

PPPPPPPPPPPP		AAAAAAAAAA		TTTTTTTTTTTTTTTT		CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA		TTTTTTTTTTTTTTTT		CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA		TTTTTTTTTTTTTTTT		CCCCCCCCCCCC	HHH	HHH
PPP	PPP	AAA	AAA	TTT	CCC	HHH	HHH	
PPP	PPP	AAA	AAA	TTT	CCC	HHH	HHH	
PPP	PPP	AAA	AAA	TTT	CCC	HHH	HHH	
PPP	PPP	AAA	AAA	TTT	CCC	HHH	HHH	
PPP	PPP	AAA	AAA	TTT	CCC	HHH	HHH	
PPP	PPP	AAA	AAA	TTT	CCC	HHH	HHH	
PPPPPPPPPPPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPPPPPPPPPPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPPPPPPPPPPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPP		AAAAAAAAAAAAAAAA		TTT	CCC	HHH	HHH	
PPP		AAAAAAAAAAAAAAAA		TTT	CCC	HHH	HHH	
PPP		AAAAAAAAAAAAAAAA		TTT	CCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCCCCCCCCCCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCCCCCCCCCCC	HHH	HHH	
PPP		AAA	AAA	TTT	CCCCCCCCCCCC	HHH	HHH	

```

PPPPPPPP      AAAAAA      TTTTTTTTTT      LL      SSSSSSSS      TTTTTTTTTT
PPPPPPPP      AAAAAA      TTTTTTTTTT      LL      SSSSSSSS      TTTTTTTTTT
PP           PP   AA          AA          TT          LL          SS          TT
PP           PP   AA          AA          TT          LL          SS          TT
PP           PP   AA          AA          TT          LL          SS          TT
PP           PP   AA          AA          TT          LL          SS          TT
PPPPPPPP      AA          AA          TT          LL          SSSSSS        TT
PPPPPPPP      AA          AA          TT          LL          SSSSSS        TT
PP           AAAAAAAAAA      TT          LL          SS          TT
PP           AAAAAAAAAA      TT          LL          SS          TT
PP           AA          AA          TT          LL          SS          TT
PP           AA          AA          TT          LL          SS          TT
PP           AA          AA          TT          LLLLLLLLLL      SSSSSSSS        TT
PP           AA          AA          TT          LLLLLLLLLL      SSSSSSSS        TT
                                     ....
                                     ....
                                     ....
                                     ....

LL           IIIIII      SSSSSSSS
LL           IIIIII      SSSSSSSS
LL           II          SS
LL           II          SS
LL           II          SS
LL           II          SS
LL           II          SSSSSS
LL           II          SSSSSS
LL           II          SS
LL           II          SS
LL           II          SS
LL           II          SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```



```
1 0001 0 MODULE PATLST (
2 L 0002 0 %IF %VARIANT EQL 1
3 0003 0 %THEN
4 0004 0 ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE),
5 0005 0 %FI
6 0006 0 IDENT = 'V04-000') =
7 0007 1 BEGIN
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 FACILITY: PATCH
33 0033 1
34 0034 1 ++
35 0035 1 FUNCTIONAL DESCRIPTION:
36 0036 1
37 0037 1 CREATES AND ADDS ITEMS TO A LINKED LIST
38 0038 1
39 0039 1 Version: X01.05
40 0040 1
41 0041 1 History:
42 0042 1 Author:
43 0043 1 Carol Peters, 28 Oct 1976: Version 01
44 0044 1
45 0045 1 Modified by:
46 0046 1 Kathleen Morse, 19 Oct 1977: Version X01.00
47 0047 1
48 0048 1 V01-05 MTR0001 Mike Rhodes 07-Oct-1981
49 0049 1 Zero the linked argument list TAIL pointer also upon
50 0050 1 command completion. Routine PAT$FREE_ARG.
51 0051 1
52 0052 1 Revision history:
53 0053 1 NO DATE PROGRAMMER PURPOSE
54 0054 1 -- ----
55 0055 1
56 0056 1 00 11-OCT-77 K.D. MORSE ADAPT VERSION 8 FOR PATCH.
57 0057 1 01 05-JAN-78 K.D. MORSE NO CHANGES FOR VERS 9.
```

PATLST
V04-000

: 58
: 59
: 60
: 61
: 62
: 63

0058 1 :
0059 1 : 02 25-APR-78
0060 1 : 03 18-MAY-78
0061 1 : 04 13-JUN-78
0062 1 :
0063 1 :--

K.D. MORSE
K.D. MORSE
K.D. MORSE

1 7
16-Sep-1984 00:27:57
14-Sep-1984 12:52:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATLST.B32;1 Page 2 (1)

INSTEAD OWN STORAGE. (10)
CONVERT TO NATIVE COMPILER.
NO CHANGES FOR VERS 11-13.
ADD FAO COUNTS TO SIGNALS.

PA
VO

PATLST
V04-000

J 7
16-Sep-1984 00:27:57
14-Sep-1984 12:52:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATLST.B32;1 Page 3 (2)

```
: 65      0064 1 FORWARD ROUTINE
: 66      0065 1          PAT$ADD_ARG
: 67      0066 1          PAT$FREE_ARG : NOVALUE;
: 68      0067 1
: 69      0068 1 LIBRARY 'SYSS$LIBRARY:STARLET.L32';
: 70      0069 1 REQUIRE 'SRC$:PATPCT.REQ';
: 71      0109 1 REQUIRE 'SRC$:BSTRUC.REQ';
: 72      0185 1 REQUIRE 'SRC$:LISTEL.REQ';
: 73      0227 1 REQUIRE 'SRC$:VXSMAC.REQ';
: 74      0292 1 REQUIRE 'SRC$:SYSSER.REQ';
```

```
! CREATES A LINKED LIST OR ADDS A NEW ENTRY
! FREES A LINKED LIST
```

PATLST
V04-000

K 7
16-Sep-1984 00:27:57
15-Sep-1984 22:50:49

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[PATCH.SRC]SYSSER.REQ;1 Page (1) 4

: R0324 1
: R0325 1
: R0326 1
: R0327 1
: R0328 1

SWITCHES LIST (SOURCE);

EXTERNAL ROUTINE
PAT\$fao_out;

! formats a line and outputs to the terminal

PATLST
V04-000

L 7
16-Sep-1984 00:27:57
14-Sep-1984 12:52:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATLST.B32;1 Page 5 (2)

```
: 75      0374 1
: 76      0375 1 EXTERNAL ROUTINE
: 77      0376 1      PAT$FREEZ,
: 78      0377 1      PAT$FREERELEASE;
: 79      0378 1
: 80      0379 1 EXTERNAL
: 81      0380 1      PAT$GL_HEAD_LST,
: 82      0381 1      PAT$GL_TAIL_LST;
```

```
: GETS A PIECE OF FREE STORAGE
: RELEASES FREE STORAGE
```

```
: ADDRESS OF HEAD OF LINKED LIST
: ADDRESS OF TAIL OF LINKED LIST
```



```

84 0382 1 GLOBAL ROUTINE PAT$ADD_ARG (EXPRESSION) =
85 0383 1
86 0384 1 ++
87 0385 1 FUNCTIONAL DESCRIPTION:
88 0386 1
89 0387 1     ADDS AN ELEMENT TO A LINKED LIST OR CREATES A LINKED
90 0388 1     LIST AND INSERTS THE FIRST ELEMENT.
91 0389 1     THE ADDRESS OF THE HEAD OF THE LIST IS AT PAT$GL_HEAD_LST
92 0390 1     AND THE ADDRESS OF THE TAIL AT PAT$GL_TAIL_LST.
93 0391 1     THIS ROUTINE ONLY PUTS A SINGLE VALUE IN THE LIST ELEMENT,
94 0392 1     AND IT PUTS IT AT THE LONGWORD AFTER THE POINTER.
95 0393 1
96 0394 1 CALLING SEQUENCE:
97 0395 1
98 0396 1     PAT$ADD_ARG (EXPRESSION)
99 0397 1
100 0398 1 INPUTS:
101 0399 1
102 0400 1     EXPRESSION      - VALUE TO PUT IN LIST ELEMENT
103 0401 1
104 0402 1 IMPLICIT INPUTS:
105 0403 1
106 0404 1     THE HEAD AND TAIL OF THE LINKED LIST.
107 0405 1     THE LENGTH OF A LIST ELEMENT (LIST_ELEM_LEN).
108 0406 1
109 0407 1 OUTPUTS:
110 0408 1
111 0409 1     THE ADDRESS OF THE NEW ELEMENT IF SPACE FOR THE LIST
112 0410 1     ELEMENT WAS FOUND. OTHERWISE A ZERO.
113 0411 1
114 0412 1 IMPLICIT OUTPUTS:
115 0413 1
116 0414 1     NONE
117 0415 1
118 0416 1 ROUTINE VALUE:
119 0417 1
120 0418 1     THE POINTER TO THE ELEMENT, OR A ZERO IF NO ELEMENT WAS ALLOCATED.
121 0419 1
122 0420 1 SIDE EFFECTS:
123 0421 1
124 0422 1     THE LIST ELEMENT IS WRITTEN. THE TAIL OF THE LIST IS REWRITTEN.
125 0423 1     IF THE LIST WAS CREATED, BOTH THE HEAD AND THE TAIL ARE WRITTEN
126 0424 1     TO THEIR GLOBAL VARIABLE NAMES.
127 0425 1
128 0426 1 --
129 0427 1
130 0428 2 BEGIN
131 0429 2
132 0430 2 LOCAL
133 0431 2     POINTER;
134 0432 2
135 0433 2     POINTER = PAT$FREEZ (LIST_ELEM_LEN);
136 0434 3     IF (.PAT$GL_HEAD_LST EQL 0)
137 0435 2     THEN
138 0436 2         PAT$GL_HEAD_LST = .POINTER
139 0437 2     ELSE
140 0438 2         LIST_ELEM_FLINK (.PAT$GL_TAIL_LST) = .POINTER;
```


PATLST
V04-000

N 7
16-Sep-1984 00:27:57
14-Sep-1984 12:52:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATLST.B32;1 Page 7 (3)

```
: 141      0439 2 LIST_ELEM_FLINK (.POINTER) = 0;
: 142      0440 2 PAT$GL_TAIL_LST = .POINTER;
: 143      0441 2 LIST_ELEM_EXP1 (.POINTER) = .EXPRESSION;
: 144      0442 2 RETURN .POINTER
: 145      0443 1 END;
```

```
                                0004 00000
                                EF 9E 00002
                                04 DD 00009
00000000G EF 01 FB 0000B
                                62 D5 00012
                                05 12 00014
                                62 50 D0 00016
                                07 11 00019
00000000G FF 50 D0 0001B 1$:
                                60 D4 00022 2$:
00000000G EF 50 D0 00024
                                04 A0 04 AC D0 0002B
                                04 00030
```

; Routine Size: 49 bytes, Routine Base: _PAT\$CODE + 0000

```
.TITLE PATLST
.IDENT \V04-000\

.EXTRN PAT$FAO_OUT, PAT$FREEZ
.EXTRN PAT$FREEERELASE
.EXTRN PAT$GL_HEAD_LST
.EXTRN PAT$GL_TAIL_LST
.WEAK ACCESS_CHECK

.PSECT _PAT$CODE,NOWRT,2

.ENTRY PAT$ADD_ARG, Save R2
MOVAB PAT$GL_HEAD_LST, R2
PUSHL #4
CALLS #1, PAT$FREEZ
TSTL PAT$GL_HEAD_LST
BNEQ 1$
MOVL POINTER, PAT$GL_HEAD_LST
BRB 2$
MOVL POINTER, @PAT$GL_TAIL_LST
CLRL (POINTER)
MOVL POINTER, PAT$GL_TAIL_LST
MOVL EXPRESSION, 4(POINTER)
RET
```

: 0382
: 0433
: 0434
: 0436
: 0438
: 0439
: 0440
: 0441
: 0443

```

147 0444 1 GLOBAL ROUTINE PAT$FREE_ARG : NOVALUE =
148 0445 1
149 0446 1 ++
150 0447 1 FUNCTIONAL DESCRIPTION:
151 0448 1
152 0449 1     FREES EACH ELEMENT OF A LINKED LIST, AND ZEROES THE
153 0450 1     POINTER TO THE LIST.
154 0451 1     THIS ROUTINE IS INTENDED TO BE CALLED AT THE END OF
155 0452 1     PROCESSING OF A COMPLETE COMMAND LINE.
156 0453 1
157 0454 1 CALLING SEQUENCE:
158 0455 1
159 0456 1     PAT$FREE_ARG ( )
160 0457 1
161 0458 1 INPUTS:
162 0459 1
163 0460 1     NONE
164 0461 1
165 0462 1 IMPLICIT INPUTS:
166 0463 1
167 0464 1     THE HEAD OF THE LINKED LIST.
168 0465 1     THE LENGTH OF A LIST ELEMENT (LIST_ELEM_LEN).
169 0466 1
170 0467 1 OUTPUTS:
171 0468 1
172 0469 1     NONE
173 0470 1
174 0471 1 IMPLICIT OUTPUTS:
175 0472 1
176 0473 1     NONE
177 0474 1
178 0475 1 ROUTINE VALUE:
179 0476 1
180 0477 1     NOVALUE
181 0478 1
182 0479 1 SIDE EFFECTS:
183 0480 1
184 0481 1     THE ELEMENTS OF THE LIST ARE RETURNED TO THE FREE STORAGE POOL.
185 0482 1
186 0483 1 --
187 0484 1
188 0485 2 BEGIN
189 0486 2
190 0487 2 LOCAL
191 0488 2     POINTER1,
192 0489 2     POINTER2;
193 0490 2
194 0491 2 IF (POINTER1 = .PAT$GL_HEAD_LST) EQL 0
195 0492 2 THEN RETURN
196 0493 2 ELSE DO
197 0494 3     BEGIN
198 0495 3         POINTER2 = .LIST_ELEM_FLINK (.POINTER1);
199 0496 3         PAT$FREERELEASE (.POINTER1, LIST_ELEM_LEN);
200 0497 3     END
201 0498 2 UNTIL (POINTER1 = .POINTER2) EQL 0;
202 0499 2 PAT$GL_HEAD_LST = 0;
203 0500 2 PAT$GL_TAIL_LST = 0;
```


PATLST
V04-000

; 204

0501 1 END;

C 8
16-Sep-1984 00:27:57
14-Sep-1984 12:52:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATLST.B32;1 Page 9
(4)

		001C 00000	.ENTRY	PAT\$FREE_ARG, Save R2,R3,R4	: 0444
54	00000000G	EF 9E 00002	MOVAB	PAT\$GL_HEAD_LST, R4	: 0491
52		64 D0 00009	MOVL	PAT\$GL_HEAD_LST, POINTER1	: 0495
		1B 13 0000C	BEQL	2\$	: 0496
53		62 D0 0000E 1\$:	MOVL	(POINTER1), POINTER2	: 0498
		04 DD 00011	PUSHL	#4	: 0499
		52 DD 00013	PUSHL	POINTER1	: 0500
00000000G	EF	02 FB 00015	CALLS	#2, PAT\$FREERELEASE	: 0501
	52	53 D0 0001C	MOVL	POINTER2, POINTER1	: 0501
		ED 12 0001F	BNEQ	1\$	: 0501
		64 D4 00021	CLRL	PAT\$GL_HEAD_LST	: 0501
	00000000G	EF D4 00023	CLRL	PAT\$GL_TAIL_LST	: 0501
		04 00029 2\$:	RET		: 0501

; Routine Size: 42 bytes, Routine Base: _PAT\$CODE + 0031

PATLST
V04-000

D 8
16-Sep-1984 00:27:57
14-Sep-1984 12:52:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATLST.B32;1 Page 10
(5)

: 206 0502 1 END
: 207 0503 0 ELUDOM

PSECT SUMMARY

: Name Bytes Attributes
: _PAT\$CODE 91 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

: File Total Symbols Loaded Percent Pages Mapped Processing Time
: _\$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 0 0 581 00:01.0

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/VARIANT:1/LIS=LIS\$:PATLST/OBJ=OBJ\$:PATLST MSRC\$:PATLST/UPDATE=(ENH\$:PATLST)

: Size: 91 code + 0 data bytes
: Run Time: 00:05.0
: Elapsed Time: 00:28.2
: Lines/CPU Min: 5988
: Lexemes/CPU-Min: 20333
: Memory Used: 48 pages
: Compilation Complete

0302 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

