

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT

:

:

:

:

SECRET

```
PPPPPPPP  RRRRRRRR  PPPPPPPP  AAAAAA  RRRRRRRR  CCCCCCCC  000000  MM  MM
PPPPPPPP  RRRRRRRR  PPPPPPPP  AAAAAA  RRRRRRRR  CCCCCCCC  000000  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MMMM  MMMM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MMMM  MMMM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MM  MM  MM
PPPPPPPP  RRRRRRRR  PPPPPPPP  AA      AA  RRRRRRRR  CCCCCCCC  00      00  MM  MM  MM
PPPPPPPP  RRRRRRRR  PPPPPPPP  AA      AA  RRRRRRRR  CCCCCCCC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AAAAAAAAAA  RR      RR  CC      CC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AAAAAAAAAA  RR      RR  CC      CC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CC      CC  00      00  MM  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CCCCCCCC  000000  MM  MM  MM
PP      PP  RR      RR  PP      PP  AA      AA  RR      RR  CCCCCCCC  000000  MM  MM  MM
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```



```
0001 0 %TITLE 'EDT$PRPARCOM - parse a command'
0002 0 MODULE EDT$PRPARCOM (
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 a command.
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$SPA_CMD from module PARSER.
0046 1 1-002 - Regularize headers. JBS 12-Mar-1981
0047 1 1-003 - Use new message codes. JBS 04-Aug-1981
0048 1 1-004 - Define a synonym for the entry point name so a call to
0049 1 this routine can be hidden from the overlay analyzer. JBS 16-Mar-1982
0050 1 1-005 - Add CRLF if TI WRSTR is being used. SMB 02-Jul-1982
0051 1 1-006 - Use EDT$FMT_CRLF instead of EDT$OUT_FMTBUF. JBS 05-Jul-1982
0052 1 1-007 - Use EDT$FMT_STR instead of EDT$FMT_CIT, so EDT$G_PRV_COL is kept
0053 1 up to date. JBS 05-Oct-1982
0054 1 --
0055 1
```

EDT\$PRPARCOM
V04-000

EDT\$PRPARCOM - parse a command
Declarations

K 2
16-Sep-1984 01:20:40
14-Sep-1984 12:24:12

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

```

57 0056 1 %SBTTL 'Declarations'
58 0057 1
59 0058 1 | TABLE OF CONTENTS:
60 0059 1 |
61 0060 1
62 0061 1 REQUIRE 'EDTSRC:TRAROUNAM';
63 0500 1
64 0501 1 FORWARD ROUTINE
65 0502 1 EDT$SPA_CMD;
66 0503 1
67 0504 1 |
68 0505 1 | INCLUDE FILES:
69 0506 1 |
70 0507 1
71 0508 1 REQUIRE 'EDTSRC:EDTREQ';
72 0643 1
73 0644 1 REQUIRE 'EDTSRC:PARLITS';
74 0928 1
75 0929 1 |
76 0930 1 | MACROS:
77 0931 1 |
78 0932 1 | NONE
79 0933 1 |
80 0934 1 | EQUATED SYMBOLS:
81 0935 1 |
82 0936 1 |
83 0937 1 GLOBAL BIND
84 0938 1 ROUTINE
85 0939 1 EDT$SPA_CMD_NOOVERLAY_REF = EDT$SPA_CMD;
86 0940 1
87 0941 1 |
88 0942 1 | OWN STORAGE:
89 0943 1 |
90 0944 1 | NONE
91 0945 1 |
92 0946 1 | EXTERNAL REFERENCES:
93 0947 1 |
94 0948 1 | In the routine
```



```

96 0949 1 %SBTTL 'EDT$PA_CMD - parse a command'
97 0950 1
98 0951 1 GLOBAL ROUTINE EDT$PA_CMD ! Parse a command
99 0952 1 =
100 0953 1
101 0954 1 ++
102 0955 1 FUNCTIONAL DESCRIPTION:
103 0956 1
104 0957 1 This routine is called to parse a single command on the current command line.
105 0958 1 EDT$PA_CMD_BUF points to the first character of the command. If the
106 0959 1 parse is successful, a 1 is returned and the parsing stack contains a
107 0960 1 description of the command. EDT$PA_CMD_BUF is left pointing at the ';'
108 0961 1 or '!' which are the only valid terminators of commands. If an error occurs, a
109 0962 1 0 is returned, and EDT$PA_CMD_BUF is forced to point to the '!' at the
110 0963 1 end of the current command line, to prevent the rest of the commands from being
111 0964 1 executed.
112 0965 1
113 0966 1 FORMAL PARAMETERS:
114 0967 1
115 0968 1 NONE
116 0969 1
117 0970 1 IMPLICIT INPUTS:
118 0971 1
119 0972 1 EDT$ST_CMD_BUF
120 0973 1 EDT$PA_CMD_BUF
121 0974 1 EDT$PA_CMD_END
122 0975 1 EDT$G_CMD_LEN
123 0976 1 EDT$G_INP_SRC
124 0977 1 EDT$G_VFY
125 0978 1
126 0979 1 IMPLICIT OUTPUTS:
127 0980 1
128 0981 1 EDT$G_PA_CURCMD
129 0982 1 EDT$G_PA_SP
130 0983 1 EDT$PA_PA_CURTOK
131 0984 1 EDT$G_PA_ERRNO
132 0985 1
133 0986 1 ROUTINE VALUE:
134 0987 1
135 0988 1 1 = parse was successful
136 0989 1 0 = parse failed, EDT$G_PA_ERRNO set
137 0990 1
138 0991 1 SIDE EFFECTS:
139 0992 1
140 0993 1 MANY
141 0994 1
142 0995 1 --
143 0996 1
144 0997 2 BEGIN
145 0998 2
146 0999 2 EXTERNAL ROUTINE
147 1000 2 EDT$FMT_CRLF, ! Terminate an output line
148 1001 2 EDT$FMT_CH,
149 1002 2 EDT$FMT_STR,
150 1003 2 EDT$PA_SCANTOK : NOVALUE, ! Get the next token
151 1004 2 EDT$PA_GETCH : NOVALUE, ! Get the next character from the input line
152 1005 2 EDT$PA_TSTMCCAL, ! Test if the current token is a macro name
```

EDT\$PRPARCOM
V04-000

EDT\$PRPARCOM - parse a command
EDT\$PA_CMD - parse a command

M 2
16-Sep-1984 01:20:40
14-Sep-1984 12:24:12

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[EDT.SRC]PRPARCOM.BLI;1 Page 4
(3)

```
153      1006      2      EDT$PA_DRIV,          ! Parse table driver
154      1007      2      EDT$MSG_TOSTR;
155      1008      2
156      1009      2      EXTERNAL
157      1010      2      EDT$ST_CMD_BUF,          ! Command line buffer.
158      1011      2      EDT$PA_CMD_BUF,          ! Pointer into command buffer.
159      1012      2      EDT$PA_CMD_END,          ! Pointer to end of current command.
160      1013      2      EDT$G_CMD_LEN,          ! Length of command.
161      1014      2      EDT$G_INP_SRC,          ! input source
162      1015      2      EDT$G_VFY,              ! verify switch
163      1016      2      EDT$G_PA_CURCMD : REF NODE_BLOCK,
164      1017      2      EDT$PA_CURTOK,          ! start of the current token
165      1018      2      EDT$G_PA_ERRNO,          ! Error number of parsing error.
166      1019      2      EDT$G_PA_SP;
167      1020      2
168      1021      2      MESSAGES ((UNXCHRAFT, UNRCOM));
169      1022      2
170      1023      2      !+ Initialize the command node pointer and the parsing stack pointer.
171      1024      2      !-
172      1025      2      EDT$G_PA_CURCMD = 0;
173      1026      2      EDT$G_PA_SP = -1;
174      1027      2      EDT$PA_CURTOK = .EDT$PA_CMD_BUF;
175      1028      2      EDT$PA_GETCH ();
176      1029      2      EDT$PA_SCANTOK ();
177      1030      2      EDT$G_PA_ERRNO = 0;
178      1031      2
179      1032      2      IF (EDT$PA_TSTMACCAL () OR EDT$PA_DRIV (0))
180      1033      2      THEN
181      1034      2      BEGIN
182      1035      2      EDT$PA_CMD_BUF = .EDT$PA_CURTOK;
183      1036      2
184      1037      2      IF (CH$RCHAR (.EDT$PA_CMD_BUF) EQL %C';')
185      1038      2      THEN
186      1039      2      BEGIN
187      1040      2      EDT$PA_CMD_BUF = CH$PLUS (.EDT$PA_CMD_BUF, 1);
188      1041      2      RETURN (1);
189      1042      2      END
190      1043      2      ELSE
191      1044      2
192      1045      2      IF (CH$RCHAR (.EDT$PA_CMD_BUF) EQL %C'!') THEN RETURN (1) ELSE EDT$G_PA_ERRNO = EDT$UNXCHRAFT;
193      1046      2
194      1047      2      END
195      1048      2      ELSE
196      1049      2      !+
197      1050      2      ! If neither call was successful, but no error number assigned,
198      1051      2      ! give the unrecognized command diagnostic.
199      1052      2      !-
200      1053      2
201      1054      2      IF (.EDT$G_PA_ERRNO EQL 0) THEN EDT$G_PA_ERRNO = EDT$UNRCOM;
202      1055      2
203      1056      2      !+
204      1057      2      ! Print the command with an indication of where the error is.
205      1058      2      !-
206      1059      2
207      1060      2      IF ((.EDT$G_INP_SRC NEQ INP_TERM) AND (.EDT$G_VFY EQL 0))
208      1061      2      THEN
209      1062      2      BEGIN
```


EDT\$PRPARCOM
V04-000

EDT\$PRPARCOM - parse a command
EDT\$PA_CMD - parse a command

N 2
16-Sep-1984 01:20:40
14-Sep-1984 12:24:12

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]PRPARCOM.BLI;1 (3) Page 5

```

210      1063      3      EDT$FMT_CH (%C' ');
211      1064      3      EDT$FMT_STR (EDT$ST_CMD_BUF, .EDT$G_CMD_LEN);
212      1065      3      EDT$FMT_CRLF ();
213      1066      2      END;
214      1067      2
215      1068      2      DECR I FROM CH$DIFF (.EDT$A_PA_CURTOK, EDT$ST_CMD_BUF) TO 0 DO
216      1069      2      EDT$FMT_CH (%C' ');
217      1070      2
218      1071      2      EDT$FMT_CH (%C'^');
219      1072      2      EDT$FMT_CRLF ();
220      1073      2      !+
221      1074      2      !- Print the message corresponding to the error code.
222      1075      2
223      1076      2      EDT$MSG_TOSTR (.EDT$G_PA_ERRNO);
224      1077      2      EDT$FMT_CRLF ();
225      1078      2      EDT$A_CMD_BUF = .EDT$A_CMD_END;
226      1079      2      CH$WCHAR (%C'!', .EDT$A_CMD_END);
227      1080      2      RETURN (0);
228      1081      1      END;
```

! of routine EDT\$PA_CMD

.TITLE EDT\$PRPARCOM EDT\$PRPARCOM - parse a command
.IDENT \V04-000\

.EXTRN EDT\$FMT_CRLF, EDT\$FMT_CH
.EXTRN EDT\$FMT_STR, EDT\$PA_SCANTOK
.EXTRN EDT\$PA_GETCH, EDT\$PA_TSTMACCAL
.EXTRN EDT\$PA_DRIV, EDT\$MSG_TOSTR
.EXTRN EDT\$ST_CMD_BUF, EDT\$A_CMD_BUF
.EXTRN EDT\$A_CMD_END, EDT\$G_CMD_LEN
.EXTRN EDT\$G_INP_SRC, EDT\$G_VFY
.EXTRN EDT\$G_PA_CURCMD
.EXTRN EDT\$A_PA_CURTOK
.EXTRN EDT\$G_PA_ERRNO
.EXTRN EDT\$G_PA_SP, EDT\$UNXCHRAFT
.EXTRN EDT\$UNRCOM

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

.ENTRY EDT\$PA_CMD, Save R2,R3,R4,R5,R6,R7,R8
MOVAB EDT\$ST_CMD_BUF, R8
MOVAB EDT\$FMT_CRLF, R7
MOVAB EDT\$FMT_CH, R6
MOVAB EDT\$A_PA_CURTOK, R5
MOVAB EDT\$G_PA_ERRNO, R4
MOVAB EDT\$A_CMD_BUF, R3
CLRL EDT\$G_PA_CURCMD
MNEGL #1, EDT\$G_PA_SP
MOVL EDT\$A_CMD_BUF, EDT\$A_PA_CURTOK
CALLS #0, EDT\$PA_GETCH
CALLS #0, EDT\$PA_SCANTOK
CLRL EDT\$G_PA_ERRNO
CALLS #0, EDT\$PA_TSTMACCAL
BLBS R0, 1\$
CLRL -(SP)
CALLS #1, EDT\$PA_DRIV
BLBC R0, 5\$

```

                                01FC 00000
58 00000000G 00 9E 00002
57 00000000G 00 9E 00009
56 00000000G 00 9E 00010
55 00000000G 00 9E 00017
54 00000000G 00 9E 0001E
53 00000000G 00 9E 00025
   00000000G 00 D4 0002C
00000000G 00 01 CE 00032
   65      63 D0 00039
00000000G 00 00 FB 0003C
00000000G 00 00 FB 00043
   64      D4 0004A
00000000G 00 00 FB 0004C
   0C      50 E8 00053
   7E      D4 00056
00000000G 00 01 FB 00058
   21      50 E9 0005F
```

```

: 0951
:
:
: 1025
: 1026
: 1027
: 1028
: 1029
: 1030
: 1032
:
```

EDT\$PRPARCOM
V04-000

EDT\$PRPARCOM - parse a command
EDT\$\$PA_CMD - parse a command

B 3
16-Sep-1984 01:20:40
14-Sep-1984 12:24:12

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

63	65	D0	00062	1\$:	MOVL	EDT\$\$A_PA_CURTOK, EDT\$\$A_CMD_BUF	:	1035
50	63	D0	00065		MOVL	EDT\$\$A_CMD_BUF, R0	:	1037
3B	60	91	00068		CMPB	(R0), #59	:	
	04	12	0006B		BNEQ	2\$	:	
	63	D6	0006D		INCL	EDT\$\$A_CMD_BUF	:	1040
	05	11	0006F		BRB	3\$	:	1041
21	60	91	00071	2\$:	CMPB	(R0), #33	:	1045
	04	12	00074		BNEQ	4\$	:	
50	01	D0	00076	3\$:	MOVL	#1, R0	:	
	04	00079			RET		:	
64	00000000G	8F	D0	0007A	4\$:	MOVL	#EDT\$_UNXCHRAFT, EDT\$\$G_PA_ERRNO	
		0B	11	00081		BRB	6\$	1032
		64	D5	00083	5\$:	TSTL	EDT\$\$G_PA_ERRNO	1054
		07	12	00085		BNEQ	6\$	
64	00000000G	8F	D0	00087		MOVL	#EDT\$_UNRCOM, EDT\$\$G_PA_ERRNO	
	00000000G	00	D5	0008E	6\$:	TSTL	EDT\$\$G_INP_SRC	1060
		1F	13	00094		BEQL	7\$	
	00000000G	00	D5	00096		TSTL	EDT\$\$G_VFY	
		17	12	0009C		BNEQ	7\$	
		20	DD	0009E		PUSHL	#32	1063
66		01	FB	000A0		CALLS	#1, EDT\$\$FMT_CH	
	00000000G	00	DD	000A3		PUSHL	EDT\$\$G_CMD_LEN	1064
		58	DD	000A9		PUSHL	R8	
00000000G	00	02	FB	000AB		CALLS	#2, EDT\$\$FMT_STR	
	67	00	FB	000B2		CALLS	#0, EDT\$\$FMT-CRLF	1065
	50	68	9E	000B5	7\$:	MOVAB	EDT\$\$T_CMD_BUF, R0	1068
52	65	50	C3	000B8		SUBL3	R0, EDT\$\$A_PA_CURTOK, R2	
		52	D6	000BC		INCL	I	
		05	11	000BE		BRB	9\$	
		20	DD	000C0	8\$:	PUSHL	#32	1069
	66	01	FB	000C2		CALLS	#1, EDT\$\$FMT_CH	
	F8	52	F4	000C5	9\$:	SOBGEQ	I, 8\$	
	7E	8F	9A	000C8		MOVZBL	#94, -(SP)	1071
	66	01	FB	000CC		CALLS	#1, EDT\$\$FMT_CH	
	67	00	FB	000CF		CALLS	#0, EDT\$\$FMT-CRLF	1072
		64	DD	000D2		PUSHL	EDT\$\$G_PA_ERRNO	1076
00000000G	00	01	FB	000D4		CALLS	#1, EDT\$\$MSG_TOSTR	
	67	00	FB	000DB		CALLS	#0, EDT\$\$FMT-CRLF	1077
	50	00	D0	000DE		MOVL	EDT\$\$A_CMD_END, R0	1078
	63	50	D0	000E5		MOVL	R0, EDT\$\$A_CMD_BUF	
	60	21	90	000E8		MOVB	#33, (R0)	1079
		50	D4	000EB		CLRL	R0	1080
		04	000ED		RET			1081

; Routine Size: 238 bytes, Routine Base: _EDT\$CODE + 0000

: 229 1082 1
: 230 1083 1 !<BLF/PAGE>


```
EDT$PRPARCOM - parse a command
EDT$SPA_CMD  - parse a command
```

16-Sep-1984 01:20:40
14-Sep-1984 12:24:12

VAX-11 Bliss-32 V4.0-742 Page 7
DISK\$VMSMASTER:[EDT.SRC]PRPARCOM.BLI:1 (4)

```

: 232      1084 1 END
: 233      1085 1
: 234      1086 0 ELUDOM

```

```
! of module EDT$PRPARCOM
```

EDT\$\$PA_CMD_NOOVERLAY_REF==
EDT\$\$PA_CMD

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	238	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	51	13	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$:PRPARCOM/OBJ=OBJ$:PRPARCOM MSRC$:PRPARCOM.BLI/UPDATE=(ENH$:PRPARCOM)
;

```

```
: Size:          238 code + 0 data bytes
: Run Time:      00:17.3
: Elapsed Time:  00:21.0
: Lines/CPU Min: 3770
: Lexemes/CPU-Min: 11923
: Memory Used:   126 pages
: Compilation Complete
```

EDT
V04

:
:
:
:
:
:
:

48
31
06
E1
27
01
5F
27
6C
3F
82
86
6E
01
34
E2
66
22
02
1D
01
06
01
64
04
20
60
2C
01
63
3E
3D
E6
44
05
C1
20
80
7D
04
2B

0138

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY