

[illegible]

FOR  
2-0

.....

```

FFF FFFF FFFF 000000 RRRRRRRR FFFF FFFF FFFF I I I I I NN NN DDDDDDDD
FFF FFFF FFFF 000000 RRRRRRRR FFFF FFFF FFFF I I I I I NN NN DDDDDDDD
FF 00 00 RR RR FF I I NN NN DD DD
FF 00 00 RR RR FF I I NN NN DD DD
FF 00 00 RR RR FF I I NNNN NN DD DD
FF 00 00 RR RR FF I I NNNN NN DD DD
FFFF FFFF 00 00 RRRRRRRR FFFF FFFF I I NN NN DD DD
FFFF FFFF 00 00 RRRRRRRR FFFF FFFF I I NN NN DD DD
FF 00 00 RR RR FF I I NN NN DD DD
FF 00 00 RR RR FF I I NN NN DD DD
FF 00 00 RR RR FF I I NN NN DD DD
FF 00 00 RR RR FF I I NN NN DD DD
FF 00 00 RR RR FF I I NN NN DD DD
FF 000000 RR RR FF I I I I I NN NN DDDDDDDD
FF 000000 RR RR FF I I I I I NN NN DDDDDDDD

```

```

LL I I I I I SSSSSSSS
LL I I I I I SSSSSSSS
LL I I SS
LL I I SS
LL I I SS
LL I I SS
LL I I SSSSSS
LL I I SSSSSS
LL I I SS
LL I I SS
LL I I SS
LL I I SS
LLLLLLLLLLLL I I I I I SSSSSSSS
LLLLLLLLLLLL I I I I I SSSSSSSS

```



```

1 0001 0 MODULE FOR$FIND (          ! Implements FORTRAN FIND statement
2 0002 0          IDENT = '2-016'    ! File: FORFIND.B32 Edit: JBS2016
3 0003 0          ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 *  ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 *  TRANSFERRED.
18 0018 1 *
19 0019 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 *  CORPORATION.
22 0022 1 *
23 0023 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: FORTRAN Support Library - User Callable
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1     This module implements the FORTRAN FIND statement.
36 0036 1     This is performed by doing everything a READ Direct
37 0037 1     I/O statement would do, except actually doing the I/O.
38 0038 1
39 0039 1 ENVIRONMENT: User access mode; mixture of AST level or not.
40 0040 1
41 0041 1 AUTHOR: Jonathan M. Taylor, CREATION DATE: 1-OCT-77 : Version 0
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1     JMT, 24-OCT-77: VERSION 2
46 0046 1 2-0 - original
47 0047 1 2-1 - Use FOR$K abcmnoxyz as EXTERNAL LITERALS. TNH 27-Oct-77
48 0048 1 2-2 - Change MIXADDMOD to OPEDEFREQ; INVRECNUM to RECNUMOUT. TNH 29-Oct-77
49 0049 1 2-4 - Use FERR. TNH 16-Dec-77
50 0050 1 2-5 - Global register CCB. JMT 8-Apr-78
51 0051 1 2-06 - Change REQUIRE files for VAX system build. DGP 28-Apr-78
52 0052 1 2-07 - Use JSB linkages. TNH 22-May-78
53 0053 1 2-08 - Pass OPEN$K_LUN_MIN to FOR$SCB PUSH. TNH 22-May-78
54 0054 1 2-09 - Change file name to FORFIND.B32, and change the names of
55 0055 1     the REQUIRE files similarly. JBS 14-NOV-78
56 0056 1 2-010 - Update the copyright notice. JBS 16-NOV-78
57 0057 1 2-011 - Change the REQUIRE file names from FOR... to OTS... JBS 07-DEC-78

```



FOR\$FIND  
2-016

D 9  
16-Sep-1984 00:22:43 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:31:58 [FORRTL.SRC]FORFIND.B32;1

Page 2  
(1)

```
: 58      0058 1 2-012 - Change OTSOPN back to FOROPN, and change the prefix on
: 59      0059 1      LUN parameters from OPEN to LUB. JBS 13-DEC-78
: 60      0060 1 2-013 - Add LIBRARY RTLSTARLE so that FOROPN can use its symbols.
: 61      0061 1      JBS 10-APR-1979
: 62      0062 1 2-014 - Actually do a $FIND here. Use IOSTAT handler. SBL 8-May-1979
: 63      0063 1 2-015 - Set LUB$V FIND_LAST. SBL 15-May-1979
: 64      0064 1 2-016 - Put BUILTIN ACTUALCOUNT in the routine, since the next version of
: 65      0065 1      the BLISS compiler will require it to be there. Since we are
: 66      0066 1      editing this module, improve the source text layout. Note that
: 67      0067 1      this edit changes no code. JBS 21-Aug-1980
: 68      0068 1  !--
: 69      0069 1
: 70      0070 1 !<BLF/PAGE>
```

```

72      0071 1 |
73      0072 1 | SWITCHES:
74      0073 1 |
75      0074 1 |
76      0075 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
77      0076 1 |
78      0077 1 |
79      0078 1 | LINKAGES:
80      0079 1 |
81      0080 1 |
82      0081 1 | REQUIRE 'RTLIN:OTSLNK';           ! define all linkages
83      0510 1 |
84      0511 1 |
85      0512 1 | TABLE OF CONTENTS:
86      0513 1 |
87      0514 1 |
88      0515 1 | FORWARD ROUTINE
89      0516 1 |     FOR$FIND;                     ! FORTRAN FIND statement
90      0517 1 |
91      0518 1 |
92      0519 1 | INCLUDE FILES:
93      0520 1 |
94      0521 1 |
95      0522 1 | LIBRARY 'RTLSTARLE';             ! System symbols
96      0523 1 |
97      0524 1 | REQUIRE 'RTLML:FORERR';          ! FORTRAN error number definitions
98      0592 1 |
99      0593 1 | REQUIRE 'RTLML:FORPAR';          ! Inter-module constants
100     0616 1 |
101     0617 1 | REQUIRE 'RTLIN:FOROPN';          ! FORTRAN OPEN/CLOSE parameters
102     0880 1 |
103     0881 1 | REQUIRE 'RTLIN:RTLPSECT';        ! Define DECLARE_PSECTS macro
104     0976 1 |
105     0977 1 | REQUIRE 'RTLML:OTSLUB';          ! Get LUB offsets
106     1117 1 |
107     1118 1 |
108     1119 1 | MACROS:
109     1120 1 |
110     1121 1 |     NONE
111     1122 1 |
112     1123 1 | EQUATED SYMBOLS:
113     1124 1 |
114     1125 1 |     NONE
115     1126 1 |
116     1127 1 | OWN STORAGE:
117     1128 1 |
118     1129 1 |     NONE
119     1130 1 |
120     1131 1 | EXTERNAL REFERENCES:
121     1132 1 |
122     1133 1 |
123     1134 1 | EXTERNAL ROUTINE
124     1135 1 |     FOR$$IOSTAT_HND,             ! Error handler
125     1136 1 |     FOR$$OPEN_DEFLT : CALL CCB NOVALUE, ! FORTRAN default file opener
126     1137 1 |     FOR$$SIGNAL_STO : NOVALUE,    ! convert error code and signal
127     1138 1 |     FOR$$ASSOC : CALL CCB NOVALUE, ! store rec. # in assoc. variable
128     1139 1 |     FOR$$CB_PUSH : JSB_CB_PUSH NOVALUE, ! Get a control block
```



FOR\$FIND  
2-016

F 9  
16-Sep-1984 00:22:43 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:31:58 [FORRTL.SRC]FORFIND.B32;1

Page 4  
(2)

```
: 129      1140 1      FOR$$CB_POP : JSB_CB_POP NOVALUE;          ! Return a control block
: 130      1141 1
: 131      1142 1
: 132      1143 1      PSECT DECLARATIONS:
: 133      1144 1
: 134      1145 1      DECLARE_PSECTS (FOR);          ! Declare PSECTS for FOR$ facility
: 135      1146 1
```

```
137 1147 1 GLOBAL ROUTINE FOR$FIND (UNIT, REC_NO, ERR_EQL) =
138 1148 1
139 1149 1 ++
140 1150 1 FUNCTIONAL DESCRIPTION:
141 1151 1
142 1152 1     Call CB_PUSH to get a control block for this UNIT.
143 1153 1     If has not been declared to be direct access then ERROR.
144 1154 1     If not already open then open ad direct-access, OLD, unformatted.
145 1155 1     Set the record number into the LUB and (optionally) into the
146 1156 1     users associated variable.
147 1157 1     Call RMS to $FIND and lock the record. Return the I/O system to its previous state.
148 1158 1
149 1159 1 FORMAL PARAMETERS:
150 1160 1
151 1161 1     UNIT.rl.v      Unit number (call by value)
152 1162 1     REC_NO.rl.v    Record number to seek (call by value)
153 1163 1     ERR_EQL.rl.v   If 1, return IOSTAT value
154 1164 1               If omitted or zero, signal all errors
155 1165 1
156 1166 1 IMPLICIT INPUTS:
157 1167 1
158 1168 1     LUB$V_DIRECT    1 if DEFINE FILE done or OPEN with access='DIRECT'
159 1169 1     LUB$V_OPENED    1 if file previously opened
160 1170 1
161 1171 1 IMPLICIT OUTPUTS:
162 1172 1
163 1173 1     LUB$L_LOG_RECNO Gets set to the record number passed as
164 1174 1                   a parameter.
165 1175 1     ass. var.       Users associated variable gets new LOG_RECNO
166 1176 1     LUB$V_FIND_LAST Set to 1 after successful find, else 0.
167 1177 1
168 1178 1 ROUTINE VALUE:
169 1179 1     NONE
170 1180 1
171 1181 1 COMPLETION CODES:
172 1182 1
173 1183 1     NONE
174 1184 1
175 1185 1 SIDE EFFECTS:
176 1186 1
177 1187 1     The file is positioned to the requested record and the record is locked.
178 1188 1     Users associated variable (if present) takes the value of
179 1189 1     the new record number. The file is opened if it was never opened before.
180 1190 1     SIGNAL_STOPS FOR$_OPEDEFREQ (26 = 'OPEN OR DEFINE FILE REQUIRED FOR DIRECT ACCESS')
181 1191 1     SIGNAL_STOPS FOR$_RECNUMOUT (25 = 'RECORD NUMBER OUT OF RANGE')
182 1192 1     SIGNAL_STOPS FOR$_ATTACCNON (36 = 'ATTEMPT TO ACCESS NON-EXISTANT RECORD')
183 1193 1     SIGNAL_STOPS FOR$_SPERECLOC (52 = 'SPECIFIED RECORD LOCKED')
184 1194 1     SIGNAL_STOPS FOR$_FINERR    (57 = 'FIND ERROR')
185 1195 1
186 1196 1 --
187 1197 1
188 1198 2 BEGIN
189 1199 2
190 1200 2 GLOBAL REGISTER
191 1201 2     CCB = K_CCB_REG : REF BLOCK [, BYTE];
192 1202 2
193 1203 2 BUILTIN
```



```
194 1204 2      ACTUALCOUNT;
195 1205 2
196 1206 2      LOCAL
197 1207 2          L_UNWIND_ACTION : VOLATILE,
198 1208 2          L_ERR_EQL_PRES : VOLATILE,
199 1209 2          STATUS;
200 1210 2
201 1211 2      ENABLE
202 1212 2          FOR$$IOSTAT_HND (L_UNWIND_ACTION, L_ERR_EQL_PRES);
203 1213 2
204 1214 2      +
205 1215 2      Set up unwind action and flag if ERR= or IOSTAT was present.
206 1216 2      -
207 1217 2          L_UNWIND_ACTION = FOR$K_UNWINDNOP;
208 1218 2
209 1219 2          IF (ACTUALCOUNT () GTR 2) THEN L_ERR_EQL_PRES = .ERR_EQL ELSE L_ERR_EQL_PRES = 0;
210 1220 2
211 1221 2      +
212 1222 2      Get a LUB for this unit.
213 1223 2      On return, CCB points to the current control block.
214 1224 2      -
215 1225 2          FOR$$CB_PUSH (.UNIT, LUB$K_LUN_MIN);
216 1226 2      +
217 1227 2      Error action is now to pop the LUB.
218 1228 2      -
219 1229 2          L_UNWIND_ACTION = FOR$K_UNWINDPOP;
220 1230 2      +
221 1231 2      If file not specified for direct access, then SIGNAL_STOP FOR$_OPEDEFREQ
222 1232 2      (26 = 'DIRECT ACCESS NOT SPECIFIED WHEN UNIT OPENED')
223 1233 2      -
224 1234 2
225 1235 2          IF (.CCB [LUB$V_DIRECT] EQL 0) THEN FOR$$SIGNAL_STO (FOR$K_OPEDEFREQ);
226 1236 2
227 1237 2      +
228 1238 2      If file has not been previously OPENed, default-open it.
229 1239 2      -
230 1240 2
231 1241 2          IF (.CCB [LUB$V_OPENED] EQL 0) THEN FOR$$OPEN_DEFLT (OPEN$K_ACC_DIR, OPEN$K_TYP_OLD, OPEN$K_FOR_UNF);
232 1242 2
233 1243 2      +
234 1244 2      If the logical record number passed is illegal then SIGNAL_STOP,
235 1245 2      else store in the LUB and users associated variable.
236 1246 2      -
237 1247 2
238 1248 2          IF ((.REC_NO EQLU 0) OR ((.CCB [LUB$L_REC_MAX] NEQ 0) AND (.REC_NO GTRU .CCB [LUB$L_REC_MAX])))
239 1249 2          THEN
240 1250 2              FOR$$SIGNAL_STO (FOR$K_RECNUMOUT);
241 1251 2
242 1252 2          CCB [LUB$L_LOG_RECNO] = .REC_NO;
243 1253 2      +
244 1254 2      Now FIND the record. If any error, signal it.
245 1255 2      -
246 1256 2      +
247 1257 2      Set LUB$V_FIND_LAST to zero. If successful, we will set to 1.
248 1258 2      -
249 1259 2          CCB [LUB$V_FIND_LAST] = 0;
250 1260 2
```



```
251 1261 3 IF ( NOT $FIND (RAB = .CCB))
252 1262 THEN
253 1263 BEGIN
254 1264
255 1265 WHILE (.CCB [RAB$L_STS] EQL RMSS_RSA) DO
256 1266 BEGIN
257 1267 $WAIT (RAB = .CCB);
258 1268 $FIND (RAB = .CCB)
259 1269 END;
260 1270
261 1271 IF ( NOT .CCB [RAB$L_STS])
262 1272 THEN
263 1273 FOR$$SIGNAL_STO (
264 1274
265 1275 SELECTONEU .CCB [RAB$L_STS] OF
266 1276 SET
267 1277
268 1278 [RMSS_RLK] :
269 1279 FOR$K_SPERECLOC;
270 1280
271 1281 [RMSS_RNF, RMSS_EOF] :
272 1282 FOR$K_ATTACNON;
273 1283
274 1284 [OTHERWISE] :
275 1285 FOR$K_FINERR;
276 1286 TES);
277 1287
278 1288 END;
279 1289
280 1290 + We succeeded. Set LUB$V_FIND_LAST.
281 1291 -
282 1292 CCB [LUB$V_FIND_LAST] = 1;
283 1293 +
284 1294 Store the associated variable
285 1295 -
286 1296 FOR$$ASSOC ();
287 1297 +
288 1298 Return the I/O system to its last state.
289 1299 -
290 1300 FOR$$CB_POP ();
291 1301 +
292 1302 Return a success IOSTAT code
293 1303 -
294 1304
295 1305 RETURN 0;
296 1306 END;
```

! of routine FOR\$FIND

```
.TITLE FOR$FIND
.IDENT \2-016\

.EXTRN FOR$$IOSTAT_HND
.EXTRN FOR$$OPEN_DEFLT
.EXTRN FOR$$SIGNAL_STO
.EXTRN FOR$$ASSOC, FOR$$CB_PUSH
.EXTRN FOR$$CB_POP, SY$$FIND
.EXTRN SY$$WAIT
```

|           |           |    |      | .PSECT | _FOR\$CODE,NOWRT, SHR, PIC,2 |                              |      |
|-----------|-----------|----|------|--------|------------------------------|------------------------------|------|
|           |           |    | 081C | 00000  | .ENTRY                       | FOR\$FIND, Save R2,R3,R4,R11 | 1147 |
| 54        | 00000000G | 00 | 9E   | 00002  | MOVAB                        | SYSS\$FIND, R4               |      |
| 53        | 00000000G | 00 | 9E   | 00009  | MOVAB                        | FOR\$\$SIGNAL_STO, R3        |      |
| 5E        |           | 04 | C2   | 00010  | SUBL2                        | #4, SP                       |      |
|           |           | 7E | D4   | 00013  | CLRL                         | L_ERR_EQL PRES               | 1198 |
|           | 04        | AE | D4   | 00015  | CLRL                         | L_UNWIND ACTION              |      |
| 6D        | 00C0      | CF | DE   | 00018  | MOVAL                        | 14\$, (FPT)                  |      |
| 04        |           | 01 | D0   | 0001D  | MOVL                         | #1, L_UNWIND ACTION          | 1217 |
| 02        |           | 6C | 91   | 00021  | CMPB                         | (AP), #2                     | 1219 |
|           |           | 06 | 1B   | 00024  | BLEQU                        | 1\$                          |      |
| 6E        | 0C        | AC | D0   | 00026  | MOVL                         | ERR_EQL, L_ERR_EQL_PRES      |      |
|           |           | 02 | 11   | 0002A  | BRB                          | 2\$                          |      |
|           |           | 6E | D4   | 0002C  | CLRL                         | L_ERR_EQL_PRES               |      |
|           |           | 50 | D4   | 0002E  | CLRL                         | R0                           | 1225 |
| 52        | 04        | AC | D0   | 00030  | MOVL                         | UNIT, R2                     |      |
|           | 00000000G | 00 | 16   | 00034  | JSB                          | FOR\$\$CB PUSH               |      |
|           | 04        | AE | D4   | 0003A  | CLRL                         | L_UNWIND ACTION              | 1229 |
| 05        | FC        | AB | 04   | E0     | BBS                          | #4, -4(CCB), 3\$             | 1235 |
|           |           | 1A | DD   | 00042  | PUSHL                        | #26                          |      |
| 63        |           | 01 | FB   | 00044  | CALLS                        | #1, FOR\$\$SIGNAL_STO        |      |
| 0D        | FC        | AB | E8   | 00047  | BLBS                         | -4(CCB), 4\$                 | 1241 |
|           |           | 02 | DD   | 0004B  | PUSHL                        | #2                           |      |
|           |           | 01 | DD   | 0004D  | PUSHL                        | #1                           |      |
|           |           | 01 | DD   | 0004F  | PUSHL                        | #1                           |      |
| 00000000G | 00        | 03 | FB   | 00051  | CALLS                        | #3, FOR\$\$OPEN_DEFLT        |      |
| 52        | 08        | AC | D0   | 00058  | MOVL                         | REC_NO, R2                   | 1248 |
|           |           | 0B | 13   | 0005C  | BEQL                         | 5\$                          |      |
|           | E4        | AB | D5   | 0005E  | TSTL                         | -28(CCB)                     |      |
|           |           | 0B | 13   | 00061  | BEQL                         | 6\$                          |      |
| E4        | AB        | 52 | D1   | 00063  | CMPL                         | R2, -28(CCB)                 |      |
|           |           | 05 | 1B   | 00067  | BLEQU                        | 6\$                          |      |
|           |           | 19 | DD   | 00069  | PUSHL                        | #25                          | 1250 |
|           |           | 01 | FB   | 0006B  | CALLS                        | #1, FOR\$\$SIGNAL_STO        |      |
| 63        |           | 52 | D0   | 0006E  | MOVL                         | R2, -32(CCB)                 | 1252 |
| E0        | AB        | 08 | 8A   | 00072  | BICB2                        | #8, -96(CCB)                 | 1259 |
| A0        | AB        | 5B | DD   | 00076  | PUSHL                        | CCB                          | 1261 |
|           |           | 01 | FB   | 00078  | CALLS                        | #1, SYSS\$FIND               |      |
| 64        |           | 50 | E8   | 0007B  | BLBS                         | R0, 13\$                     |      |
| 4A        |           | AB | D1   | 0007E  | CMPL                         | 8(CCB), #99034               | 1265 |
| 000182DA  | 8F        | 08 | 10   | 12     | BNEQ                         | 8\$                          |      |
|           |           | 5B | DD   | 00088  | PUSHL                        | CCB                          | 1267 |
| 00000000G | 00        | 01 | FB   | 0008A  | CALLS                        | #1, SYSS\$WAIT               |      |
|           |           | 5B | DD   | 00091  | PUSHL                        | CCB                          | 1268 |
| 64        |           | 01 | FB   | 00093  | CALLS                        | #1, SYSS\$FIND               |      |
|           |           | E6 | 11   | 00096  | BRB                          | 7\$                          |      |
|           |           | AB | E8   | 00098  | BLBS                         | 8(CCB), 13\$                 | 1271 |
| 2C        | 08        | AB | D0   | 0009C  | MOVL                         | 8(CCB), R0                   | 1275 |
| 50        |           | 50 | D1   | 000A0  | CMPL                         | R0, #98986                   | 1278 |
| 000182AA  | 8F        | 04 | 12   | 000A7  | BNEQ                         | 9\$                          |      |
|           |           | 34 | DD   | 000A9  | PUSHL                        | #52                          |      |
|           |           | 18 | 11   | 000AB  | BRB                          | 12\$                         |      |
| 0001827A  | 8F        | 50 | D1   | 000AD  | CMPL                         | R0, #98938                   | 1281 |
|           |           | 09 | 13   | 000B4  | BEQL                         | 10\$                         |      |
| 000182B2  | 8F        | 50 | D1   | 000B6  | CMPL                         | R0, #98994                   |      |



|  |  |  |    |      |       |       |        |                       |  |
|--|--|--|----|------|-------|-------|--------|-----------------------|--|
|  |  |  | 04 | 12   | 000BD |       | BNEQ   | 11\$                  |  |
|  |  |  | 24 | DD   | 000BF | 10\$: | PUSHL  | #36                   |  |
|  |  |  | 02 | 11   | 000C1 |       | BRB    | 12\$                  |  |
|  |  |  | 39 | DD   | 000C3 | 11\$: | PUSHL  | #57                   |  |
|  |  |  | 01 | FB   | 000C5 | 12\$: | CALLS  | #1, FOR\$\$SIGNAL_STO |  |
|  |  |  | 08 | 88   | 000C8 | 13\$: | BISB2  | #8, -96(CCB)          |  |
|  |  |  | 00 | FB   | 000CC |       | CALLS  | #0, FOR\$\$ASSOC      |  |
|  |  |  | 00 | 16   | 000D3 |       | JSB    | FOR\$\$CB_POP         |  |
|  |  |  | 50 | D4   | 000D9 |       | CLRL   | R0                    |  |
|  |  |  |    | 04   | 000DB |       | RET    |                       |  |
|  |  |  |    | 0000 | 000DC | 14\$: | .WORD  | Save nothing          |  |
|  |  |  |    | AC   | D0    | 000DE | MOVL   | 8(AP), R0             |  |
|  |  |  |    | A0   | D0    | 000E2 | MOVL   | 4(R0), R0             |  |
|  |  |  |    | A0   | 9F    | 000E6 | PUSHAB | L_ERR_EQL PRES        |  |
|  |  |  |    | A0   | 9F    | 000E9 | PUSHAB | L_UNWIND_ACTION       |  |
|  |  |  |    | 02   | DD    | 000EC | PUSHL  | #2                    |  |
|  |  |  |    | 5E   | DD    | 000EE | SP     |                       |  |
|  |  |  |    | AC   | 7D    | 000F0 | MOVQ   | 4(AP), -(SP)          |  |
|  |  |  |    | 03   | FB    | 000F4 | CALLS  | #3, FOR\$\$IOSTAT_HND |  |
|  |  |  |    | 04   | 000FB |       | RET    |                       |  |

```
; Routine Size: 252 bytes,    Routine Base: _FOR$CODE + 0000
```

```

: 297      1307  1
: 298      1308  1  END
: 299      1309  1
: 300      1310  0  ELUDOM

```

```
! of module FOR$FIND
```

### PSECT SUMMARY

| Name       | Bytes | Attributes                                              |
|------------|-------|---------------------------------------------------------|
| _FOR\$CODE | 252   | NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2) |

## Library Statistics

| File                               | -----<br>Total | Symbols<br>Loaded | -----<br>Percent | Pages<br>Mapped | Processing<br>Time |
|------------------------------------|----------------|-------------------|------------------|-----------------|--------------------|
| \$255\$DUA28:[SYSLIB]STARLET.L32;1 | 9776           | 11                | 0                | 581             | 00:01.0            |

## COMMAND QUALIFIERS

FOR\$FIND  
2-016

L 9  
16-Sep-1984 00:22:43  
14-Sep-1984 12:31:58

VAX-11 Bliss-32 V4.0-742  
[FORRTL.SRC]FORFIND.B32;1

Page 10  
(3)

```
;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:FORFIND/OBJ=OBJ$:FORFIND MSRC$:FORFIND/UPDATE=(ENHS:FORFIND)
; Size:      252 code + 0 data bytes
; Run Time:   00:11.5
; Elapsed Time: 00:29.6
; Lines/CPU Min: 6822
; Lexemes/CPU-Min: 32119
; Memory Used: 142 pages
; Compilation Complete
```



0180 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY