

[illegible]

EXE

Mod

EDT

ED

ED
EDED
EDED
EDED
ED

ED

ED

ED

ED

ED
EDSYN
LBA

110

[illegible]

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL IIIII
LLLLLLLLLLL IIIII
SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSSS
SSSSSSSSS

```

.....


```
0001 0 %TITLE 'EDT$PRPARDRV - parse driver'
0002 0 MODULE EDT$PRPARDRV (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *****
0028 1
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 Parse driver.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: December 12, 1978
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 25-Feb-1981. This module was created by
0045 1 extracting routine EDT$SPA_DRIV from module PARSER.
0046 1 1-002 - Regularize headers. JBS 12-Mar-1981
0047 1 1-003 - Use the ASSERT macro. JBS 01-Jun-1981
0048 1 1-004 - Improve the appearance of the listing. JBS 17-Jun-1983
0049 1 --
0050 1
```

EDT\$PRPARDRV
V04-000

EDT\$PRPARDRV - parse driver
Declarations

L 3
16-Sep-1984 01:21:43
14-Sep-1984 12:24:13

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]PRPARDRV.BLI;1 Page 2 (2)

```
: 52      0051 1 %SBTTL 'Declarations'
: 53      0052 1
: 54      0053 1 TABLE OF CONTENTS:
: 55      0054 1
: 56      0055 1
: 57      0056 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 58      0495 1
: 59      0496 1 FORWARD ROUTINE
: 60      0497 1     EDT$PA_DRIV;
: 61      0498 1
: 62      0499 1
: 63      0500 1 INCLUDE FILES:
: 64      0501 1
: 65      0502 1
: 66      0503 1 REQUIRE 'EDTSRC:EDTREQ';
: 67      0638 1
: 68      0639 1 REQUIRE 'EDTSRC:PARLITS';
: 69      0923 1
: 70      0924 1 REQUIRE 'EDTSRC:PARSYNTBL';
: 71      1355 1
: 72      1356 1 REQUIRE 'EDTSRC:PARLABTBL';
: 73      1363 1
: 74      1364 1
: 75      1365 1 MACROS:
: 76      1366 1
: 77      1367 1     NONE
: 78      1368 1
: 79      1369 1 EQUATED SYMBOLS:
: 80      1370 1
: 81      1371 1     NONE
: 82      1372 1
: 83      1373 1 OWN STORAGE:
: 84      1374 1
: 85      1375 1     NONE
: 86      1376 1
: 87      1377 1 EXTERNAL REFERENCES:
: 88      1378 1
: 89      1379 1     In the routine
```

EDT
V04

: R
:

```

91 1380 1 %SBTTL 'EDT$PA_DRIV - parse driver'
92 1381 1
93 1382 1 GLOBAL ROUTINE EDT$PA_DRIV (
94 1383 1     TAB_INDEX
95 1384 1     ) =
96 1385 1
97 1386 1 ++
98 1387 1 FUNCTIONAL DESCRIPTION:
99 1388 1
100 1389 1     This is the parse table driver routine. TAB_INDEX indexes a byte in
101 1390 1     the parse table at which the parse starts.
102 1391 1
103 1392 1     This routine loops, fetching the next operator and execting until
104 1393 1     either a RETURN or ABORT command is executed, or a semantic routine
105 1394 1     returns failure.
106 1395 1
107 1396 1 FORMAL PARAMETERS:
108 1397 1
109 1398 1     TAB_INDEX             Index into the parse table at which to start the parse
110 1399 1
111 1400 1 IMPLICIT INPUTS:
112 1401 1
113 1402 1     NONE
114 1403 1
115 1404 1 IMPLICIT OUTPUTS:
116 1405 1
117 1406 1     EDT$PA_TBLPTR
118 1407 1
119 1408 1 ROUTINE VALUE:
120 1409 1
121 1410 1     The value of the routine is 1 for success and zero for failure.
122 1411 1
123 1412 1 SIDE EFFECTS:
124 1413 1
125 1414 1     Calls semantic routines
126 1415 1
127 1416 1 --
128 1417 1
129 1418 2 BEGIN
130 1419 2
131 1420 2 EXTERNAL ROUTINE
132 1421 2     EDT$PA_SCANTOK : NOVALUE,
133 1422 2     EDT$PA_TSTTOK,
134 1423 2     EDT$PA_SEMRUT;
135 1424 2
136 1425 2 EXTERNAL
137 1426 2     EDT$PA_TBLPTR;
138 1427 2
139 1428 2 LOCAL
140 1429 2     OP_CODE,
141 1430 2     OPERAND,
142 1431 2     SEM_ROUT,
143 1432 2     RETURN_ADDR;
144 1433 2
145 1434 2 EDT$PA_TBLPTR = PARSE_TABLE [.TAB_INDEX];
146 1435 2
147 1436 2 DO
```



```
148      BEGIN
149
150      !+ Fetch the op_code and the operand from the table and bump
151      !- past them.
152
153      OP_CODE = (.EDT$$PA_TBLPTR)<5, 3>;
154      OPERAND = (.EDT$$PA_TBLPTR)<0, 5>;
155      EDT$$PA_TBLPTR = .EDT$$PA_TBLPTR + 1;
156
157      IF (.OPERAND EQL 0)
158      THEN
159      !+ Here if operand is 0. This means it is a long-form operand.
160      !-
161
162      BEGIN
163      OPERAND = (.EDT$$PA_TBLPTR)<0, 8>;
164      EDT$$PA_TBLPTR = .EDT$$PA_TBLPTR + 1;
165      END;
166
167      !+ And now, let's case on the operand.
168      !-
169
170      CASE .OP_CODE FROM OPC_ABORT TO OPC_SELECT OF
171      SET
172
173      [OPC_ABORT] : ! This one is easy enough
174      BEGIN
175      RETURN (0);
176      END;
177
178      [OPC_ACTION] : ! Perform the specified action routine.
179      BEGIN
180
181      IF ( NOT EDT$$PA_SEMRUT (.OPERAND, EDT$$PA_TBLPTR)) THEN RETURN (0);
182
183      END;
184
185      [OPC_CALL] : ! Call; save current pointer and call yourself.
186      BEGIN
187      RETURN_ADDR = .EDT$$PA_TBLPTR;
188
189      IF EDT$$PA_DRIV (.LAB_TAB [.OPERAND - 1]) !
190      THEN
191      EDT$$PA_TBLPTR = .RETURN_ADDR
192      ELSE
193      RETURN (0);
194
195      END;
196
197      [OPC_GOTO] : ! Just get the new table address and continue.
198      BEGIN
199      EDT$$PA_TBLPTR = PARSE_TABLE [.LAB_TAB [.OPERAND - 1]];
200      END;
201
202      [OPC_OPTION] : ! Skip if the current token is not the optional one.
203      BEGIN
204
```

```
205 1494 4
206 1495 4
207 1496 4
208 1497 5
209 1498 5
210 1499 5
211 1500 5
212 1501 4
213 1502 4
214 1503 4
215 1504 4
216 1505 3
217 1506 3
218 1507 4
219 1508 4
220 1509 4
221 1510 4
222 1511 3
223 1512 3
224 1513 3
225 1514 4
226 1515 4
227 1516 4
228 1517 4
229 1518 4
230 1519 4
231 1520 4
232 1521 4
233 1522 4
234 1523 4
235 1524 5
236 1525 5
237 1526 5
238 1527 5
239 1528 6
240 1529 6
241 1530 6
242 1531 6
243 1532 5
244 1533 5
245 1534 5
246 1535 4
247 1536 4
248 1537 4
249 1538 4
250 1539 4
251 1540 4
252 1541 3
253 1542 3
254 1543 3
255 1544 4
256 1545 4
257 1546 3
258 1547 3
259 1548 3
260 1549 3
261 1550 3

IF EDT$$PA_TSTTOK (.OPERAND)
THEN
  BEGIN
    EDT$$PA_SCANTOK ();
    EDT$$PA_TBLPTR = .EDT$$PA_TBLPTR + 1;
  END
ELSE
  EDT$$PA_TBLPTR = PARSE_TABLE [.LAB_TAB [.(.EDT$$PA_TBLPTR)<0, 8> - 1]];
END;

[OPC_REQUIRE] : ! Abort if the current token is not the required one.
  BEGIN
    IF EDT$$PA_TSTTOK (.OPERAND) THEN EDT$$PA_SCANTOK () ELSE RETURN (0);
  END;

[OPC_SELECT] : ! Loop through the possible tokens, looking for it.
  BEGIN
    LOCAL
      SELECTED;

    SEM_ROUT = .(.EDT$$PA_TBLPTR)<0, 8>;
    EDT$$PA_TBLPTR = .EDT$$PA_TBLPTR + 1;
    SELECTED = 0;

    INCR I FROM 1 TO .OPERAND DO
      BEGIN
        IF EDT$$PA_TSTTOK (.(.EDT$$PA_TBLPTR)<0, 8>)
        THEN
          BEGIN
            EDT$$PA_TBLPTR = PARSE_TABLE [.LAB_TAB [.(.EDT$$PA_TBLPTR + 1)<0, 8> - 1]];
            SELECTED = .I;
            EXITLOOP;
          END;

          EDT$$PA_TBLPTR = .EDT$$PA_TBLPTR + 2;
        END;

        IF ( NOT EDT$$PA_SEMRUT (.SEM_ROUT, .SELECTED)) THEN RETURN (0);
        IF (.SELECTED NEQ 0) THEN EDT$$PA_SCANTOK ();
      END;

    [OPC_RETURN] : ! And another easy one.
      BEGIN
        RETURN (1);
      END;

    [OUTRANGE] :
      ASSERT (0);
  TES;
```


EDT\$PRPARDRV
V04-000

EDT\$PRPARDRV - parse driver
EDT\$SPA_DRIV - parse driver

C 4
16-Sep-1984 01:21:43
14-Sep-1984 12:24:13

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]PRPARDRV.BLI;1 Page 6 (3)

EDT\$
V04-

```

: 262      1551 3
: 263      1552 3
: 264      1553 3
: 265      1554 3
: 266      1555 3
: 267      1556 3
: 268      1557 1

                END
            UNTIL 0;

            ASSERT (0);
            RETURN (0);
            END;
```

! of routine EDT\$SPA_DRIV

.TITLE EDT\$PRPARDRV EDT\$PRPARDRV - parse driver
.IDENT \V04-000\

.PSECT _EDT\$CODE,NOWRT, SHR, PIC,2

