: 433      1489      3      RETURN true;
: 434      1490      3      END;
: 435      1491      3
: 436      1492      2      perform (copyrec ());          !Copy record to library
: 437      1493      2
: 438      1494      2      RETURN true
: 439      1495      1      END;          ! OF prohdr

```

|  |       |           |      |       |         |        |                                      |        |
|--|-------|-----------|------|-------|---------|--------|--------------------------------------|--------|
|  |       |           | OFFC | 00000 | PROHDR: | .WORD  | Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 | : 1406 |
|  | 5B    | 00000000G | 8F   | D0    |         | MOVL   | #LIB\$_RECLNG, R11                   |        |
|  | 5A    | 00000000G | 8F   | D0    |         | MOVL   | #LIB\$_STRLVL, R10                   |        |
|  | 59    | 0000G     | CF   | 9E    |         | MOVAB  | LIB\$G[INPFDB], R9                   |        |
|  | 58    | 00000000G | 00   | 9E    |         | MOVAB  | LIB\$SIGNAL, R8                      |        |
|  | 57    | 0000'     | CF   | 9E    |         | MOVAB  | OBJREC, R7                           |        |
|  | 5E    |           | 08   | C2    |         | SUBL2  | #8, SP                               |        |
|  | 51    |           | 67   | D0    |         | MOVL   | OBJREC, R1                           | : 1429 |
|  | 50    | 05        | A1   | 9A    |         | MOVZBL | 5(R1), R0                            |        |
|  | 56    | 06        | A140 | 9F    |         | MOVAB  | 6(R1)[R0], R6                        |        |
|  | 0000V | CF        | 00   | F8    |         | CALLS  | #0, SEQCHK                           | : 1431 |
|  | 01    |           | 50   | E8    |         | BLBS   | STATUS, 1\$                          |        |
|  |       |           |      | 04    |         | RET    |                                      |        |
|  | 50    |           | 67   | D0    |         | MOVL   | OBJREC, R0                           | : 1432 |
|  |       | 01        | A0   | 95    |         | TSTB   | 1(R0)                                |        |
|  |       |           | 0F   | 13    |         | BEQL   | 3\$                                  |        |
|  | 03    | 0000G     | CF   | 05    |         | BBC    | #5, LIB\$GL_CTLMSK, 2\$              | : 1433 |



06

|    |           |    |           |                   |        |                             |      |
|----|-----------|----|-----------|-------------------|--------|-----------------------------|------|
|    | 00000000G | 00 | 0000G     | CF 9F 00109       | PUSHAB | LIB\$GL_LIBCTL              |      |
|    |           | 13 |           | 02 FB 0010D       | CALLS  | #2, LBR\$SET_INDEX          |      |
|    |           |    |           | 50 E8 00114       | BLBS   | STATUS, 11\$                |      |
| 7E | 0000G     | CF |           | 50 DD 00117       | PUSHL  | STATUS                      |      |
|    |           |    |           | 10 C1 00119       | ADDL3  | #16, LIB\$GL_LIBFDB, -(SP)  |      |
|    |           |    | 00000000G | 01 DD 0011F       | PUSHL  | #1                          |      |
|    |           | 68 |           | 8F DD 00121       | PUSHL  | #LIB\$ INDEXERR             |      |
|    |           |    | 40        | 04 FB 00127       | CALLS  | #4, LIB\$SIGNAL             |      |
|    | E4        | A7 | 00000000G | A7 D4 0012A 11\$: | CLRL   | REPLACING                   | 1470 |
| 00 |           | 6E |           | 8F D0 0012D       | MOVL   | #LIB\$ INSERTED, OPERATION  | 1471 |
|    |           |    |           | 00 2C 00135       | MOVCS  | #0, (SP), #0, #6, OLDMODRFA | 1473 |
|    |           |    | 38        | A7 9F 0013A       |        |                             |      |
|    |           |    | 38        | A7 9F 0013C       | PUSHAB | OLDMODRFA                   | 1474 |
|    |           |    | 44        | A7 9F 0013F       | PUSHAB | MODULEDESC                  |      |
|    |           |    | 0000G     | CF 9F 00142       | PUSHAB | LIB\$GL_LIB TL              |      |
|    | 00000000G | 00 |           | 03 FB 00146       | CALLS  | #3, LBR\$LOOKUP_KEY         |      |
|    |           | 48 |           | 50 E9 0014D       | BLBC   | R0, 13\$                    |      |
| 24 | 0000G     | CF |           | 05 E1 00150       | BBC    | #5, LIB\$GL_CTLMSK+1, 12\$  | 1475 |
|    |           |    | 0000V     | CF 9F 00156       | PUSHAB | DELSYM                      | 1481 |
|    |           |    | 38        | A7 9F 0015A       | PUSHAB | OLDMODRFA                   |      |
|    |           |    | 0000G     | CF 9F 0015D       | PUSHAB | LIB\$GL_OBJGSDIX            |      |
|    |           |    | 0000G     | CF 9F 00161       | PUSHAB | LIB\$GL_LIBCTL              |      |
|    | 00000000G | 00 |           | 04 FB 00165       | CALLS  | #4, LBR\$SEARCH             |      |
|    |           | 40 |           | 01 D0 0016C       | MOVL   | #1, REPLACING               | 1482 |
|    |           | E4 | 00000000G | 8F D0 00170       | MOVL   | #LIB\$ REPLACED, OPERATION  | 1483 |
|    |           |    |           | 1E 11 00178       | BRB    | 13\$                        | 1475 |
| 7E | 0000G     | CF |           | 10 C1 0017A 12\$: | ADDL3  | #16, LIB\$GL_LIBFDB, -(SP)  | 1487 |
| 7E |           | 69 |           | 10 C1 00180       | ADDL3  | #16, LIB\$GL_INPFDB, -(SP)  | 1486 |
|    |           |    | 10        | A7 9F 00184       | PUSHAB | MODNAMING                   |      |
|    |           |    |           | 03 DD 00187       | PUSHL  | #3                          | 1487 |
|    |           |    | 00000000G | 8F DD 00189       | PUSHL  | #LIB\$ DUPMODULE            |      |
|    |           | 68 |           | 05 FB 0018F       | CALLS  | #5, LIB\$SIGNAL             |      |
|    | F0        | A7 |           | 01 D0 00192       | MOVL   | #1, DUPSEEN                 | 1488 |
|    |           |    |           | 08 11 00196       | BRB    | 14\$                        | 1489 |
|    | 0000V     | CF |           | 00 FB 00198 13\$: | CALLS  | #0, COPYREC                 | 1492 |
|    |           | 03 |           | 50 E9 0019D       | BLBC   | STATUS, 15\$                |      |
|    |           | 50 |           | 01 D0 001A0 14\$: | MOVL   | #1, R0                      | 1494 |
|    |           |    |           | 04 001A3 15\$:    | RET    |                             | 1495 |

: Routine Size: 420 bytes, Routine Base: \$CODE\$ + 0212

```

: 441      1496 1 %SBTTL 'delsym';
: 442      1497 1
: 443      1498 1 ROUTINE delsym (keydesc) =
: 444      1499 2 BEGIN
: 445      1500 2
: 446      1501 2 | This routine is called by LBR$SEARCH for all globals that are in the module
: 447      1502 2 | about to be replaced. The names will be put on delist which will be scanned
: 448      1503 2 | by prosymbol.
: 449      1504 2 |
: 450      1505 2 MAP
: 451      1506 2     keydesc : REF BBLOCK;
: 452      1507 2
: 453      1508 2 LOCAL
: 454      1509 2     keynb : REF BBLOCK;
: 455      1510 2
: 456      1511 2 perform (lib_get_mem (lnb$w_fixedsize + .keydesc [dsc$w_length], keynb));
: 457      1512 2 keynb [lnb$b_nam[ng]] = .keydesc [dsc$w_length];
: 458      1513 2 keynb [lnb$b_flags] = 0;
: 459      1514 2 CH$MOVE (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer], keynb [lnb$t_name]);
: 460      1515 2 INSQUE (.keynb, .delist [1]);
: 461      1516 2 RETURN true
: 462      1517 1 END;

```

!Of delsym

|  |       |    |       |       |         |        |                      |        |
|--|-------|----|-------|-------|---------|--------|----------------------|--------|
|  |       |    | 007C  | 0000  | DELSYM: | .WORD  | Save R2,R3,R4,R5,R6  | : 1498 |
|  | 5E    |    | 04    | C2    |         | SUBL2  | #4, SP               |        |
|  |       |    | 5E    | DD    |         | PUSHL  | SP                   | : 1511 |
|  | 52    | 04 | AC    | D0    |         | MOVL   | KEYDESC, R2          |        |
|  | 7E    |    | 62    | 3C    |         | MOVZWL | (R2), -(SP)          |        |
|  | 6E    |    | 0A    | C0    |         | ADDL2  | #10, (SP)            |        |
|  | 0000G |    | 02    | FB    |         | CALLS  | #2, LIB_GET_MEM      |        |
|  | 18    |    | 50    | E9    |         | BLBC   | STATUS, -1\$         |        |
|  | 56    |    | 6E    | D0    |         | MOVL   | KEYNB, R6            | : 1512 |
|  | 09    |    | 62    | 90    |         | MOVB   | (R2), 9(R6)          |        |
|  |       |    | A6    | 94    |         | CLRB   | 8(R6)                | : 1513 |
|  | 0A    | A6 | 04    | B2    | 08      | MOVC3  | (R2), @4(R2), 10(R6) | : 1514 |
|  |       |    | 0000' | DF    |         | INSQUE | (R6), @DELIST+4      | : 1515 |
|  |       |    | 50    | D0    |         | MOVL   | #1, R0               | : 1516 |
|  |       |    | 04    | 00031 | 1\$:    | RET    |                      | : 1517 |

: Routine Size: 50 bytes, Routine Base: \$CODE\$ + 03B6

```

: 464      1518 1 %SBTTL 'protir';
: 465      1519 1
: 466      1520 1 ROUTINE prctir =
: 467      1521 2 BEGIN
: 468      1522 2
: 469      1523 2 | This routine processes TIR records. The OBJTIR flag is set in
: 470      1524 2 | the module flags byte and the record is copied.
: 471      1525 2
: 472      1526 2 moduleflags = mhd$m_objtir;
: 473      1527 2 RETURN prorec ()
: 474      1528 1 END;

```

! Of protir

|          |             |               |                 |        |
|----------|-------------|---------------|-----------------|--------|
| 0000' CF | 0000 00000  | PROTIR: .WORD | Save nothing    | : 1520 |
| 0000V CF | 02 90 00002 | MOVB          | #2, MODULEFLAGS | : 1526 |
|          | 00 FB 00007 | CALLS         | #0, PROREC      | : 1527 |
|          | 04 0000C    | RET           |                 | : 1528 |

; Routine Size: 13 bytes, Routine Base: \$CODE\$ + 03E8

```
progsd
: 476 1520 1 %SBTTL 'progsd';
: 477 1530 1
: 478 1531 1 ROUTINE progsd =
: 479 1532 2 BEGIN
: 480 1533 2
: 481 1534 2 ++
: 482 1535 2 Verify GSD records and dispatch on the sub-types:
: 483 1536 2 (0) P-SECTION definition
: 484 1537 2 (1) Symbol definition/reference
: 485 1538 2 (2) Entry point definition
: 486 1539 2 (3) Procedure declaration
: 487 1540 2 (4) Symbol definition with word psect
: 488 1541 2 (5) Entry point definition with word psect
: 489 1542 2 (6) Procedure definition with word psect
: 490 1543 2 (7) Random entity check
: 491 1544 2 (8) Environment definition
: 492 1545 2 (9) Local symbol definition/reference
: 493 1546 2 (10) Local symbol entry point definition
: 494 1547 2 (11) Local symbol procedure definition
: 495 1548 2 (12) Shareable image psect definition
: 496 1549 2
: 497 1550 2 --
: 498 1551 2
: 499 1552 2 RIND
: 500 1553 2 gsddispatch = PLIT (
: 501 1554 2     propsectdef,      | index          structure name
: 502 1555 2     symbols,         | gsd_psc        gps$
: 503 1556 2     entpnts,         | gsd_sym        gsy$, srf$, sdf$
: 504 1557 2     procedef,       | gsd_epm
: 505 1558 2     symbols,         | gsd_pro        pro$, fml$, arg$
: 506 1559 2     pro_epmw,       | gsd_symw       sdfw$
: 507 1560 2     procedef,       | gsd_epmw
: 508 1561 2     pro_idc,        | gsd_prow
: 509 1562 2     pro_env,        | gsd_idc
: 510 1563 2     pro_lsy,        | gsd_env
: 511 1564 2     pro_lepm,       | gsd_lsy
: 512 1565 2     pro_lpro,       | gsd_lepm
: 513 1566 2     pro_spsec,     | gsd_lpro
: 514 1567 2     ) : VECTOR;
: 515 1568 2
: 516 1569 2 LOCAL
: 517 1570 2     gsdtype;
: 518 1571 2
: 519 1572 2 perform (seqchk ());
: 520 1573 2 gsdoffset = obj$c_subtyp;
: 521 1574 2
: 522 1575 2 WHILE .gsdoffset LSSU .reclng DO
: 523 1576 3 BEGIN
: 524 1577 3     IF ( gsdtype = .objvec [.gsdoffset]) GEQU .gsddispatch [-1]
: 525 1578 4     THEN BEGIN
: 526 1579 4         SIGNAL (lib$gsdtyp, 3, modnamlng,
: 527 1580 4             lib$gl_inpfdb [fdb$l_namdesc], .gsdtype);
: 528 1581 4         RETURN lib$gsdtyp;
: 529 1582 4     END
: 530 1583 3     ELSE
: 531 1584 3     perform (( .gsddispatch [.gsdtype]) ());
: 532 1585 2 END;
```

```

: 533      1586  2
: 534      1587  2 IF NOT .lib$gl ctlmsk [lib$v_shrstb]
: 535      1588  2   THEN RETURN copyrec ()
: 536      1589  2   ELSE RETURN true;
: 537      1590  2
: 538      1591  1 END;           ! Of progsd
    
```

```

                                .PSECT $SPLITS,NOWRT,NOEXE,2
                                .LONG 13
00000000V 00000000V 00000000V 00000000V 00000000V 00000000V 00064 P.AAG: .ADDRESS PROPSECTDEF, SYMBOLS, ENTPNTS, PROCEDEF, -
00000000V 00000000V 00000000V 00000000V 00000000V 00000000V 0007C   SYMBOLS, PRO_EPMW, PROCEDEF, PRO_IDC, -
                                00000000V 00094   PRO_ENV, PRO_LSY, PRO_LEPM, PRO_[PRO, -
                                PRO_SPSC
                                GSDDISPATCH= P.AAG
                                .PSECT $CODE$,NOWRT,2
                                PROGSD: .WORD Save R2,R3,R4 : 1531
                                54 00000000G 8F D0 00002   MOVL #LIB$ GSDTYP, R4
                                53 0000'  CF 9E 00009   MOVAB GSDOFFSET, R3
                                0000V  CF 00 FB 0000E   CALLS #0, SEQCHK : 1572
                                50 50 E9 00013   BLBC STATUS, 5$
                                63 01 D0 00016   MOVL #1, GSDOFFSET : 1573
                                10 00 ED 00019 1$: CMPZV #0, #16, RECLNG, GSDOFFSET : 1575
                                36 1B 0001F   BLEQU 3$
                                50 0C A3 63 C1 00021   ADDL3 GSDOFFSET, OBJVEC, R0 : 1577
                                52 60 9A 00026   MOVZBL (R0), GSDTYPE
                                0000' CF 52 D1 00029   Cmpl GSDTYPE, GSDDISPATCH-4
                                1A 1F 0002E   BLSSU 2$
                                7E 0000G CF 52 DD 00030   PUSHL GSDTYPE : 1580
                                10 C1 00032   ADDL3 #16, LIB$GL_INPFDB, -(SP)
                                1C A3 9F 00038   PUSHAB MODNAMLNG : 1579
                                03 DD 0003B   PUSHL #3 : 1580
                                54 DD 0003D   PUSHL R4
                                00000000G 00 05 FB 0003F   CALLS #5, LIB$SIGNAL
                                50 54 D0 00046   MOVL R4, R0 : 1581
                                04 00049   RET
                                50 0000'CF42 D0 0004A 2$: MOVL GSDDISPATCH[GSDTYPE], R0 : 1584
                                60 00 FB 00050   CALLS #0, (R0)
                                C3 50 E8 00053   BLBS STATUS, 1$
                                04 00056   RET
                                06 0000G CF 05 E0 00057 3$: BBS #5, LIB$GL_CTLMSK, 4$ : 1587
                                0000V CF 00 FB 0005D   CALLS #0, COPYREC : 1588
                                04 00062   RET : 1589
                                50 01 D0 00063 4$: MOVL #1, R0
                                04 00066 5$: RET : 1591
    
```

; Routine Size: 103 bytes, Routine Base: \$CODE\$ + 03F5





LIB INPUTOBJ  
V04=000 propsectdef

; Routine Size: 72 bytes, Routine Base: \$CODES + 045C

B 13  
16-Sep-1984 01:57:57  
14-Sep-1984 12:38:04

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

Page 23  
(9)

Li  
VF

```

: 570      1621 1 %SBTTL 'symbols';
: 571      1622 1
: 572      1623 1 ROUTINE symbols =
: 573      1624 2 BEGIN
: 574      1625 2 !
: 575      1626 2 LOCAL
: 576      1627 2     length;
: 577      1628 2 BIND
: 578      1629 2     symbolrec = objvec [.gsdoffset] : BBLOCK;
: 579      1630 2
: 580      1631 2
: 581      1632 2 IF NOT .symbolrec [gsy$v_def]
: 582      1633 2 THEN BEGIN
: 583      1634 2     length = $BYTEOFFSET(srf$t_name) - $BYTEOFFSET(srf$t_start) +
: 584      1635 2         .symbolrec [srf$b_namlng];
: 585      1636 2     symbolstring = symbolrec [srf$b_namlng];           ! Point to the symbol string
: 586      1637 2     END
: 587      1638 2
: 588      1639 2 ELSE
: 589      1640 2 BEGIN
: 590      1641 2 IF .objvec [.gsdoffset] EQL obj$c_gsd_symw           ! If word psect
: 591      1642 2 THEN
: 592      1643 2     BEGIN
: 593      1644 2     length = $BYTEOFFSET(sdfw$t_name) - $BYTEOFFSET(sdfw$t_start) +
: 594      1645 2         .symbolrec [sdfw$b_namlng];
: 595      1646 2     symbolstring = symbolrec [sdfw$b_namlng];       ! Point to the symbol
: 596      1647 2     END
: 597      1648 2 ELSE
: 598      1649 2 BEGIN
: 599      1650 2     length = $BYTEOFFSET(sdf$t_name) - $BYTEOFFSET(sdf$t_start) +
: 600      1651 2         .symbolrec [sdf$b_namlng];
: 601      1652 2     symbolstring = symbolrec [sdf$b_namlng];       ! Point to the symbol
: 602      1653 2     END;
: 603      1654 2 IF NOT .symbolrec [gsy$v_weak]
: 604      1655 2 THEN
: 605      1656 2     perform (prosymbol ());
: 606      1657 2 END;
: 607      1658 2 gsdoffset = .gsdoffset + .length;           ! Update the gsd offset for next
: 608      1659 2 RETURN true
: 609      1660 1 END;                                           !Of symbols

```

```

: 1623
: 1629
: 1632
: 1634
: 1636
: 1632
: 1641
: 1644
: 1644

```

|    |    |    |       |        |                     |       |                |                       |        |
|----|----|----|-------|--------|---------------------|-------|----------------|-----------------------|--------|
| 50 | 08 | A3 | 0000' | CF     | 9E                  | 00002 | SYMBOLS: .WORD | Save R2,R3            | : 1623 |
| 0D | 02 | A0 | FC    | A3     | C1                  | 00007 | MOVAB          | SYMBOLSTRING, R3      | : 1629 |
|    |    | 52 | 04    | 01     | E0                  | 0000D | ADDL3          | GSDOFFSET, OBJVEC, R0 | : 1632 |
|    |    | 52 | 04    | A0     | 9A                  | 00012 | BBS            | #1, 2(R0), 1\$        | : 1634 |
|    |    | 63 | 04    | 05     | C0                  | 00016 | MOVZBL         | 4(R0), LENGTH         | : 1636 |
|    |    | 04 |       | ADDL2  | #5, LENGTH          |       |                |                       | : 1632 |
|    |    | 04 |       | MOVAB  | 4(R0), SYMBOLSTRING |       |                |                       | : 1641 |
|    |    | 52 | 0A    | BRB    | 4\$                 |       |                |                       | : 1644 |
|    |    | 52 | 0A    | 60     | 91                  | 0001D | 1\$:           | CMPB (R0), #4         |        |
|    |    |    |       | 0D     | 12                  | 00022 |                | BNEQ 2\$              |        |
|    |    |    |       | MOVZBL | 10(R0), LENGTH      |       |                |                       |        |
|    |    |    |       | ADDL2  | #11, LENGTH         |       |                |                       |        |

|       |    |    |    |       |       |      |        |                      |   |      |
|-------|----|----|----|-------|-------|------|--------|----------------------|---|------|
|       | 63 | 0A | A0 | 9E    | 0002B |      | MOVAB  | 10(R0), SYMBOLSTRING | : | 1646 |
|       |    |    | 0B | 11    | 0002F |      | BRB    | 3\$                  | : | 1641 |
|       | 52 | 09 | A0 | 9A    | 00031 | 2\$: | MOVZBL | 9(R0), LENGTH        | : | 1650 |
|       | 52 |    | 0A | C0    | 00035 |      | ADDL2  | #10, LENGTH          | : |      |
|       | 63 | 09 | A0 | 9E    | 00038 |      | MOVAB  | 9(R0), SYMBOLSTRING  | : | 1652 |
|       | 08 | 02 | A0 | E8    | 0003C | 3\$: | BLBS   | 2(R0), 4\$           | : | 1654 |
| 0000V | CF |    | 00 | FB    | 00040 |      | CALLS  | #0, PROSYMBOL        | : | 1656 |
|       | 07 |    | 50 | E9    | 00045 |      | BLBC   | STATUS, 5\$          | : |      |
| FC    | A3 |    | 52 | C0    | 00048 | 4\$: | ADDL2  | LENGTH, GSDOFFSET    | : | 1658 |
|       | 50 |    | 01 | D0    | 0004C |      | MOVL   | #1, R0               | : | 1659 |
|       |    |    | 04 | 0004F | 5\$:  |      | RET    |                      | : | 1660 |

; Routine Size: 80 bytes, Routine Base: \$CODE\$ + 04A4

```

entpnts
: 611 1661 1 %SBTTL 'entpnts';
: 612 1662 1
: 613 1663 1 ROUTINE entpnts =
: 614 1664 2 BEGIN
: 615 1665 2 !
: 616 1666 2 LOCAL
: 617 1667 2 length;
: 618 1668 2 BIND
: 619 1669 2 symbolrec = objvec [.gsdoffset] : BBLOCK;
: 620 1670 2
: 621 1671 2
: 622 1672 2 length = $BYTEOFFSET(epm$t_name) - $BYTEOFFSET(epm$t_start) +
: 623 1673 2 .symbolrec [epm$b_namlng];
: 624 1674 2 symbolstring = symbolrec [epm$b_namlng]; ! Point to the symbol
: 625 1675 2 perform (p-osymbol ());
: 626 1676 2 gsdoffset = .gsdoffset + .length; ! Else update the offset for next
: 627 1677 2 RETURN true
: 628 1678 1 END; ! Of entpnts

```

|    |       |    |       |             |                |                       |        |
|----|-------|----|-------|-------------|----------------|-----------------------|--------|
|    |       |    |       | 000C 00000  | ENTPNTS: .WORD | Save R2,R3            | : 1663 |
|    |       | 53 | 0000' | CF 9E 00002 | MOVAB          | GSDOFFSET, R3         | : 1669 |
| 50 | 0C    | A3 |       | 63 C1 00007 | ADDL3          | GSDOFFSET, OBJVEC, R0 | : 1672 |
|    |       | 52 | OB    | A0 9A 0000C | MOVZBL         | 11(R0), LENGTH        | : 1674 |
|    |       | 52 |       | 0C C0 00010 | ADDL2          | #12, LENGTH           | : 1675 |
|    | 04    | A3 | OB    | A0 9E 00013 | MOVAB          | 11(R0), SYMBOLSTRING  | : 1676 |
|    | 0000V | CF |       | 00 FB 00018 | CALLS          | #0, PROSYMBOL         | : 1677 |
|    |       | 06 |       | 50 E9 0001D | BLBC           | STATUS, 1\$           | : 1678 |
|    |       | 63 |       | 52 r0 00020 | ADDL2          | LENGTH, GSDOFFSET     | : 1679 |
|    |       | 50 |       | 01 D0 00023 | MOVL           | #1, R0                | : 1680 |
|    |       |    |       | 04 00026    | 1\$: RET       |                       | : 1681 |

: Routine Size: 39 bytes, Routine Base: \$CODE\$ + 04F4

```

procedef
: 630 1679 1 %SBTTL 'procedef';
: 631 1680 1
: 632 1681 1 ROUTINE procedef =
: 633 1682 2 BEGIN
: 634 1683 2
: 635 1684 2 A procedure definition is an extended entry point definition, carrying with
: 636 1685 2 it a description of the procedure's formal arguments. processing these consists
: 637 1686 2 in normal symbol definition processing followed by:-
: 638 1687 2 (1) Validation of the format of formal description (i.e. just check
: 639 1688 2 that minimum number of arguments specified is less than
: 640 1689 2 or equal to the maximum.
: 641 1690 2
: 642 1691 2
: 643 1692 2 LOCAL
: 644 1693 2 argcount;
: 645 1694 2
: 646 1695 2 IF .objvec [.gsdoffset] EQL obj$c_gsd_prow
: 647 1696 2 THEN
: 648 1697 2 perform (pro_epmw ())
: 649 1698 2 ELSE
: 650 1699 2 perform (entpnts ());
: 651 1700 2
: 652 1701 2 BEGIN
: 653 1702 2 BIND
: 654 1703 2 formals = objvec [.gsdoffset] : BBLOCK;
: 655 1704 2 gsdoffset = .gsdoffset + fml$c_size; ! Update record pointer
: 656 1705 2 IF (argcount = .formals [fml$b_maxargs]) NEQ 0 ! If args
: 657 1706 2 THEN INCRU i FROM 1 TO .argcount ! then skip them
: 658 1707 2 DO BEGIN
: 659 1708 2 BIND
: 660 1709 2 argdesc = objvec [.gsdoffset] : BBLOCK;
: 661 1710 2
: 662 1711 2 gsdoffset = .gsdoffset + .argdesc [arg$b_bytecnt] + arg$c_size;
: 663 1712 2 END;
: 664 1713 2 RETURN true
: 665 1714 2 END;
: 666 1715 1 END; ! Of procedef

```

| 000C 0000 PROCEDEF: |       |    |       |                  |        |                       |        |
|---------------------|-------|----|-------|------------------|--------|-----------------------|--------|
|                     |       | 53 | 0000' | CF 9E 00002      | .WORD  | Save R2,R3            | : 1681 |
| 50                  | 0C    | A3 |       | 63 C1 00007      | MOVAB  | GSDOFFSET, R3         | : 1695 |
|                     |       | 06 |       | 60 91 0000C      | ADDL3  | GSDOFFSET, OBJVEC, R0 |        |
|                     |       |    |       | 07 12 0000F      | CMPB   | (R0), #6              |        |
|                     | 0000V | CF |       | 00 FB 00011      | BNEQ   | 1\$                   | : 1697 |
|                     |       |    |       | 04 11 00016      | CALLS  | #0, PRO_EPMW          |        |
|                     |       | BD |       | 00 FB 00018 1\$: | BRB    | 2\$                   | : 1699 |
|                     |       | 2D |       | 50 E9 0001C 2\$: | CALLS  | #0, ENTPNTS           |        |
| 50                  | 0C    | A3 |       | 63 C1 0001F      | BLBC   | STATUS, 6\$           | : 1703 |
|                     |       | 63 |       | 02 C0 00024      | ADDL3  | GSDOFFSET, OBJVEC, R0 | : 1704 |
|                     |       | 52 | 01    | A0 9A 00027      | ADDL2  | #2, GSDOFFSET         | : 1705 |
|                     |       |    |       | 1C 13 0002B      | MOVZBL | 1(R0), ARGCOUNT       |        |
|                     |       | 51 |       | 01 D0 0002D      | BEQL   | 5\$                   | : 1706 |
|                     |       |    |       |                  | MOVL   | #1, I                 |        |

|    |    |    |    |    |       |       |      |        |                       |  |        |
|----|----|----|----|----|-------|-------|------|--------|-----------------------|--|--------|
| 50 | 0C | A3 |    | 12 | 11    | 00030 |      | BRB    | 4\$                   |  |        |
|    |    | 50 | 01 | 63 | C1    | 00032 | 3\$: | ADDL3  | GSDOFFSET, OBJVEC, R0 |  | : 1709 |
|    |    | 50 |    | A0 | 9A    | 00037 |      | MOVZBL | 1(R0), R0             |  | : 1711 |
|    |    | 63 | 02 | 63 | C0    | 0003B |      | ADDL2  | GSDOFFSET, R0         |  |        |
|    |    | 63 |    | A0 | 9E    | 0003E |      | MOVAB  | 2(R0), GSDOFFSET      |  |        |
|    |    | 52 |    | 51 | D6    | 00042 |      | INCL   | I                     |  | : 1706 |
|    |    | 50 |    | 51 | D1    | 00044 | 4\$: | CMPL   | I, ARGCOUNT           |  |        |
|    |    |    |    | E9 | 1B    | 00047 |      | BLEQU  | 3\$                   |  |        |
|    |    |    |    | 01 | D0    | 00049 | 5\$: | MOVL   | #1, R0                |  | : 1713 |
|    |    |    |    | 04 | 0004C | 6\$:  |      | RET    |                       |  | : 1715 |

; Routine Size: 77 bytes, Routine Base: \$CODE\$ + 051B

```

: 668      1716 1 %SBTTL 'pro_epmw';
: 669      1717 1
: 670      1718 1 ROUTINE pro_epmw =
: 671      1719 2 BEGIN
: 672      1720 2
: 673      1721 2 | Process entry points with word psect
: 674      1722 2 |
: 675      1723 2 LOCAL
: 676      1724 2 length;
: 677      1725 2 BIND
: 678      1726 2 symbolrec = objvec [.gsdoffset] : BBLOCK;
: 679      1727 2
: 680      1728 2
: 681      1729 2 length = $BYTEOFFSET(epmw$st_name) - $BYTEOFFSET(epmw$st_start) +
: 682      1730 2 .symbolrec [epmw$b_namlng];
: 683      1731 2 symbolstring = symbolrec [epmw$b_namlng]; ! Point to the symbol
: 684      1732 2 perform (prosymbol ());
: 685      1733 2 gsdoffset = .gsdoffset + .length; ! Else update the offset for next
: 686      1734 2 RETURN true
: 687      1735 1 END; ! Of pro_epmw

```

000C 00000 PRO\_EPMW:

|    |       |    |       |               |        |                       |        |
|----|-------|----|-------|---------------|--------|-----------------------|--------|
|    |       | 53 | 0000' | CF 9E 00002   | .WORD  | Save R2,R3            | : 1718 |
| 50 | 0C    | A3 |       | 63 C1 00007   | MOVAB  | GSDOFFSET, R3         | : 1726 |
|    |       | 52 | 0C    | A0 9A 0000C   | ADDL3  | GSDOFFSET, OBJVEC, R0 | : 1729 |
|    |       | 52 |       | 0D C0 00010   | MOVZBL | 12(R0), LENGTH        | : 1729 |
|    | 04    | A3 | 0C    | A0 9E 00013   | ADDL2  | #13, LENGTH           | : 1731 |
|    | 0000V | CF |       | 00 FB 00018   | MOVAB  | 12(R0), SYMBOLSTRING  | : 1732 |
|    |       | 06 |       | 50 E9 0001D   | CALLS  | #0, PROSYMBOL         | : 1732 |
|    |       | 63 |       | 52 C0 00020   | BLBC   | STATUS, 1\$           | : 1733 |
|    |       | 50 |       | 01 D0 00023   | ADDL2  | LENGTH, GSDOFFSET     | : 1733 |
|    |       |    |       | 04 00026 1\$: | MOVL   | #1, R0                | : 1734 |
|    |       |    |       |               | RET    |                       | : 1735 |

: Routine Size: 39 bytes. Routine Base: \$CODE\$ + 0568

: 688 1736 1

```

: 690      1737 1 %SBTTL 'pro_idc';
: 691      1738 1
: 692      1739 1 ROUTINE pro_idc =
: 693      1740 2 BEGIN
: 694      1741 2
: 695      1742 2         Process random entity check
: 696      1743 2         by skipping it.
: 697      1744 2
: 698      1745 2 LOCAL
: 699      1746 2     identstring : REF VECTOR [,BYTE],    ! pointer to ident string
: 700      1747 2     objectname : REF VECTOR [,BYTE],   ! pointer to object name string
: 701      1748 2     length;
: 702      1749 2 BIND
: 703      1750 2     idc_rec = objvec [.gsdoffset] : BBLOCK;
: 704      1751 2
: 705      1752 2     identstring = idc_rec [idc$b_namlng] + 1 + .idc_rec [idc$b_namlng];
: 706      1753 2     objectname = identstring [1] + .identstring [0];
: 707      1754 2     length = objectname [1] + .objectname [0] - idc_rec;
: 708      1755 2     gsdoffset = .gsdoffset + .length;
: 709      1756 2 RETURN true
: 710      1757 1 END;

```

! Of pro\_idc

|    |       |    |       |      |       |       |                |                                |        |
|----|-------|----|-------|------|-------|-------|----------------|--------------------------------|--------|
| 52 | 0000' | CF | 0000' | CF   | C1    | 00002 | PRO_IDC: .WORD | Save R2                        | : 1739 |
|    |       | 50 | 03    | A2   | 9A    | 0000A | ADDL3          | GSDOFFSET, OBJVEC, R2          | : 1750 |
|    |       | 50 | 04    | A042 | 9E    | 0000E | MOVZBL         | 3(R2), R0                      | : 1752 |
|    |       | 51 |       | 60   | 9A    | 00013 | MOVAB          | 4(R0)[R2], IDENTSTRING         |        |
|    |       | 50 | 01    | A140 | 9E    | 00016 | MOVZBL         | (IDENTSTRING), R1              | : 1753 |
|    |       | 51 |       | 60   | 9A    | 0001B | MOVAB          | 1(R1)[IDENTSTRING], OBJECTNAME |        |
|    |       | 50 |       | 51   | C0    | 0001E | MOVZBL         | (OBJECTNAME), R1               | : 1754 |
|    |       | 50 |       | 52   | C2    | 00021 | ADDL2          | R1, OBJECTNAME                 |        |
|    |       |    |       | 50   | D6    | 00024 | SUBL2          | R2, R0                         |        |
|    | 0000' | CF |       | 50   | C0    | 00026 | INCL           | LENGTH                         |        |
|    |       | 50 |       | 01   | D0    | 0002B | ADDL2          | LENGTH, GSDOFFSET              | : 1755 |
|    |       |    |       | 04   | 0002E |       | MOVL           | #1, R0                         | : 1756 |
|    |       |    |       |      |       |       | RET            |                                | : 1757 |

: Routine Size: 47 bytes, Routine Base: \$CODE\$ + 058F

: 711 1758 1



```

: 713      1759  1 %SBTTL  'pro_env';
: 714      1760  1
: 715      1761  1 ROUTINE pro_env =
: 716      1762  2 BEGIN
: 717      1763  2
: 718      1764  2
: 719      1765  2
: 720      1766  2
: 721      1767  2 LOCAL
: 722      1768  2 length;
: 723      1769  2 BIND
: 724      1770  2 env_rec = objvec [.gsdoffset] : BBLOCK;
: 725      1771  2
: 726      1772  2
: 727      1773  2 length = env_rec [env$st_name] - objvec [.gsdoffset] +
: 728      1774  2 .env_rec [env$b_namlng];
: 729      1775  2 gsdoffset = .gsdoffset + .length;
: 730      1776  2 RETURN true
: 731      1777  1 END;

```

! Of pro\_env

|    |       |    |       |    |    |       |                |                       |        |
|----|-------|----|-------|----|----|-------|----------------|-----------------------|--------|
| 50 | 0000' | CF | 0000' | CF | C1 | 00002 | PRO_ENV: .WORD | Save R2               | : 1761 |
| 51 |       | 50 |       | 50 | C3 | 0000A | ADDL3          | GSDOFFSET, OBJVEC, R0 | : 1770 |
|    |       | 52 | 05    | A0 | 9A | 0000E | SUBL3          | R0, R0, R1            | : 1773 |
|    |       | 51 |       | 52 | C0 | 00012 | MOVZBL         | 5(R0), R2             | : 1774 |
|    |       | 50 | 06    | A1 | 9E | 00015 | ADDL2          | R2, R1                | : 1773 |
|    | 0000' | CF |       | 50 | C0 | 00019 | MOVAB          | 6(R1), LENGTH         | : 1775 |
|    |       | 50 |       | 01 | D0 | 0001E | ADDL2          | LENGTH, GSDOFFSET     | : 1776 |
|    |       |    |       |    | 04 | 00021 | MOVL           | #1, R0                | : 1777 |
|    |       |    |       |    |    |       | RET            |                       |        |

; Routine Size: 34 bytes, Routine Base: \$CODE\$ + 05BE

; 732 1778 1

```

: 734      1779  1 %SBTTL 'pro_lsy';
: 735      1780  1
: 736      1781  1 ROUTINE pro_lsy =
: 737      1782  2 BEGIN
: 738      1783  2 |
: 739      1784  2 |         Process local symbol definition/reference
: 740      1785  2 |         by skipping it.
: 741      1786  2 |
: 742      1787  2 LOCAL
: 743      1788  2     length;
: 744      1789  2 BIND
: 745      1790  2     lsy_rec = objvec [.gsdoffset] : BBLOCK;
: 746      1791  2
: 747      1792  2 IF NOT .lsy_rec [lsy$v_def]
: 748      1793  2 THEN
: 749      1794  2     length = $BYTEOFFSET(lsrfs_name) - $BYTEOFFSET(lsrfs_start) +
: 750      1795  2     .lsy_rec [lsrfs_b_namlng]
: 751      1796  2 ELSE
: 752      1797  2     length = $BYTEOFFSET(lsdofs_name) - $BYTEOFFSET(lsdofs_start) +
: 753      1798  2     .lsy_rec [lsdofs_b_namlng];
: 754      1799  2 gsdoffset = .gsdoffset + .length;
: 755      1800  2 RETURN true
: 756      1801  1 END;

```

! Of pro\_lsy

|    |       |    |       |      |          |                |                       |        |
|----|-------|----|-------|------|----------|----------------|-----------------------|--------|
|    |       |    |       | 0000 | 0000     | PRO_LSY: .WORD | Save nothing          | : 1781 |
| 50 | 0000' | CF | 0000' | CF   | C1 00002 | ADDL3          | GSDOFFSET, OBJVEC, R0 | : 1790 |
| 09 | 02    | A0 |       | 01   | E0 0000A | BBS            | #1, 2(R0), 1\$        | : 1792 |
|    |       | 50 | 06    | A0   | 9A 0000F | MOVZBL         | 6(R0), LENGTH         | : 1794 |
|    |       | 50 |       | 07   | C0 00013 | ADDL2          | #7, LENGTH            |        |
|    |       |    |       | 07   | 11 00016 | BRB            | 2\$                   |        |
|    |       | 50 | 0C    | A0   | 9A 00018 | MOVZBL         | 12(R0), LENGTH        | : 1797 |
|    |       | 50 |       | 0D   | C0 0001C | ADDL2          | #13, LENGTH           |        |
|    | 0000' | CF |       | 50   | C0 0001F | ADDL2          | LENGTH, GSDOFFSET     | : 1799 |
|    |       | 50 |       | 01   | D0 00024 | MOVL           | #1, R0                | : 1800 |
|    |       |    |       | 04   | 00027    | RET            |                       | : 1801 |

: Routine Size: 40 bytes, Routine Base: \$CCDES + 05E0

: 757 1802 1

```

: 759      1803 1 %SBTTL 'pro_lepm';
: 760      1804 1
: 761      1805 1 ROUTINE pro_lepm =
: 762      1806 2 BEGIN
: 763      1807 2
: 764      1808 2          Process local symbol entry point definition
: 765      1809 2          by skipping it.
: 766      1810 2
: 767      1811 2 LOCAL
: 768      1812 2     length;
: 769      1813 2 BIND
: 770      1814 2     lepm_rec = objvec [.gsdoffset] : BBLOCK;
: 771      1815 2
: 772      1816 2
: 773      1817 2     length = $BYTEOFFSET(lepm$st_name) - $BYTEOFFSET(lepm$st_start) +
: 774      1818 2     .lepm_rec [lepm$b_namlng];
: 775      1819 2     gsdoffset = .gsdoffset + .length;           ! Else update the offset for next
: 776      1820 2 RETURN true
: 777      1821 1 END;                                     ! Of pro_lepm

```

0000 00000 PRO\_LEPM:

|    |       |    |       |    |          |        |                       |   |      |
|----|-------|----|-------|----|----------|--------|-----------------------|---|------|
| 50 | 0000' | CF | 0000' | CF | C1 00002 | .WORD  | Save nothing          | : | 1805 |
|    |       | 50 | OE    | AO | 9A 0000A | ADDL3  | GSDOFFSET, OBJVEC, R0 | : | 1814 |
|    |       | 50 |       | OF | CO 0000E | MOVZBL | 14(R0), LENGTH        | : | 1817 |
|    | 0000' | CF |       | 50 | CO 00011 | ADDL2  | #15, LENGTH           | : | 1819 |
|    |       | 50 |       | 01 | DO 00016 | MOVL   | LENGTH, GSDOFFSET     | : | 1820 |
|    |       |    |       | 04 | 00019    | RET    | #1, R0                | : | 1821 |

: Routine Size: 26 bytes, Routine Base: \$CODES + 0608

: 778 1822 1

```

: 780      1823 1 %SBTTL 'pro_lpro';
: 781      1824 1
: 782      1825 1 ROUTINE pro_lpro =
: 783      1826 2 BEGIN
: 784      1827 2
: 785      1828 2 | Process local symbol procedure definition
: 786      1829 2 | by skipping it.
: 787      1830 2 |
: 788      1831 2 LOCAL
: 789      1832 2 length;
: 790      1833 2 BIND
: 791      1834 2 lpro_rec = objvec [.gsdoffset] : BBLOCK;
: 792      1835 2
: 793      1836 2
: 794      1837 2 length = $BYTEOFFSET(lpro$t_name) - $BYTEOFFSET(lpro$t_start) +
: 795      1838 2 | .lpro_rec [lpro$b_namlng];
: 796      1839 2 gsdoffset = .gsdoffset + .length; ! Else update the offset for next
: 797      1840 2 RETURN true
: 798      1841 1 END; ! Of pro_lpro

```

```

                                0000 00000 PRO_LPRO:
                                .WORD Save nothing
50 0000' CF 0000' CF C1 00002 ADDL3 GSDOFFSET, OBJVEC, R0 : 1825
                                AE 0E A0 9A 0000A MOVZBL 14(R0), LENGTH : 1834
                                50 OF C0 0000E ADDL2 #15, LENGTH : 1837
0000' CF 50 C0 00011 ADDL2 LENGTH, GSDOFFSET : 1839
                                50 01 D0 00016 MOVL #1, R0 : 1840
                                04 00019 RET : 1841

```

: Routine Size: 26 bytes, Routine Base: \$CODE\$ + 0622

: 799 1842 1

```

: 801      1843 1 %SBTTL 'pro_spisc';
: 802      1844 1
: 803      1845 1 ROUTINE pro_spisc =
: 804      1846 2 BEGIN
: 805      1847 2
: 806      1848 2
: 807      1849 2
: 808      1850 2
: 809      1851 2 LOCAL
: 810      1852 2
: 811      1853 2 BIND
: 812      1854 2
: 813      1855 2
: 814      1856 2
: 815      1857 2
: 816      1858 2
: 817      1859 2 IF .spisc_def [sgps$b_namng] GTRU sym$c_maxng
: 818      1860 2 OR .spisc_def [sgps$b_namng] EQL 0
: 819      1861 2 THEN BEGIN
: 820      1862 2
: 821      1863 2
: 822      1864 2
: 823      1865 2
: 824      1866 2
: 825      1867 2 length = $BYTEOFFSET(sgps$t_name) - $BYTEOFFSET(sgps$t_start) +
: 826      1868 2
: 827      1869 2
: 828      1870 2
: 829      1871 1

```

| 001C 00000 PRO_SPISC: |           |    |           |    |       |       |      |  |        |                            |
|-----------------------|-----------|----|-----------|----|-------|-------|------|--|--------|----------------------------|
|                       |           |    |           |    |       |       |      |  |        | 1845                       |
|                       |           | 54 | 0000'     | CF | 9E    | 00002 |      |  | .WORD  | Save R2,R3,R4              |
|                       |           | 53 | 00000000G | 8F | D0    | 00007 |      |  | MOVAB  | GSDOFFSET, R4              |
| 52                    | 0C        | A4 |           | 64 | C1    | 0000E |      |  | MOVL   | #LIB\$ SPNAMLNG, R3        |
|                       |           | 1F |           | A2 | 91    | 00013 |      |  | ADDL3  | GSDOFFSET, OBJVEC, R2      |
|                       |           |    | 0C        | 05 | 1A    | 00017 |      |  | CMPB   | 12(R2), #31                |
|                       |           |    |           | 05 | 1A    | 00017 |      |  | BGTRU  | 1\$                        |
|                       |           |    | 0C        | A2 | 95    | 00019 |      |  | TSTB   | 12(R2)                     |
|                       |           |    |           | 1C | 12    | 0001C |      |  | BNEQ   | 2\$                        |
|                       |           | 7E | 0C        | A2 | 9A    | 0001E | 1\$: |  | MOVZBL | 12(R2), -(SP)              |
| 7E                    | 0000G     | CF |           | 10 | C1    | 00022 |      |  | ADDL3  | #16, LIB\$GL_INPFDB, -(SP) |
|                       |           |    | 1C        | A4 | 9F    | 00028 |      |  | PUSHAB | MODNAMLNG                  |
|                       |           |    |           | 03 | DD    | 0002B |      |  | PUSHL  | #3                         |
|                       |           |    |           | 53 | DD    | 0002D |      |  | PUSHL  | R3                         |
|                       | 00000000G | 00 |           | 05 | FB    | 0002F |      |  | CALLS  | #5, LIB\$SIGNAL            |
|                       |           | 50 |           | 53 | D0    | 00036 |      |  | MOVL   | R3, R0                     |
|                       |           |    |           | 04 | 00039 |       |      |  | RET    |                            |
|                       |           | 50 | 0C        | A2 | 9A    | 0003A | 2\$: |  | MOVZBL | 12(R2), LENGTH             |
|                       |           | 50 |           | 0D | C0    | 0003E |      |  | ADDL2  | #13, LENGTH                |
|                       |           | 64 |           | 50 | C0    | 00041 |      |  | ADDL2  | LENGTH, GSDOFFSET          |
|                       |           | 50 |           | 01 | D0    | 00044 |      |  | MOVL   | #1, R0                     |
|                       |           |    |           | 04 | 00047 |       |      |  | RET    |                            |
|                       |           |    |           |    |       |       |      |  |        | 1864                       |
|                       |           |    |           |    |       |       |      |  |        | 1867                       |
|                       |           |    |           |    |       |       |      |  |        | 1869                       |
|                       |           |    |           |    |       |       |      |  |        | 1870                       |
|                       |           |    |           |    |       |       |      |  |        | 1871                       |

! Of pro\_spisc

LIB INPUTOBJ  
V04=000 pro\_spsc

: Routine Size: 72 bytes, Routine Base: \$CODE\$ + 063C

: 830 1872 1

B 14  
16-Sep-1984 01:57:57  
14-Sep-1984 12:38:04

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

Page 36  
(19)

LII  
VO

prosymbol

```
832 1873 1 %SBTTL 'prosymbol';
833 1874 1
834 1875 1 ROUTINE prosymbol =
835 1876 2 BEGIN
836 1877 2 |++
837 1878 2 |
838 1879 2 |
839 1880 2 |--
840 1881 2 IF .symbolstring [0] GTRU .lib$gl_keysize ! If the symbol length is outside
841 1882 2 OR .symbolstring [0] EQL 0 ! Legal range
842 1883 2 THEN BEGIN
843 1884 2 | SIGNAL (lib$ symnamlng, 4, symbolstring [0], modnamlng,
844 1885 2 | |lib$gl_inpfdb [fdb$l_namdesc], .symbolstring [0]);
845 1886 2 | RETURN lib$_symnamlng;
846 1887 2 | END;
847 1888 2 IF NOT .lib$gl_ctlmsk [lib$v_globals]
848 1889 2 THEN RETURN true
849 1890 2 ELSE BEGIN
850 1891 2 |
851 1892 2 | LOCAL
852 1893 2 | | status,
853 1894 2 | | replacekey,
854 1895 2 | | keynb : REF BBLOCK,
855 1896 2 | | txtrfa : BBLOCK [rfa$c_length],
856 1897 2 | | keydesc : BBLOCK [dsc$c_s_bln];
857 1898 2 |
858 1899 2 | keydesc [dsc$w_length] = .symbolstring [0];
859 1900 2 | keydesc [dsc$a_pointer] = symbolstring [1];
860 1901 2 | perform (lbr$set_index (lib$gl_libctl, lib$gl_objgsdix,
861 1902 2 | |lib$_indexerr, 1, lib$gl_[libfdb [fdb$l_namdesc]]);
862 1903 2 |
863 1904 2 | | If the symbol is already in the index and we are not replacing, then that is
864 1905 2 | | an error. If we are replacing, it must be from the same module, otherwise
865 1906 2 | | that is an error.
866 1907 2 |
867 1908 2 | IF 'replacekey = lbr$lookup_key (lib$gl_libctl, keydesc, txtrfa) !If key already in index
868 1909 2 | AND (IF .lib$gl_ctlmsk [lib$v_replace]
869 1910 2 | THEN NOT (H$EQL (rfa$c_length, txtrfa, rfa$c_length, oldmodrfa)
870 1911 2 | ELSE true)
871 1912 2 | THEN BEGIN
872 1913 2 | | SIGNAL (lib$_dupglobal, 3, keydesc, lib$gl_inpfdb [fdb$l_namdesc], !Tell user of error
873 1914 2 | | |lib$gl_libfdb [fdb$l_namdesc]);
874 1915 2 | | RETURN lib$_dupglobal;
875 1916 2 | | END;
876 1917 2 |
877 1918 2 | | If replacing the key, look and see if its on the deleted key list. If it is, remove it
878 1919 2 | | from that list, and put on the global list. If not replacing, just put on the global
879 1920 2 | | list.
880 1921 2 |
881 1922 2 | | status = false;
882 1923 2 | | IF NOT (
883 1924 2 | | | IF .replacekey
884 1925 2 | | | THEN BEGIN
885 1926 2 | | | | keynb = delist [0]; !Initialize to search queue
886 1927 2 | | | | WHILE (keynb = .keynb [lnb$l_flink]) NEQ delist [0]
887 1928 2 | | | | DO IF (H$EQL (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer],
888 1929 2 | | | | |.keynb [lnb$b_namlng], keynb [lnb$t_name])
```







LIB\_INPUTOBJ  
V04=000

prosymbol

F 14  
16-Sep-1984 01:57:57  
14-Sep-1984 12:38:04

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

Page 40  
(20)

|    |    |    |    |    |    |    |       |       |       |        |                             |   |      |
|----|----|----|----|----|----|----|-------|-------|-------|--------|-----------------------------|---|------|
|    |    |    | 09 | 50 |    | 6E | D0    | 00125 |       | MOVL   | KEYNB, R0                   | : | 1949 |
|    |    |    | 08 | A0 | 04 | AE | 90    | 00128 |       | MOVB   | KEYDESC, 9(R0)              | : |      |
|    | 0A | A0 |    | BE | 04 | AE | 28    | 0012L |       | MOV3   | KEYDESC, @KEYDESC+4, 10(R0) | : | 1950 |
|    |    |    |    | 50 |    | 6E | D0    | 00134 | 11\$: | MOVL   | KEYNB, R0                   | : | 1952 |
| 08 | A0 |    |    | 00 |    | 56 | F0    | 00137 |       | INSV   | REPLACEKEY, #0, #1, 8(R0)   | : |      |
|    |    |    | 7C | B7 |    | 60 | 0E    | 0013D |       | INSQUE | (R0), @GLOBLIST+4           | : | 1953 |
|    |    |    |    | 50 |    | 01 | D0    | 00141 | 12\$: | MOVL   | #1, R0                      | : | 1955 |
|    |    |    |    |    |    | 04 | 00144 | 13\$: |       | RET    |                             | : | 1956 |

: Routine Size: 325 bytes, Routine Base: \$CODE\$ + 0684

```
proeom
: 917 1957 1 %SBTTL 'proeom';
: 918 1958 1
: 919 1959 1 ROUTINE proeom =
: 920 1960 2 BEGIN
: 921 1961 2
: 922 1962 2 Process end of module records:
: 923 1963 2 (1) Validate sequence
: 924 1964 2 (2) Interpret compiler completion code,
: 925 1965 2 issuing appropriate error or warning message
: 926 1966 2
: 927 1967 2
: 928 1968 2 LOCAL
: 929 1969 2 datadesc : BBLOCK [dsc$c_s_bln],
: 930 1970 2 modnamdesc : BBLOCK [dsc$c_s_bln],
: 931 1971 2 comcode;
: 932 1972 2
: 933 1973 2 maxreclng = obj$c_maxreclsiz; !Reset max record length
: 934 1974 2 perform (seqchk ());
: 935 1975 2 IF (comcode = .objrec [eom$b_comcod]) NEQ 0 ! If non zero compilation cplete code
: 936 1976 2 THEN BEGIN ! CHECK
: 937 1977 2 IF .comcode GTRU 3 THEN comcode = 4; !Make illegal index legal
: 938 1978 2 IF .comcode NEQ 0
: 939 1979 2 THEN SIGNAL (lib$c_comcod, 3, compilecods [.comcode * dsc$c_s_bln,0,0,0], !Signal the error (warning)
: 940 1980 2 modnamlng, lib$gl_inpfdb [fdb$l_namdesc]);
: 941 1981 2 END;
: 942 1982 2 perform (copyrec ());
P 943 1983 2 rms_perform (lbr$put_end (lib$gl_libctl),
: 944 1984 2 lib$_writeerr, .lbr$gl_rmsstv, 1, lib$gl_libfdb [fdb$l_namdesc]);
: 945 1985 2
: 946 1986 2 Update the module header
: 947 1987 2
: 948 1988 2 IF .lib$gl_ctlmsk [lib$v_selective]
: 949 1989 2 THEN moduleflags = .moduleflags OR mhd$m_selsrc;
: 950 1990 2 datadesc [dsc$a_length] = .idlng + 2; !include flag and id length bytes
: 951 1991 2 datadesc [dsc$a_pointer] = moduleflags;
: 952 1992 2 modnamdesc [dsc$a_length] = .modnamlng;
: 953 1993 2 modnamdesc [dsc$a_pointer] = modulename;
P 954 1994 2 rms_perform (lbr$set_module (lib$gl_libctl, modulerfa ,0,0, datadesc),
: 955 1995 2 lib$_mhderr, .lbr$gl_rmsstv, 2, modnamdesc, lib$gl_libfdb [fdb$l_namdesc]);
: 956 1996 2
: 957 1997 2 Insert all the keys now
: 958 1998 2
: 959 1999 2 perform (finish_object (true));
: 960 2000 2
: 961 2001 2 Log operation if logging on console
: 962 2002 2
: 963 2003 2 lib_log_upd (
: 964 2004 2 (IF .operation EQL lib$_replaced THEN lhc$c_replaced ELSE lhc$c_inserted),
: 965 2005 2 modnamdesc ); ! log module name for LUH record
: 966 2006 2 lib_log_op (.operation, modnamdesc, .lib$gl_libfdb); !Log insert if /LOG
: 967 2007 2 modulerfa [rfa$l_vbn] = 0; !Reset module VBN address
: 968 2008 2 globlist [0] = globlist [0];
: 969 2009 2 globlist [1] = globlist [0];
: 970 2010 2 moduleflags = 0;
: 971 2011 2 modnamlng = 0;
: 972 2012 2 RETURN true
: 973 2013 1 END; ! END OF EOM PROCESSING
```



|           |    |           |    |       |       |        |                            |   |      |
|-----------|----|-----------|----|-------|-------|--------|----------------------------|---|------|
|           |    | 00000000G | 8F | DD    | 000C4 | PUSHL  | #LIB\$ MHDERR              | : |      |
|           | 63 |           | 06 | FB    | 000CA | CALLS  | #6, LIB\$SIGNAL            | : |      |
|           |    |           | 01 | DD    | 000CD | PUSHL  | #1                         | : | 1999 |
| 0000V     | CF |           | 01 | FB    | 000CF | CALLS  | #1, FINISH OBJECT          | : |      |
|           | 39 |           | 50 | E9    | 000D4 | BLBC   | STATUS, 10\$               | : |      |
|           |    |           | 5E | DD    | 000D7 | PUSHL  | SP                         | : | 2003 |
| 00000000G | 8F | D4        | A2 | D1    | 000D9 | CMPL   | OPERATION, #LIB\$_REPLACED | : | 2004 |
|           |    |           | 04 | 12    | 000E1 | BNEQ   | 8\$                        | : |      |
|           |    |           | 03 | DD    | 000E3 | PUSHL  | #3                         | : |      |
|           |    |           | 02 | 11    | 000E5 | BRB    | 9\$                        | : |      |
|           |    |           | 02 | DD    | 000E7 | PUSHL  | #2                         | : |      |
| 0000G     | CF |           | 02 | FB    | 000E9 | CALLS  | #2, LIB LOG UPD            | : |      |
|           |    |           | 65 | DD    | 000EE | PUSHL  | LIB\$GL_CIBFDB             | : | 2006 |
|           |    | 04        | AE | 9F    | 000F0 | PUSHAB | MODNAMDESC                 | : |      |
|           |    | D4        | A2 | DD    | 000F3 | PUSHL  | OPERATION                  | : |      |
| 0000G     | CF |           | 03 | FB    | 000F6 | CALLS  | #3, LIB LOG_OP             | : |      |
|           |    | 20        | A2 | D4    | 000FB | CLRL   | MODULERFA                  | : | 2007 |
| 60        | A2 | 60        | A2 | 9E    | 000FE | MOVAB  | GLOBLIST, GLOBLIST         | : | 2008 |
| 64        | A2 | 60        | A2 | 9E    | 00103 | MOVAB  | GLOBLIST, GLOBLIST+4       | : | 2009 |
|           |    | 3C        | A2 | 94    | 00108 | CLRB   | MODULEFLAGS                | : | 2010 |
|           |    |           | 62 | 94    | 0010B | CLRB   | MODNAMLNG                  | : | 2011 |
|           | 50 |           | 01 | D0    | 0010D | MOVL   | #1, R0                     | : | 2012 |
|           |    |           | 04 | 00110 | 10\$: | RET    |                            | : | 2013 |

; Routine Size: 273 bytes, Routine Base: \$CODE\$ + 07C9

finish\_object

```

: 975      2014 1 %SBTTL 'finish_object';
: 976      2015 1
: 977      2016 1 ROUTINE finish_object (allswell) =
: 978      2017 2 BEGIN
: 979      2018 2
: 980      2019 2     This routine is called when the processing for a module is complete.
: 981      2020 2     if allswell is true, the symbols in the queue and the module name
: 982      2021 2     are entered in the index, and the old data and any symbols not replaced
: 983      2022 2     (if replacing) are deleted from the index. If allswell is false,
: 984      2023 2     the list is merely deallocated.
: 985      2024 2
: 986      2025 2 LOCAL
: 987      2026 2     keydesc : BBLOCK [dsc$_s_bln],
: 988      2027 2     keynb  : REF BBLOCK;
: 989      2028 2
: 990      2029 2
: 991      2030 2
: 992      2031 2     Write the end of the data if there was an error and then delete it
: 993      2032 2
: 994      2033 2 IF .modulerfa [rfa$_vbn] NEQ 0                                !If data was written
: 995      2034 2     AND NOT .allswell                                     ! and there was an error
: 996      2035 2 THEN BEGIN
: 997      2036 2     lbr$put_end (lib$gl_libctl);
: 998      2037 2     lbr$delete_data (lib$gl_libctl, modulerfa);          !Delete the new data
: 999      2038 2     modulerfa [rfa$_vbn] = 0;
: 1000     2039 2     END;
: 1001     2040 2
: 1002     2041 2     Set index to the global symbol index
: 1003     2042 2
: 1004     P 2043 2 perform (lbr$set_index (lib$gl_libctl, lib$gl_objgsdix),
: 1005     2044 2     lib$_indexerr, 1, lib$gl_libfdb [fdb$_namdesc]);
: 1006     2045 2
: 1007     2046 2     Enter the new symbols
: 1008     2047 2
: 1009     2048 2 WHILE NOT REMQUE (.globlist, keynb)                       !Insert/replace symbols for module
: 1010     2049 2 DO BEGIN
: 1011     2050 2     IF .allswell
: 1012     2051 2     THEN BEGIN
: 1013     2052 2     keydesc [dsc$_length] = .keynb [lnb$_namlng];
: 1014     2053 2     keydesc [dsc$_pointer] = keynb [lnb$_name];
: 1015     P 2054 2     rms_perform (lbr$replace_key (lib$gl_libctl, keydesc,
: 1016     P 2055 2     oldmodrfa, modulerfa),
: 1017     P 2056 2     lib$_inserterr, .(lbr$gl_rmsstv,
: 1018     2057 2     2, keydesc, lib$gl_libfdb [fdb$_namdesc]);
: 1019     2058 2     END;
: 1020     2059 2     lib_free_mem (lnb$_fixedsize + .keynb [lnb$_namlng], .keynb);
: 1021     2060 2     END;
: 1022     2061 2
: 1023     2062 2     Delete any symbols not replaced
: 1024     2063 2
: 1025     2064 2 WHILE NOT REMQUE (.delist, keynb)
: 1026     2065 2 DO BEGIN
: 1027     2066 2     IF .allswell
: 1028     2067 2     THEN BEGIN
: 1029     2068 2     keydesc [dsc$_length] = .keynb [lnb$_namlng];
: 1030     2069 2     keydesc [dsc$_pointer] = keynb [lnb$_name];
: 1031     P 2070 2     perform (lbr$delete_key (lib$gl_libctl, keydesc),
```

```

: 1032      2071      4      lib$delkeyerr, 2, keydesc, lib$gl_libfdb [fdb$l_namdesc]);
: 1033      2072      3      END;
: 1034      2073      3      lib_free_mem (lmb$fixedsize + .keynb [lmb$b_namlng], .keynb);
: 1035      2074      2      END;
: 1036      2075      2      IF .allswell
: 1037      2076      3      THEN BEGIN
: 1038      P 2077      3      perform (lbr$set_index (lib$gl_libctl, lib$gl_objmodix),
: 1039      P 2078      3      lib$indexerr, 1, lib$gl_libfdb [fdb$l_namdesc]);
: 1040      P 2079      3      rms_perform (lbr$replace_key (lib$gl_libctl, moduledesc,
: 1041      P 2080      3      oldmodrfa, modulerfa),
: 1042      P 2081      3      lib$inserterr, lib$gl_rmsstv,
: 1043      2082      3      2, moduledesc, lib$gl_libfdb [fdb$l_namdesc]);
: 1044      2083      3      !
: 1045      2084      3      ! If replacing, delete the old data
: 1046      2085      3      !
: 1047      2086      3      IF .replacing
: 1048      P 2087      3      THEN rms_perform (lbr$delete_data (lib$gl_libctl, oldmodrfa),
: 1049      2088      3      lib$deldaterr, lib$gl_rmsstv, 1, lib$gl_libfdb [fdb$l_namdesc]);
: 1050      2089      2      END;
: 1051      2090      2      RETURN true
: 1052      2091      2      END;
: 1053      2092      1      !of deallocate_list

```

OFFC 0000 FINISH\_OBJECT:

|           |           |    |    |       |        |                                      |      |
|-----------|-----------|----|----|-------|--------|--------------------------------------|------|
| 5B        | 00000000G | 00 | 9E | 00002 | .WORD  | Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 | 2016 |
| 5A        | 00000000G | 8F | D0 | 00009 | MOVAB  | LBR\$REPLACE KEY, R11                |      |
| 59        | 00000000G | 00 | 9E | 00010 | MOVL   | #LIB\$ INDEXERR, R10                 |      |
| 58        | 00000000G | 00 | 9E | 00017 | MOVAB  | LBR\$SET INDEX, R9                   |      |
| 57        | 00000000G | 00 | 9E | 0001E | MOVAB  | LBR\$DELETE DATA, R8                 |      |
| 56        | 0000G     | CF | 9E | 00025 | MOVAB  | LBR\$GL_RMSSTV, R7                   |      |
| 55        | 0000G     | CF | 9E | 0002A | MOVAB  | LIB\$GL_LIBFDB, R6                   |      |
| 54        | 00000000G | 00 | 9E | 0002F | MOVAB  | LIB\$GL_LIBCTL, R5                   |      |
| 53        | 0000'     | CF | 9E | 00036 | MOVAB  | LIB\$SIGNAL, R4                      |      |
| 5E        |           | 08 | C2 | 0003B | MOVAB  | MODULERFA, R3                        |      |
|           |           | 63 | D5 | 0003E | SUBL2  | #8, SP                               |      |
|           |           | 16 | 13 | 00040 | TSTL   | MODULERFA                            | 2033 |
| 12        | 04        | AC | E8 | 00042 | BEQL   | 1\$                                  | 2034 |
|           |           | 55 | DD | 00046 | BLBS   | ALLSWELL, 1\$                        | 2036 |
| 00000000G | 00        | 01 | FB | 00048 | PUSHL  | R5                                   |      |
|           |           | 53 | DD | 0004F | CALLS  | #1, LBR\$PUT_END                     | 2037 |
|           |           | 55 | DD | 00051 | PUSHL  | R3                                   |      |
| 68        |           | 02 | FB | 00053 | PUSHL  | R5                                   |      |
|           |           | 63 | D4 | 00056 | CALLS  | #2, LBR\$DELETE_DATA                 | 2038 |
|           | 0000G     | CF | 9F | 00058 | CLRL   | MODULERFA                            | 2044 |
|           |           | 55 | DD | 0005C | PUSHAB | LIB\$GL_OBJGSDIX                     |      |
| 69        |           | 02 | FB | 0005E | PUSHL  | R5                                   |      |
| 0D        |           | 50 | E8 | 00061 | CALLS  | #2, LBR\$SET_INDEX                   |      |
|           |           | 50 | DD | 00064 | BLBS   | STATUS, 2\$                          |      |
| 7E        |           | 10 | C1 | 00066 | PUSHL  | STATUS                               |      |
|           |           | 01 | DD | 0006A | ADDL3  | #16, LIB\$GL_LIBFDB, -(SP)           |      |
|           |           | 5A | DD | 0006C | PUSHL  | #1                                   |      |
|           |           | 04 | FB | 0006E | PUSHL  | R10                                  |      |
|           |           |    |    |       | CALLS  | #4, LIB\$SIGNAL                      |      |

|           |    |           |    |       |       |      |        |                            |      |
|-----------|----|-----------|----|-------|-------|------|--------|----------------------------|------|
|           | 52 | 40        | B3 | 0F    | 00071 | 2\$: | REMQUE | @GLOBLIST, KEYNB           | 2048 |
|           |    |           | 43 | 1D    | 00075 |      | BVS    | 4\$                        | 2050 |
|           | 2F | 04        | AC | E9    | 00077 |      | BLBC   | ALLSWELL, 3\$              | 2052 |
|           | 6E | 09        | A2 | 9B    | 0007B |      | MOVZBW | 9(KEYNB), KEYDESC          | 2053 |
| 04        | AE | 0A        | A2 | 9E    | 0007F |      | MOVAB  | 10(R2), KEYDESC+4          | 2057 |
|           |    |           | 53 | DD    | 00084 |      | PUSHL  | R3                         |      |
|           |    | 08        | A3 | 9F    | 00086 |      | PUSHAB | OLDMODRFA                  |      |
|           |    | 08        | AE | 9F    | 00089 |      | PUSHAB | KEYDESC                    |      |
|           |    |           | 55 | DD    | 0008C |      | PUSHL  | R5                         |      |
|           | 6B | 04        | FB | 0008E |       |      | CALLS  | #4, LBR\$REPLACE_KEY       |      |
|           | 16 | 50        | E8 | 00091 |       |      | BLBS   | STATUS, 3\$                |      |
|           |    |           | 67 | DD    | 00094 |      | PUSHL  | LBR\$GL_RMSSTV             |      |
|           |    |           | 50 | DD    | 00096 |      | PUSHL  | STATUS                     |      |
| 7E        | 66 | 10        | C1 | 00098 |       |      | ADDL3  | #16, LIB\$GL_LIBFDB, -(SP) |      |
|           |    | 0C        | AE | 9F    | 0009C |      | PUSHAB | KEYDESC                    |      |
|           |    |           | 02 | DD    | 0009F |      | PUSHL  | #2                         |      |
|           | 64 | 0000000G  | 8F | DD    | 000A1 |      | PUSHL  | #LIB\$ INSERTERR           |      |
|           |    |           | 06 | FB    | 000A7 |      | CALLS  | #6, LIB\$SIGNAL            |      |
|           |    |           | 52 | DD    | 000AA | 3\$: | PUSHL  | KEYNB                      | 2059 |
|           | 7E | 09        | A2 | 9A    | 000AC |      | MOVZBL | 9(KEYNB), -(SP)            |      |
|           | 6E | 0A        | C0 | 000B0 |       |      | ADDL2  | #10, (SP)                  |      |
| 0000G     | CF | 02        | FB | 000B3 |       |      | CALLS  | #2, LIB_FREE_MEM           |      |
|           |    |           | B7 | 11    | 000B8 |      | BRB    | 2\$                        | 2048 |
|           | 52 | 48        | B3 | 0F    | 000BA | 4\$: | REMQUE | @DELIST, KEYNB             | 2064 |
|           |    |           | 3F | 1D    | 000BE |      | BVS    | 6\$                        |      |
|           | 2B | 04        | AC | E9    | 000C0 |      | BLBC   | ALLSWELL, 5\$              | 2066 |
|           | 6E | 09        | A2 | 9B    | 000C4 |      | MOVZBW | 9(KEYNB), KEYDESC          | 2068 |
| 04        | AE | 0A        | A2 | 9E    | 000C8 |      | MOVAB  | 10(R2), KEYDESC+4          | 2069 |
|           |    | 4020      | 8F | BB    | 000CD |      | PUSHR  | #^M<R5, SP>                | 2071 |
| 00000000G | 00 |           | 02 | FB    | 000D1 |      | CALLS  | #2, LBR\$DELETE_KEY        |      |
|           | 14 |           | 50 | E8    | 000D8 |      | BLBS   | STATUS, 5\$                |      |
|           |    |           | 50 | DD    | 000DB |      | PUSHL  | STATUS                     |      |
| 7E        | 66 | 10        | C1 | 000DD |       |      | ADDL3  | #16, LIB\$GL_LIBFDB, -(SP) |      |
|           |    | 08        | AE | 9F    | 000E1 |      | PUSHAB | KEYDESC                    |      |
|           |    |           | 02 | DD    | 000E4 |      | PUSHL  | #2                         |      |
|           | 64 | 00000000G | 8F | DD    | 000E6 |      | PUSHL  | #LIB\$ DELKEYERR           |      |
|           |    |           | 05 | FB    | 000EC |      | CALLS  | #5, LIB\$SIGNAL            |      |
|           |    |           | 52 | DD    | 000EF | 5\$: | PUSHL  | KEYNB                      | 2073 |
|           | 7E | 09        | A2 | 9A    | 000F1 |      | MOVZBL | 9(KEYNB), -(SP)            |      |
|           | 6E | 0A        | C0 | 000F5 |       |      | ADDL2  | #10, (SP)                  |      |
| 0000G     | CF | 02        | FB | 000F8 |       |      | CALLS  | #2, LIB_FREE_MEM           |      |
|           |    |           | BB | 11    | 000FD |      | BRB    | 4\$                        | 2064 |
|           | 61 | 04        | AC | E9    | 000FF | 6\$: | BLBC   | ALLSWELL, 9\$              | 2075 |
|           |    | 0000G     | CF | 9F    | 00103 |      | PUSHAB | LIB\$GL_OBJMODIX           | 2078 |
|           |    |           | 55 | DD    | 00107 |      | PUSHL  | R5                         |      |
|           | 69 | 02        | FB | 00109 |       |      | CALLS  | #2, LBR\$SET_INDEX         |      |
|           | 0D | 50        | E8 | 0010C |       |      | BLBS   | STATUS, 7\$                |      |
|           |    |           | 50 | DD    | 0010F |      | PUSHL  | STATUS                     |      |
| 7E        | 66 | 10        | C1 | 00111 |       |      | ADDL3  | #16, LIB\$GL_LIBFDB, -(SP) |      |
|           |    |           | 01 | DD    | 00115 |      | PUSHL  | #1                         |      |
|           |    |           | 5A | DD    | 00117 |      | PUSHL  | R10                        |      |
|           | 64 | 04        | FB | 00119 |       |      | CALLS  | #4, LIB\$SIGNAL            |      |
|           |    |           | 53 | DD    | 0011C | 7\$: | PUSHL  | R3                         | 2082 |
|           |    | 08        | A3 | 9F    | 0011E |      | PUSHAB | OLDMODRFA                  |      |
|           |    | 14        | A3 | 9F    | 00121 |      | PUSHAB | MODULEDESC                 |      |
|           |    |           | 55 | DD    | 00124 |      | PUSHL  | R5                         |      |
|           | 6B | 04        | FB | 00126 |       |      | CALLS  | #4, LBR\$REPLACE_KEY       |      |



|    |           |    |       |          |           |                            |      |
|----|-----------|----|-------|----------|-----------|----------------------------|------|
| 16 |           | 50 | E8    | 00129    | BLBS      | STATUS, 8\$                | :    |
|    |           | 67 | DD    | 0012C    | PUSHL     | LBR\$GL_RMSSTV             | :    |
|    |           | 50 | DD    | 0012E    | PUSHL     | STATUS                     | :    |
| 7E |           | 66 | 10    | C1 00130 | ADDL3     | #16, LIB\$GL_LIBFDB, -(SP) | :    |
|    | 14        | A3 | 9F    | 00134    | PUSHAB    | MODULEDESC                 | :    |
|    |           | 02 | DD    | 00137    | PUSHL     | #2                         | :    |
|    | 00000000G | 8F | DD    | 00139    | PUSHL     | #LIB\$ INSERTERR           | :    |
| 64 |           | 06 | FB    | 0013F    | CALLS     | #6, LIB\$SIGNAL            | :    |
| 1E | 10        | A3 | E9    | 00142    | 8\$: BLBC | REPLACING, 9\$             | 2086 |
|    | 08        | A3 | 9F    | 00146    | PUSHAB    | OLDMODRFA                  | 2088 |
|    |           | 55 | DD    | 00149    | PUSHL     | R5                         | :    |
| 68 |           | 02 | FB    | 0014B    | CALLS     | #2, LBR\$DELETE_DATA       | :    |
| 13 |           | 50 | E8    | 0014E    | BLBS      | STATUS, 9\$                | :    |
|    |           | 67 | DD    | 00151    | PUSHL     | LBR\$GL_RMSSTV             | :    |
|    |           | 50 | DD    | 00153    | PUSHL     | STATUS                     | :    |
| 7E |           | 66 | 10    | C1 00155 | ADDL3     | #16, LIB\$GL_LIBFDB, -(SP) | :    |
|    |           | 01 | DD    | 00159    | PUSHL     | #1                         | :    |
|    | 00000000G | 8F | DD    | 0015B    | PUSHL     | #LIB\$ DELDATERR           | :    |
| 64 |           | 05 | FB    | 00161    | CALLS     | #5, LIB\$SIGNAL            | :    |
| 50 |           | 01 | D0    | 00164    | 9\$: MOVL | #1, R0                     | 2091 |
|    |           | 04 | 00167 | RET      |           |                            | 2092 |

; Routine Size: 360 bytes, Routine Base: \$CODE\$ + 08DA

```
1055 2093 1 %SBTTL 'seqchk';
1056 2094 1
1057 2095 1 ROUTINE seqchk =
1058 2096 1
1059 2097 1 Routine which validates that records are in correct sequence.
1060 2098 1 Returns value false if not, true otherwise.
1061 2099 1
1062 2100 2 BEGIN
1063 2101 2 BIND
1064 2102 2     hdrsubtyp = objrec [obj$b_subtyp] : BYTE;
1065 2103 2
1066 2104 2 IF .correctyp EQL obj$c_hdr
1067 2105 2 THEN
1068 2106 2     IF .hdrsubtyp EQL obj$c_hdr_mhd
1069 2107 2     THEN
1070 2108 2         IF (.lastrectyp EQL obj$c_eom) OR
1071 2109 2         (.lastrectyp EQL obj$c_eomw)
1072 2110 2         THEN (mhdseen = true;
1073 2111 2             lnmseen = false;
1074 2112 2             RETURN true)
1075 2113 2         ELSE BEGIN
1076 2114 2             SIGNAL (lib$seqnce, 2, modnamlng,
1077 2115 2                 lib$gl_inpfdb [fdb$l_namdesc]);
1078 2116 2             RETURN lib$seqnce;
1079 2117 2         END
1080 2118 2     ELSE
1081 2119 2     IF .mhdseen
1082 2120 2     THEN (IF .hdrsubtyp EQL obj$c_hdr_lnm
1083 2121 2         THEN lnmseen = true;
1084 2122 2         RETURN true)
1085 2123 2     ELSE BEGIN
1086 2124 2         SIGNAL (lib$seqnce, 2, modnamlng,
1087 2125 2             lib$gl_inpfdb [fdb$l_namdesc]);
1088 2126 2         RETURN lib$seqnce;
1089 2127 2     END
1090 2128 2 ELSE
1091 2129 2 IF .mhdseen
1092 2130 2 AND .lnmseen
1093 2131 2 THEN
1094 2132 2     BEGIN
1095 2133 2     IF (.correctyp EQL obj$c_eom) OR
1096 2134 2     (.correctyp EQL obj$c_eomw)
1097 2135 2     THEN mhdseen = false;
1098 2136 2     RETURN true;
1099 2137 2     END
1100 2138 2 ELSE BEGIN
1101 2139 2     SIGNAL (lib$seqnce, 2, modnamlng,
1102 2140 2         lib$gl_inpfdb [fdb$l_namdesc]);
1103 2141 2     RETURN lib$seqnce;
1104 2142 2     END;
1105 2143 1 END;
```

|    |           |    |           |    |       |       |      |        |                            |  |      |
|----|-----------|----|-----------|----|-------|-------|------|--------|----------------------------|--|------|
|    |           | 53 | 00000000G | 8F | D0    | 00002 |      | MOVL   | #LIB\$ SEQNCE, R3          |  |      |
|    |           | 52 | 0000'     | CF | 9E    | 00009 |      | MOVAB  | MHDSEEN, R2                |  |      |
| 51 | 18        | A2 |           | 01 | C1    | 0000E |      | ADDL3  | #1, OBJREC, R1             |  | 2102 |
|    |           | 50 | 20        | A2 | D0    | 00013 |      | MOVL   | CURRECTYP, R0              |  | 2104 |
|    |           |    |           | 23 | 12    | 00017 |      | BNEQ   | 3\$                        |  |      |
|    |           |    |           | 61 | 95    | 00019 |      | TSTB   | (R1)                       |  | 2106 |
|    |           |    |           | 11 | 12    | 0001B |      | BNEQ   | 2\$                        |  |      |
|    |           | 03 | 1C        | A2 | D1    | 0001D |      | CMPL   | LASTRECTYP, #3             |  | 2108 |
|    |           |    |           | 06 | 13    | 00021 |      | BEQL   | 1\$                        |  |      |
|    |           | 07 | 1C        | A2 | D1    | 00023 |      | CMPL   | LASTRECTYP, #7             |  | 2109 |
|    |           |    |           | 2A | 12    | 00027 |      | BNEQ   | 6\$                        |  |      |
|    |           | 62 |           | 01 | 7D    | 00029 | 1\$: | MOVQ   | #1, MHDSEEN                |  | 2110 |
|    |           |    |           | 21 | 11    | 0002C |      | BRB    | 5\$                        |  | 2112 |
|    |           | 22 |           | 62 | E9    | 0002E | 2\$: | BLBC   | MHDSEEN, 6\$               |  | 2119 |
|    |           | 01 |           | 61 | 91    | 00031 |      | CMPB   | (R1), #1                   |  | 2120 |
|    |           |    |           | 19 | 12    | 00034 |      | BNEQ   | 5\$                        |  |      |
|    | 04        | A2 |           | 01 | D0    | 00036 |      | MOVL   | #1, LNMSEEN                |  | 2121 |
|    |           |    |           | 13 | 11    | 0003A |      | BRB    | 5\$                        |  | 2122 |
|    |           | 14 |           | 62 | E9    | 0003C | 3\$: | BLBC   | MHDSEEN, 6\$               |  | 2129 |
|    |           | 10 | 04        | A2 | F9    | 0003F |      | BLBC   | LNSEEN, 6\$                |  | 2130 |
|    |           | 03 |           | 50 | D1    | 00043 |      | CMPL   | R0, #3                     |  | 2133 |
|    |           |    |           | 05 | 13    | 00046 |      | BEQL   | 4\$                        |  |      |
|    |           | 07 |           | 50 | D1    | 00048 |      | CMPL   | R0, #7                     |  | 2134 |
|    |           |    |           | 02 | 12    | 0004B |      | BNEQ   | 5\$                        |  |      |
|    |           |    |           | 62 | D4    | 0004D | 4\$: | CLRL   | MHDSEEN                    |  | 2135 |
|    |           | 50 |           | 01 | D0    | 0004F | 5\$: | MOVL   | #1, R0                     |  | 2136 |
|    |           |    |           |    | 04    | 00052 |      | RET    |                            |  |      |
| 7E | 0000G     | CF |           | 10 | C1    | 00053 | 6\$: | ADDL3  | #16, LIB\$GL_INPFDB, -(SP) |  | 2140 |
|    |           |    | 28        | A2 | 9F    | 00059 |      | PUSHAB | MODNAMLANG                 |  | 2139 |
|    |           |    |           | 02 | DD    | 0005C |      | PUSHL  | #2                         |  | 2140 |
|    |           |    |           | 53 | DD    | 0005E |      | PUSHL  | R3                         |  |      |
|    | 00000000G | 00 |           | 04 | FB    | 00060 |      | CALLS  | #4, LIB\$SIGNAL            |  |      |
|    |           | 50 |           | 53 | D0    | 00067 |      | MOVL   | R3, R0                     |  | 2141 |
|    |           |    |           | 04 | 0006A |       |      | RET    |                            |  | 2143 |

; Routine Size: 107 bytes, Routine Base: \$CODE\$ + 0A42

```

: 1107      2144 1 %SBTTL 'prorec';
: 1108      2145 1
: 1109      2146 1 ROUTINE prorec =
: 1110      2147 2 BEGIN
: 1111      2148 2
: 1112      2149 2 | This routine checks for proper record sequence and then
: 1113      2150 2 | copies the record to the object library.
: 1114      2151 2
: 1115      2152 2 perform (seqchk ());           .Check sequence
: 1116      2153 2 IF NOT .lib$gl_ctlmsk [lib$v_shrstb]
: 1117      2154 2     THEN RETURN copyrec ()      'Copy to library
: 1118      2155 2     ELSE RETURN true
: 1119      2156 1 END;

```

.Of prorec

|    |       |    |            |         |       |              |                         |        |
|----|-------|----|------------|---------|-------|--------------|-------------------------|--------|
|    |       |    | 0000 00000 | PROREC: | .WORD | Save nothing | : 2146                  |        |
|    | 8F    | AF | 00         | FB      | 00002 | CALLS        | #0, SEQCHK              | : 2152 |
|    |       | OF | 50         | E9      | 00006 | BLBC         | STATUS, 2\$             | :      |
| 06 | 0000G | CF | 05         | E0      | 00009 | BBS          | #5, LIB\$GL_CTLMSK, 1\$ | : 2153 |
|    | 0000V | CF | 00         | FB      | 0000F | CALLS        | #0, COPYREC             | : 2154 |
|    |       |    |            | 04      | 00014 | RET          |                         | : 2155 |
|    |       | 50 | 01         | D0      | 00015 | 1\$:         | MOVL                    | #1, R0 |
|    |       |    | 04         | 00018   | 2\$:  | RET          |                         | : 2156 |

: Routine Size: 25 bytes, Routine Base: \$CODE\$ + 0AAD

```

: 1120      2157 1 ROUTINE copyrec =
: 1121      2158 2 BEGIN
: 1122      2159 2
: 1123      2160 2 | This routine copies the record to the object library
: 1124      2161 2
: 1125      2162 2 LOCAL
: 1126      2163 2     txtrfa : BBLOCK [rfa$c_length],
: 1127      2164 2     bufdesc : BBLOCK [dsc$c_s_bln];
: 1128      2165 2
: 1129      2166 2 bufdesc [dsc$w_length] = .reclng;
: 1130      2167 2 bufdesc [dsc$a_pointer] = .objrec;
: 1131      P 2168 2 rms_perform (lbr$put_record (lib$gl_libctl, bufdesc, txtrfa),
: 1132      2169 2     lib$writeerr, .lbr$gl_rmsstv, 1, lib$gl_libfdb [fdb$_namdesc]);
: 1133      2170 2 IF .modulerfa [rfa$l_vbn] EQL 0
: 1134      2171 2 THEN BEGIN
: 1135      2172 2     modulerfa [rfa$l_vbn] = .txtrfa [rfa$l_vbn];
: 1136      2173 2     modulerfa [rfa$w_offset] = .txtrfa [rfa$w_offset];
: 1137      2174 2 END;
: 1138      2175 2 RETURN true
: 1139      2176 1 END;

```

!Of copyrec

|    |       |    |            |          |       |               |        |
|----|-------|----|------------|----------|-------|---------------|--------|
|    |       |    | 0004 00000 | COPYREC: | .WORD | Save R2       | : 2157 |
| 52 | 0000' | CF | 9E         | 00002    | MOVAB | MODULERFA, R2 | :      |

|           |       |           |    |       |       |        |                            |   |      |
|-----------|-------|-----------|----|-------|-------|--------|----------------------------|---|------|
|           | 5E    |           | 10 | C2    | 00007 | SUBL2  | #16, SP                    |   |      |
|           | 6E    | CC        | A2 | B0    | 0000A | MOVW   | RECLNG, BUFDESC            | : | 2166 |
| 04        | AE    | D0        | A2 | D0    | 0000E | MOVL   | OBJREC, BUFDESC+4          | : | 2167 |
|           |       | 08        | AE | 9F    | 00013 | PUSHAB | TXTRFA                     | : | 2169 |
|           |       | 04        | AE | 9F    | 00016 | PUSHAB | BUFDESC                    | : |      |
|           |       | 0000G     | CF | 9F    | 00019 | PUSHAB | LIB\$GL_LIBCTL             | : |      |
| 00000000G | 00    |           | 03 | FB    | 0001D | CALLS  | #3, LBR\$PUT_RECORD        | : |      |
|           | 1D    |           | 50 | E8    | 00024 | BLBS   | STATUS, 1\$                | : |      |
|           |       | 00000000G | 00 | DD    | 00027 | PUSHL  | LBR\$GL_RMSSTV             | : |      |
|           |       |           | 50 | DD    | 0002D | PUSHL  | STATUS                     | : |      |
| 7E        | 0000G | CF        | 10 | C1    | 0002F | ADDL3  | #16, LIB\$GL_LIBFDB, -(SP) | : |      |
|           |       |           | 01 | DD    | 00035 | PUSHL  | #1                         | : |      |
|           |       | 008610D2  | 8F | DD    | 00037 | PUSHL  | #8786130                   | : |      |
| 00000000G | 00    |           | 05 | FB    | 0003D | CALLS  | #5, LIB\$SIGNAL            | : |      |
|           |       |           | 62 | D5    | 00044 | TSTL   | MODULERFA                  | : | 2170 |
|           |       |           | 09 | 12    | 00046 | BNEQ   | 2\$                        | : |      |
|           | 62    | 08        | AE | D0    | 00048 | MOVL   | TXTRFA, MODULERFA          | : | 2172 |
| 04        | A2    | 0C        | AE | B0    | 0004C | MOVW   | TXTRFA+4, MODULERFA+4      | : | 2173 |
|           | 50    |           | 01 | D0    | 00051 | MOVL   | #1, R0                     | : | 2175 |
|           |       |           | 04 | 00054 | 2\$:  | RET    |                            | : | 2176 |

: Routine Size: 85 bytes, Routine Base: \$CODE\$ + 0AC6

: 1140 2177 1  
: 1141 2178 1 END  
: 1142 2179 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

| Name     | Bytes | Attributes   |
|----------|-------|--|
| \$OWNS   | 200   | NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)   |
| \$PLITS  | 152   | NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |
| \$CODE\$ | 2843  | NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)   |

Library Statistics

| File                            | ----- Symbols ----- |        |         | Pages Mapped | Processing Time |
|---------------------------------|---------------------|--------|---------|--------------|-----------------|
|                                 | Total               | Loaded | Percent |              |                 |
| _\$255\$DUA28:[SYSLIB]LIB.L32;1 | 18619               | 120    | 0       | 1000         | 00:01.9         |

LIB INPUTOBJ  
V04=000

prorec

E 15  
16-Sep-1984 01:57:57  
14-Sep-1984 12:38:04

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

Page 52  
(24)

L1  
V0

COMMAND QUALIFIERS

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

: Size: 2843 code + 352 data bytes  
: Run Time: 00:56.3  
: Elapsed Time: 02:02.7  
: Lines/CPU Min: 2321  
: Lexemes/CPU-Min: 28165  
: Memory Used: 275 pages  
: Compilation Complete