```

48 03 51 36 4D 03 4E 40 4F 14 52 03 23 14 F7 00000 P.AAA: .BYTE
31 54 03 55 03 5C 04 5D 09 59 14 5A 03 4B 09 0000F
06 2C 71 2D 07 15 15 09 58 03 56 34 57 35 53 0001E
E1 83 27 83 39 C0 32 4F 84 2A 4F 33 03 6B 2F 0002D
27 60 28 60 2E 1C 7C A0 60 27 83 28 15 15 10 0003C
01 22 0D 24 E0 83 2F 6A C0 60 2A 26 60 31 84 0004B
5F 01 0B 01 60 01 18 24 1C 25 1D 23 03 01 79 0005A
27 11 08 09 28 1E 01 61 08 08 01 5B 01 19 01 00069
6C 01 0F 01 72 08 06 01 6F 01 0C 1B 10 08 04 00078
3F 2B 39 2D 3B 31 01 6D 01 33 01 74 01 14 01 00087
82 01 E3 00 23 13 32 01 7F 01 29 13 07 2E 2C 00096
86 01 04 01 23 01 E2 00 32 20 01 83 01 80 01 000A5
6E 01 88 01 81 01 20 01 21 01 86 01 89 01 EB 000B4
01 E2 60 49 C0 86 11 5E 11 03 01 4C 01 4A 01 000C3
34 01 E7 86 01 87 01 85 01 E2 86 01 65 01 68 000D2
E2 86 02 2E 02 26 02 25 02 24 02 28 02 27 02 000E1
66 01 69 01 84 01 E9 2F 20 86 02 2A 02 2F 01 000F0
22 86 01 3F 01 41 01 40 01 36 01 37 01 35 01 000FF
02 0A 01 E4 60 30 20 C1 22 C2 31 20 2E 20 C1 0010E
1D 01 03 01 17 0D 0D 00 2C 02 13 02 0D 02 12 0011D
01 22 01 09 01 1E 0C 1B 17 10 01 62 01 1C 01 0012C
06 01 0C 01 04 01 11 01 08 01 19 01 0B 01 18 0013B
01 2C 01 2B 20 2D 22 31 01 33 01 14 01 0F 01 0014A
64 2B 01 2A E5 00 23 01 46 01 32 01 29 01 07 00159
04 E4 60 2C 20 60 29 20 19 30 18 0E 1D 1F 1F 00168
20 60 2D 20 60 2B 20 21 30 29 0E 2A 1F 26 01 00177
60 34 20 60 3A 20 60 3B 20 60 38 20 C1 60 39 00186
2C 01 13 01 0D 01 12 01 0A 01 E4 60 33 20 C1 00195
01 26 01 25 01 24 01 28 01 27 01 34 01 E7 00 001A4
63 04 E2 00 23 01 2A 01 2F 01 E2 00 23 01 2E 001B3
3E C5 37 20 67 C0 35 20 4C 00 36 20 1A 1B 1E 001C2
3D 4E 26 0E B7 38 80 37 3E A0 3A 60 3B 26 60 001D1
E6 4A 01 73 07 47 07 3C 07 3D 01 45 1C E5 60 001E0
44 A0 0B 16 0B 1A 10 50 10 38 12 3A 12 02 04 001EF
05 21 05 20 3D 05 05 42 3C 01 15 E6 60 36 01 001FE
C1 22 37 85 38 25 60 30 3B 3E 43 04 E2 0D 70 0020D
20 85 24 20 0D A5 25 20 85 29 05 A1 39 87 29 0021C
80 07 BA 8B 33 B6 3F 4A 20 20 87 26 20 4A 27 0022B
7D 41 75 16 7E 04 7A 21 E7 22 20 01 7B A0 32 0023A
04 A2 28 20 22 00 23 20 04 77 04 78 16 76 3A 00249
2B C1 22 04 A2 84 3C 20 25 C1 00258
      00 60 84 00267
```

```

-9, 20, 35, 3, 82, 20, 79, 64, 78, 3, 77, -
54, 81, 3, 72, 9, 75, 3, 90, 20, 89, 9, -
93, 4, 92, 3, 85, 3, 84, 49, 83, 53, 87, -
52, 86, 3, 88, 9, 21, 21, 7, 45, 113, 44, -
6, 47, 107, 3, 51, 79, 42, -124, 79, 50, -
-64, 57, -125, 39, -125, -31, 16, 21, 21, -
40, -125, 39, 96, -96, 124, 28, 46, 96, -
40, 96, 39, -124, 49, 96, 38, 42, 96, -
-64, 106, 47, -125, -32, 36, 13, 34, 1, -
121, 1, 3, 35, 29, 37, 28, 36, 24, 1, 96, -
1, 11, 1, 95, 1, 25, 1, 91, 1, 8, 8, 97, -
1, 30, 40, 9, 8, 17, 39, 4, 8, 16, 27, -
12, 1, 111, 1, 6, 8, 114, 1, 15, 1, 108, -
1, 20, 1, 116, 1, 51, 1, 109, 1, 49, 59, -
45, 57, 43, 63, 44, 46, 7, 19, 41, 1, -
127, 1, 50, 19, 35, 0, -29, 1, -126, 1, -
-128, 1, -125, 1, 32, 50, 0, -30, 1, 35, -
1, 4, 1, -122, -21, 1, -119, 1, -122, 1, -
33, 1, 32, 1, -127, 1, -120, 1, 110, 1, -
74, 1, 76, 1, 3, 17, 64, 17, -122, -64, -
73, 96, -30, 1, 104, 1, 101, 1, -122, -
-30, 1, -123, 1, -121, 1, -122, -25, 1, -
52, 2, 39, 2, 40, 2, 36, 2, 37, 2, 38, 2, -
46, 2, -122, -30, 1, 47, 2, 42, 2, -122, -
32, 47, -23, 1, -124, 1, 105, 1, 102, 1, -
53, 1, 55, 1, 54, 1, 64, 1, 65, 1, 63, 1, -
-122, 34, -63, 32, 46, 32, 49, -62, 34, -
-63, 32, 48, 96, -28, 1, 10, 2, 18, 2, -
13, 2, 19, 2, 44, 0, -3, 13, 23, 1, 3, 1, -
29, 1, 28, 1, 98, 1, 16, 23, 27, 12, 30, -
1, 9, 1, 34, 1, 24, 1, 11, 1, 25, 1, 8, -
1, 17, 1, 4, 1, 12, 1, 6, 1, 15, 1, 20, -
1, 51, 1, 49, 34, 45, 32, 43, 1, 44, 1, -
7, 1, 41, 1, 50, 1, 70, 1, 35, 0, -27, -
42, 1, 43, 100, 31, 31, 29, 14, 24, 48, -
25, 32, 41, 96, 32, 44, 96, -28, 4, 1, -
38, 31, 42, 14, 41, 48, 33, 32, 43, 96, -
32, 45, 96, 32, 57, 96, -63, 32, 56, 96, -
32, 59, 96, 32, 58, 96, 32, 52, 96, -63, -
32, 51, 96, -28, 1, 10, 1, 18, 1, 13, 1, -
19, 1, 44, 0, -25, 1, 52, 1, 39, 1, 40, -
1, 36, 1, 37, 1, 38, 1, 46, 1, 35, 0, -
-30, 1, 47, 1, 42, 1, 35, 0, -30, 4, 99, -
```


EDT\$PRPARDRV - parse driver
EDT\$PA_DRIV - parse driver

D 4
16-Sep-1984 01:21:43
14-Sep-1984 12:24:13

VAX-11 Bliss-32 V4.0-742 Page 7
DISK\$VMSMASTER:[EDT.SRC]PRPARDRV.BLI;1 (3)

EDTS
V04-

[illegible]

30	27	26	32	54	0	76	32	53	-64	-
103	32	55	-59	62	96	38	59	96	58	-
-96	62	55	-128	56	-73	14	38	78	-	-
61	96	-27	28	69	1	61	7	60	7	71
7	115	1	74	-26	4	2	18	58	18	56
16	80	16	26	11	22	11	-96	68	1	-
54	96	-26	21	1	60	66	5	5	61	32
5	33	5	112	13	-30	4	67	62	59	-
48	96	37	56	-123	55	34	-63	41	-	-
-121	57	-95	5	41	-123	32	37	-61	-	-
13	32	36	-123	32	39	74	32	38	-	-
-121	32	32	74	63	-74	51	-117	-70	-	-
7	-128	50	-96	123	1	32	34	-25	-	-
33	122	4	126	22	117	65	125	58	-	-
118	22	120	4	116	4	32	35	0	34	-
32	40	-94	4	-63	37	43	-94	4	-63	-
37	32	60	-124	-94	4	34	-63	43	-	-
-124	96	0								-
2										-
616	461	49	571	527	170	508	276	-		-
57	513	560	355	543	481	469	537	-		-
205	554	65	52	67	610	409	389	392	-	-
456	281	72	386	466	373	439	405	-		-
421	162	246	180	396	173	269	402	-		-
399	370	78	76	215	81	547	85	562	-	-
567	59	293	74	475	477	239	595	-		-
222	534	551	542	208	447	0	494	-		-

PARSE_TABLE=
LAB_TAB=

P.AAA
P.AAB

EXTRN
EXTRN
EXTRN
EXTRN

EDTSSPA_SCANTOK
EDTSSPA_TSTTOK, EDTSSPA_SEMRUT
EDTSSPA_PATBLPTR
EDTSSINTER_ERR

OFFC 00000

.ENTRY

EDTSSA_DRIV, Save R2,R3,R4,R5,R6,R7,R8,R9,-;	1382
R10,R11	:
EDTSSA_TSTTOK, R11	:
PARSE_TABLE, R10	:
EDTSSA_PA_TBLPTR, R9	:
PARSE_TABLE, R0	1434
@TAB_INDEX[R0], EDTSSA_PA_TBLPTR	:
EDTSSA_PA_TBLPTR, R0	1442
#5, #3, (R0), OP_CODE	:
#0, #5, (R0), OPERAND	1443
EDTSSA_PA_TBLPTR	1444
OPERAND	1446
2\$	:
EDTSSA_PA_TBLPTR, R0	1452
(R0), OPERAND	:
EDTSSA_PA_TBLPTR	1453
OP_CODE, #0, #7	1460
20\$-3\$,-	:
4\$-3\$,-	:
6\$-3\$,-	:
19\$-3\$,-	:
7\$-3\$,-	:

		5B	00000000G	00	9E	00002	
		5A	FD03	CF	9E	00009	
		59	00000000G	00	9E	0000E	
		50		6A	9E	00015	
		69	04 BC	40	9E	00018	
		50		69	D0	0001D	1\$:
		03		05	EF	00020	
57	60	05		00	EF	00025	
52	60			69	D6	0002A	
				52	D5	0002C	
				08	12	0002E	
		50		69	D0	00030	
		52		60	9A	00033	
				69	D6	00036	
				57	CF	00038	2\$:
00D4	07	00				0003C	3\$:
0080	002A	0019		00D8		00044	
	0076	004E		003F			

00000000G	00	00	FB	0004C	CALLS	9\$-3\$ -	
		C8	11	00053	BRB	11\$-3\$ -	
		8F	BB	00055	PUSHR	12\$-3\$	
00000000G	00	02	FB	00059	CALLS	#0, EDT\$INTER_ERR	1549
BA		50	E8	00060	BLBS	1\$	1460
	55	00AE	31	00063	BRW	#^M<R2,R9>	1471
	7E	69	D0	00066	MOVL	#2, EDT\$PA_SEMRUT	
8D	AF	026A	CA42	3C	00069	R0, 1\$	
	ED		01	FB	0006F	20\$	
	69		50	E9	00073	EDT\$PA_TBLPTR, RETURN_ADDR	1477
			55	D0	00076	LAB_TAB-2[OPERAND], -(SPT)	1479
			A2	11	00079	#1, EDT\$PA_DRIV	
	50		6A	9E	0007B	R0, 5\$	
	51	026A	CA42	3C	0007E	RETURN_ADDR, EDT\$PA_TBLPTR	1481
69	50		51	C1	00084	1\$	1489
			93	11	00088	PARSE TABLE, R0	
			52	DD	0008A	LAB_TAB-2[OPERAND], R1	
	6B		01	FB	0008C	R1, R0, EDT\$PA_TBLPTR	
00000000G	0B		50	E9	0008F	1\$	1460
	00		00	FB	00092	OPERAND	1495
			69	D6	00099	#1, EDT\$PA_TSTTOK	
			70	11	0009B	R0, 10\$	
	50		69	D0	0009D	#0, EDT\$PA_SCANTOK	1498
	50		60	9A	000A0	EDT\$PA_TBLPTR	1499
	51		6A	9E	000A3	18\$	1495
	53	026A	CA40	3C	000A6	EDT\$PA_TBLPTR, R0	
69	51		53	C1	000AC	(R0), R0	
			5B	11	000B0	PARSE TABLE, R1	
			52	DD	000B2	LAB_TAB-2[R0], R3	
	6B		01	FB	000B4	R3, R1, EDT\$PA_TBLPTR	
	5A		50	E9	000B7	18\$	1460
			4A	11	000BA	OPERAND	1509
	50		69	D0	000BC	#1, EDT\$PA_TSTTOK	
	56		60	9A	000BF	R0, 20\$	
			69	D6	000C2	17\$	
			53	7C	000C4	EDT\$PA_TBLPTR, R0	1519
			28	11	000C6	(R0), SEM_ROUT	
	50		69	D0	000C8	EDT\$PA_TBLPTR	1520
	7E		60	9A	000CB	I	1523
	6B		01	FB	000CE	15\$	
	19		50	E9	000D1	EDT\$PA_TBLPTR, R0	1526
	50		69	D0	000D4	#1, EDT\$PA_TSTTOK	
	50	01	A0	9A	000D7	R0, 14\$	
	51		6A	9E	000DB	EDT\$PA_TBLPTR, R0	1529
	58	026A	CA40	3C	000DE	1(R0), R0	
69	51		58	C1	000E4	PARSE TABLE, R1	
	54		53	D0	000E8	LAB_TAB-2[R0], R8	
			07	11	000EB	R8, R1, EDT\$PA_TBLPTR	
	69		02	C0	000ED	I, SELECTED	1530
D4	53		52	F3	000F0	16\$	1528
			54	DD	000F4	#2, EDT\$PA_TBLPTR	1534
			56	DD	000F6	OPERAND, I, 13\$	1523
00000000G	00		02	FB	000F8	SELECTED	1537
	12		50	E9	000FF	SEM_ROUT	
						#2, EDT\$PA_SEMRUT	
						R0, 20\$	

EDT\$PRPARDRV - parse driver
EDT\$\$PA_DRV - parse driver

F 4
16-Sep-1984 01:21:43
14-Sep-1984 12:24:13

VAX-11 Bliss-32 V4.0-742 Page 9
DISK\$VMSMASTER:[EDT.SRC]PRPARDRV.BLI;1 (3)

00000000G 00
50

54	D5	00102	
82	13	00104	
00	FB	00106	17\$:
FF0D	31	0010D	18\$:
01	D0	00110	19\$:
	04	00113	
50	D4	00114	20\$:
	04	00116	

```
TSTL      SELECTED
BEQL      8$
CALLS     #0, EDT$$$PA_SCANTOK
BRW       1$
MOVL      #1, R0
RET
CLRL      R0
RET
```

: 1539
 :
 : 1460
 : 1545
 :
 : 1557

```
; Routine Size: 279 bytes,    Routine Base: _EDT$CODE + 02F0
```

```

; 269      1558 1
; 270      1559 1 !<BLF/PAGE>

```

EDT\$PRPARDRV
V04-000

EDT\$PRPARDRV - parse driver
EDT\$PA_DRIV - parse driver

G 4
16-Sep-1984 01:21:43
14-Sep-1984 12:24:13

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]PRPARDRV.BLI;1 Page 10
(4)

: 272 1560 1 END
: 273 1561 1
: 274 1562 0 ELUDOM

! of module EDT\$PRPARDRV

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	1031	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	17	4	40	00:00.2
\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:PRPARDRV/OBJ=OBJ\$:PRPARDRV MSRC\$:PRPARDRV.BLI/UPDATE=(ENHS:PRPARDRV)

: Size: 279 code + 752 data bytes
: Run Time: 00:28.8
: Elapsed Time: 00:35.5
: Lines/CPU Min: 3254
: Lexemes/CPU-Min: 11385
: Memory Used: 181 pages
: Compilation Complete

0138

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY