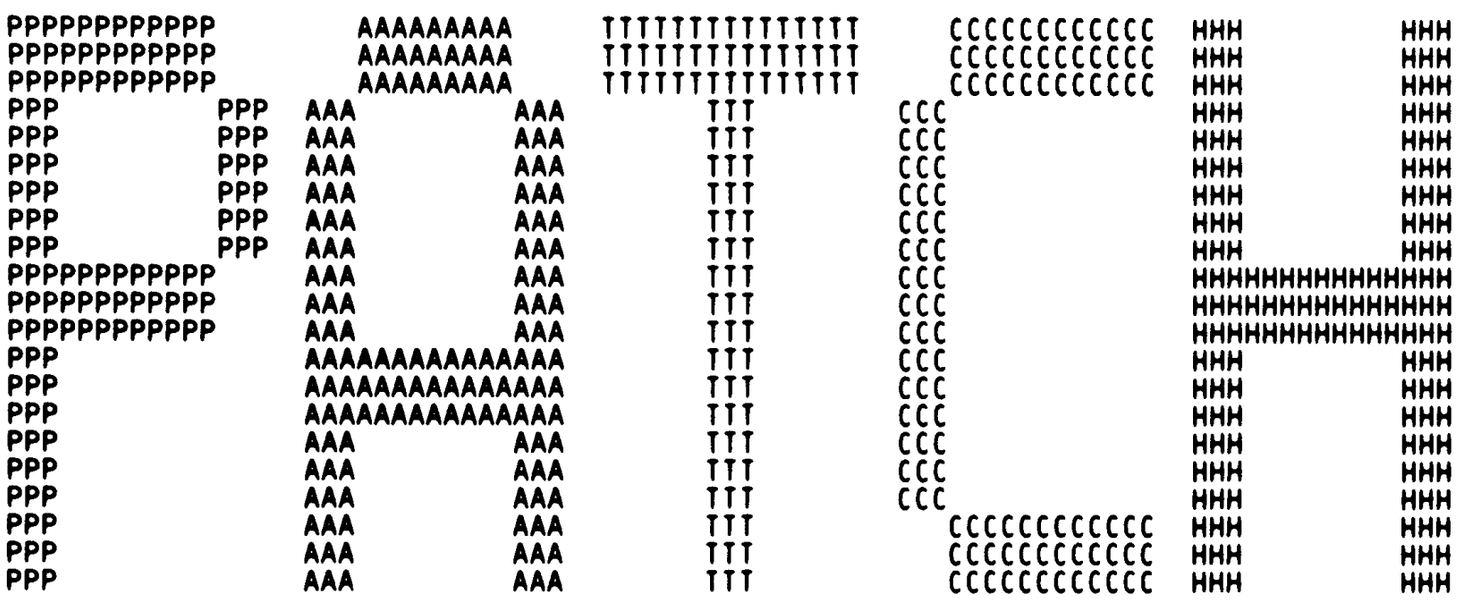
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPPPPPPPPPPP AAAAAAAAAAAAAAAA TTT CCC HHHHHHHHHHHHHHHH
PPP AAA AAA TTT CCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCCC HHH HHH

FILEID**PATIO

```
PPPPPPPP      AAAAAA    TTTTTTTTTT    IIIIII      000000
PPPPPPPP      AAAAAA    TTTTTTTTTT    IIIIII      000000
PP      PP   AA    AA       TT         II       00      00
PP      PP   AA    AA       TT         II       00      00
PP      PP   AA    AA       TT         II       00      00
PP      PP   AA    AA       TT         II       00      00
PPPPPPPP     AA    AA       TT         II       00      00
PPPPPPPP     AA    AA       TT         II       00      00
PP           AAAAAAAAAA     TT         II       00      00
PP           AAAAAAAAAA     TT         II       00      00
PP           AA    AA       TT         II       00      00      ....
PP           AA    AA       TT         II       00      00      ....
PP           AA    AA       TT       IIIIII      000000        ....
PP           AA    AA       TT       IIIIII      000000        ....

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
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

```
;     1        L 0001  0 MODULE PATIO (%IF %VARIANT EQL 1                                  ! I/O ROUTINES FOR PATCH
;     2          0002  0                 %THEN
;     3          0003  0                         ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
;     4          0004  0                                          NONEXTERNAL = LONG_RELATIVE),
;     5          0005  0                 %FI
;     6          0006  0                 IDENT = 'V04-000'
;     7          0007  0                 ) =
;     8          0008  1 BEGIN
;     9          0009  1
;    10          0010  1 !
;    11          0011  1 !**************************************************************************
;    12          0012  1 !*                                                                        *
;    13          0013  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    14          0014  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    15          0015  1 !*  ALL RIGHTS RESERVED.                                                   *
;    16          0016  1 !*                                                                        *
;    17          0017  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    18          0018  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    19          0019  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    20          0020  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    21          0021  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    22          0022  1 !*  TRANSFERRED.                                                           *
;    23          0023  1 !*                                                                        *
;    24          0024  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    25          0025  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    26          0026  1 !*  CORPORATION.                                                           *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    29          0029  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    30          0030  1 !*                                                                        *
;    31          0031  1 !*                                                                        *
;    32          0032  1 !**************************************************************************
;    33          0033  1
;    34          0034  1 !++
;    35          0035  1 ! FACILITY: PATCH
;    36          0036  1 !
;    37          0037  1 ! ABSTRACT: THESE ROUTINES HANDLE THE I/O AND CLOSE ALL FILES.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT: IT IS PART OF THE IMAGE FILE PATCH UTILTIY.
;    40          0040  1 !
;    41          0041  1 ! AUTHOR: K.D. MORSE     , CREATION DATE: 3-OCT-77
;    42          0042  1 !
;    43          0043  1 ! MODIFIED BY:
;    44          0044  1 !
;    45          0045  1 !       V03-004 MCN0161         Maria del C. Nasr       29-Mar-1984
;    46          0046  1 !               Pass correct FAB_ADDR to GETFILDSC.
;    47          0047  1 !
;    48          0048  1 !       V03-003 MCN0157         Maria del C. Nasr       14-Feb-1984
;    49          0049  1 !               We did a user file open of the input file, so deassign
;    50          0050  1 !               channel to close file.
;    51          0051  1 !
;    52          0052  1 !       V03-002 MTR0012         Mike Rhodes             16-Aug-1982
;    53          0053  1 !               Modify file names to remove duplicate file name useage
;    54          0054  1 !               between code and require files.
;    55          0055  1 !
;    56          0056  1 !       V03-001 MTR0007         Mike Rhodes             14-Jun-1982
;    57          0057  1 !               Use shared system messages.  Affected modules include:
```

```
;    58          0058  1 !               DYNMEM.B32, PATBAS.B32, PATCMD.B32, PATIHD.B32, PATINT.B32,
;    59          0059  1 !               PATIO.B32, PATMAI.B32, PATMSG.MSG, PATWRT.B32, and PATSPA.B32.
;    60          0060  1 !
;    61          0061  1 !               The shared messages are defined by DYNMEM.B32's invocation of
;    62          0062  1 !               SHRMSG.REQ and we simply link against these symbols.  They are
;    63          0063  1 !               declared as external literals below.
;    64          0064  1 !
;    65          0065  1 !       V02-008 MTR0001         Mike Rhodes             14-Oct-1981
;    66          0066  1 !               Modify routine PAT$CLOSEFILES to not worry about trying
;    67          0067  1 !               to delete virtual addresses that are mapped to the input
;    68          0068  1 !               image file, since these are deleted during image rundown.
;    69          0069  1 !
;    70          0070  1 !       V02-007 PCG0001         Peter George            02-FEB-1981
;    71          0071  1 !               Add require statement for LIB$:PATDEF.REQ
;    72          0072  1 !
;    73          0073  1 ! MODIFICATIONS:
;    74          0074  1 !
;    75          0075  1 !  NO   DATE            PROGRAMMER              PURPOSE
;    76          0076  1 !  --   ----            ----------              -------
;    77          0077  1 !
;    78          0078  1 !  01   14-FEB-78       K.D. MORSE              ADD ROUTINES PAT$WRITE_EXP1,
;    79          0079  1 !                                               PAT$WRITE_NAME, AND PAT$WRITE_INS.
;    80          0080  1 !  02   7-MAR-78        K.D. MORSE              ADD ROUTINE PAT$OUT_PAL_EXP AND
;    81          0081  1 !                                               CALLS TO IT.
;    82          0082  1 !  03   25-APR-78       K.D. MORSE              CONVERT TO NATIVE COMPILER.
;    83          0083  1 !  04   05-MAY-78       K.D. MORSE              ADD CALL TO PAT$REDUCE_INS.
;    84          0084  1 !  05   13-JUN-78       K.D. MORSE              ADD FAO COUNTS TO SIGNALS.
;    85          0085  1 !  06   15-JUN-78       K.D. MORSE              ADD OUTPUT TO APPENDED PATCH
;    86          0086  1 !                                               COMMAND TEXT BUFFERS.
;    87          0087  1 !
;    88          0088  1 !--
```

```
;    90          0089  1 !
;    91          0090  1 ! TABLE OF CONTENTS:
;    92          0091  1 !
;    93          0092  1
;    94          0093  1 FORWARD ROUTINE
;    95          0094  1         PAT$WRITEFILE : NOVALUE,                        ! OUTPUTS TO A FILE
;    96          0095  1         PAT$WRITE_EXP1 : NOVALUE,                       ! WRITES EXPRESSIONS TO COMMAND FILE
;    97          0096  1         PAT$WRITE_NAME : NOVALUE,                       ! WRITES NAMES TO COMMAND FILE
;    98          0097  1         PAT$WRITE_INS : NOVALUE,                        ! WRITES COMMANDS WITH INSTRUCTIONS TO COMMA
;    99          0098  1         PAT$OUT_PAL_EXP : NOVALUE,                      ! OUTPUTS PATCH AREA ADDRESSES AS NAME+OFFSE
;   100          0099  1         PAT$CLOSEFILES : NOVALUE;                       ! CLOSES ALL PATCH FILES OPEN
;   101          0100  1
;   102          0101  1 !
;   103          0102  1 ! INCLUDE FILES:
;   104          0103  1 !
;   105          0104  1
;   106          0105  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   107          0106  1 REQUIRE 'SRC$:VXSMAC.REQ';
;   108          0171  1 REQUIRE 'SRC$:PATPCT.REQ';                              ! DEFINE PSECTS
;   109          0211  1 REQUIRE 'SRC$:PREFIX.REQ';                              ! UTILITY MACROS
;   110          0399  1 REQUIRE 'SRC$:PATPRE.REQ';                              ! COMMAND LINE LITERALS
;   111          0562  1 REQUIRE 'LIB$:PATDEF.REQ';                          ! Defines literals
;   112          0616  1 REQUIRE 'LIB$:PATMSG.REQ';                              ! PATCH ERROR MESSAGE CODES
;   113          0790  1 REQUIRE 'SRC$:SYSLIT.REQ';                              ! DEFINE OUTPUT BUFFER SIZE
;   114          0840  1 REQUIRE 'SRC$:PATTER.REQ';                              ! PATCH COMMAND TOKENS
;   115          1047  1 REQUIRE 'SRC$:PATGEN.REQ';                              ! DEFINE CONTEXT AND MODE BITS
;   116          1269  1 REQUIRE 'SRC$:PATRTS.REQ';                              ! DEFINE MODULE CHAIN OFFSETS
;   117          2365  1 REQUIRE 'SRC$:BSTRUC.REQ';                              ! STRUCTURE DEFINITIONS
;   118          2441  1 REQUIRE 'SRC$:LISTEL.REQ';                              ! DEFINE ARGUMENT LIST OFFSETS
;   119          2483  1 REQUIRE 'SRC$:DLLNAM.REQ';                              ! DEFINE SYMBOLIC NAME STRUCTURES
;   120          2541  1
;   121          2542  1 !
;   122          2543  1 ! MACROS:
;   123          2544  1 !
;   124          2545  1
;   125          2546  1 !
;   126          2547  1 ! EQUATED SYMBOLS:
;   127          2548  1 !
;   128          2549  1
;   129          2550  1 !
;   130          2551  1 ! OWN STORAGE:
;   131          2552  1 !
;   132          2553  1
;   133          2554  1 !
;   134          2555  1 ! EXTERNAL REFERENCES:
;   135          2556  1 !
;   136          2557  1
;   137          2558  1 EXTERNAL ROUTINE
;   138          2559  1         PAT$FIND_VAL,                                   ! Searches user-defined symbols for closest
;   139          2560  1         PAT$FAO_PUT,                                    ! FORMATS AN OUTPUT LINE
;   140          2561  1         PAT$FREEZ,                                      ! Allocates free storage and zeroes it
;   141          2562  1         PAT$REDUCE_INS : NOVALUE,                       ! REDUCES EXPRESSIONS AND SYMBOLS INSIDE INS
;   142          2563  1         GETFILDSC;                                      ! Returns the address of a file name descrip
;   143          2564  1
;   144          2565  1 EXTERNAL
;   145          2566  1         PAT$GL_CHANUM,                                  ! Channel number
;   146          2567  1         PAT$GL_SYMTBPTR,                                ! Pointer to current symbol table
```

```
;  147     2568  1          PAT$GL_SYMHEAD,                                  ! Pointer to listhead entry of user-defined
;  148     2569  1          PAT$GL_PAL_LHD,                                  ! POINTER TO PATCH AREA LIST (PAL)
;  149     2570  1          PAT$GL_LAST_LOC,                                 ! LAST LOCATION EXAMINED
;  150     2571  1          PAT$GL_CONTEXT : BITVECTOR,                      ! CONTEXT BITS FOR COMMAND
;  151     2572  1          PAT$GL_SEMAN1 : VECTOR,                          ! PARSE STACK
;  152     2573  1          PAT$CP_OUT_STR,                                  ! POINTER TO OUTPUT STRING
;  153     2574  1          PAT$GL_BUF_SIZ,                                  ! SIZE OF TEXT IN OUTPUT BUFFER
;  154     2575  1          PAT$GB_MOD_PTR : REF VECTOR[,BYTE],
;  155     2576  1          PAT$GL_HEAD_LST,                                 ! POINTER TO ARGUMENT LIST FOR COMMAND
;  156     2577  1          PAT$GL_COMRAB,                                   ! RAB FOR COMMAND FILE
;  157     2578  1                                   ! POINTER TO MODE LEVEL
;  158     2579  1          PAT$GL_ISVADDR : VECTOR [,LONG],                 ! VIRTUAL ADDRESSES OF LAST IMAGE SECTION MA
;  159     2580  1          PAT$GL_EXPANDVA,                                 ! FIRST EXPREG ADDRESS
;  160     2581  1          PAT$GL_ERRCODE,                                  ! ERROR CODE
;  161     2582  1          PAT$GL_FLAGS,                                    ! CLI FLAGS
;  162     2583  1          PAT$GL_OLDNBK : BLOCK[,BYTE],                    ! INPUT IMAGE FILE FAB
;  163     2584  1          PAT$GL_NEWNBK : BLOCK[,BYTE],                    ! OUTPUT IMAGE FILE FAB
;  164     2585  1          PAT$GL_JNLNBK : BLOCK[,BYTE],                    ! JOURNAL FILE FAB
;  165     2586  1          PAT$GL_COMNBK : BLOCK[,BYTE],                    ! OUTPUT COMMAND FILE FAB
;  166     2587  1          PAT$GL_INPFAB : BLOCK[,BYTE],                    ! INPUT COMMAND CHANNEL FAB
;  167     2588  1          PAT$GL_OUTFAB : BLOCK[,BYTE],                    ! INFORMATION CHANNEL FAB
;  168     2589  1          PAT$GL_ERRFAB : BLOCK[,BYTE],                    ! ERROR CHANNEL FAB
;  169     2590  1          PAT$GL_OLDFAB : BLOCK[,BYTE],                    ! INPUT IMAGE FILE NBK
;  170     2591  1          PAT$GL_NEWFAB : BLOCK[,BYTE],                    ! OUTPUT IMAGE FILE NBK
;  171     2592  1          PAT$GL_JNLFAB : BLOCK[,BYTE],                    ! JOURNAL FILE NBK
;  172     2593  1          PAT$GL_COMFAB : BLOCK[,BYTE],                    ! OUTPUT COMMAND FILE NBK
;  173     2594  1          PAT$GB_OLDNAME,                                  ! OLD IMAGE FILE NAME
;  174     2595  1          PAT$GB_NEWNAME,                                  ! NEW IMAGE FILE NAME
;  175     2596  1          PAT$GB_JNLNAME,                                  ! JOURNAL FILE NAME
;  176     2597  1          PAT$GB_COMNAME,                                  ! OUTPUT COMMAND FILE NAME
;  177     2598  1          PAT$GL_TXTLHD : REF BLOCK[,BYTE],                ! Pointer to first text buffer
;  178     2599  1          PAT$GL_TXTTAIL : REF BLOCK[,BYTE],               ! Pointer to last text buffer
;  179     2600  1          PAT$GL_TXTFREE : REF VECTOR[,BYTE];              ! Pointer to next free byte in last text buf
;  180     2601  1
;  181     2602  1  EXTERNAL LITERAL
;  182     2603  1  !
;  183     2604  1  ! Define shared message references.  (resolved @ link time)
;  184     2605  1  !
;  185     2606  1          PAT$_CLOSEIN,                                    ! Error closing input file.
;  186     2607  1          PAT$_CLOSEOUT,                                   ! Error closing output file.
;  187     2608  1          PAT$_OPENIN,                                     ! Error opening input file.
;  188     2609  1          PAT$_OPENOUT,                                    ! Error opening output file.
;  189     2610  1          PAT$_READERR,                                    ! Error reading from file.
;  190     2611  1          PAT$_SYSERROR,                                   ! System Service error.
;  191     2612  1          PAT$_WRITEERR;                                   ! Error writing to file.
;  192     2613  1
;  193     2614  1  !++
;  194     2615  1  ! FAO CONTROL STRINGS
;  195     2616  1  !--
;  196     2617  1  BIND
;  197     2618  1          ASC_STG = UPLIT BYTE (%ASCIC '''!AD'''),         ! FOR ASCII MODE
;  198     2619  1          COM_HYP_STG = UPLIT BYTE (%ASCIC ',-'),          ! CONTINUATION FOR EXAMINE COMMANDS
;  199     2620  1          EXP_STG = UPLIT BYTE (%ASCIC '^X!XL'),           ! FOR NUMERICAL EXPRESSIONS
;  200     2621  1          RAN_STG = UPLIT BYTE (%ASCIC '^X!XL:^X!XL'),     ! FOR NUMERICAL EXPRESSIONS
;  201     2622  1          NAM_OFF_STG = UPLIT BYTE (%ASCIC '!AD+^X!XL'),   ! FOR SYMBOLIC NAME + OFFSET
;  202     2623  1          NAM_OFF_RAN_STG = UPLIT BYTE (%ASCIC '!AD+^X!XL:!AD+^X!XL'),    ! FOR SYMBOLIC NAME + OFFSET RANGES
;  203     2624  1          INS_STG = UPLIT BYTE (%ASCIC '''!AD'''),         ! FOR SYMBOLIC INSTRUCTIONS
```

```
;  204          2625  1          MOD_STG = UPLIT BYTE (%ASCIC '!AC'),                    ! FOR MODULE NAMES
;  205          2626  1          NAM_STG = UPLIT BYTE (%ASCIC '!AS'),                    ! FOR NAMES
;  206          2627  1
;  207          2628  1      !++
;  208          2629  1      ! COMMAND STRING FOR EXIT.
;  209          2630  1      !--
;  210          2631  1          EXIT_CMD = UPLIT BYTE (%ASCIC 'EXIT') : VECTOR[,BYTE];  ! EXIT COMMAND STRING
```

```
;  212     2632  1 GLOBAL ROUTINE PAT$WRITEFILE (MSGSIZ,MSGADR,WRTRAB) :NOVALUE =  ! WRITES FORMATTED MSG TO FILE
;  213     2633  1
;  214     2634  1 !++
;  215     2635  1 ! FUNCTIONAL DESCRIPTION:
;  216     2636  1 !
;  217     2637  1 !       This routine outputs formatted messages to files.  If the file is the
;  218     2638  1 !       output command file, then the output is also written into the appended
;  219     2639  1 !       patch command text buffers and only written to the command file if one
;  220     2640  1 !       is open for output.  In case of error, an appropriate message is SIGNALed.
;  221     2641  1 !
;  222     2642  1 ! FORMAL PARAMETERS:
;  223     2643  1 !
;  224     2644  1 !       MSGSIZ - SIZE IN BYTES OF THE MESSAGE TO BE OUTPUT
;  225     2645  1 !       MSGADR - ADDRESS OF THE MESSAGE TO BE OUTPUT
;  226     2646  1 !       WRTRAB - RAB OF THE FILE MESSAGE IS TO OUTPUT TO
;  227     2647  1 !
;  228     2648  1 ! IMPLICIT INPUTS:
;  229     2649  1 !
;  230     2650  1 !       MESSAGE IS ALREADY FORMATTED, FILE IS ALREADY OPEN, AND
;  231     2651  1 !       CHANNEL IS ALREADY CONNECTED.
;  232     2652  1 !       PAT$GL_TXTFREE - Pointer to next free byte in last text block
;  233     2653  1 !       PAT$GL_TXTLHD -  Pointer to first block of command text
;  234     2654  1 !       PAT$GL_TXTTAIL - Pointer to last block of command text
;  235     2655  1 !
;  236     2656  1 ! IMPLICIT OUTPUTS:
;  237     2657  1 !
;  238     2658  1 !       THE MESSAGE IS WRITTEN TO THE FILE.  IN CASE OF ERROR, AN
;  239     2659  1 !       ERROR MESSAGE IS WRITTEN TO THE JOURNAL FILE AND 'SYS$ERROR'.
;  240     2660  1 !
;  241     2661  1 ! ROUTINE VALUE:
;  242     2662  1 !
;  243     2663  1 ! COMPLETION CODES:
;  244     2664  1 !
;  245     2665  1 !       NONE
;  246     2666  1 !
;  247     2667  1 ! SIDE EFFECTS:
;  248     2668  1 !
;  249     2669  1 !       NONE
;  250     2670  1 !
;  251     2671  1 !--
;  252     2672  1
;  253     2673  2 BEGIN
;  254     2674  2
;  255     2675  2 MAP
;  256     2676  2         WRTRAB : REF BLOCK [,BYTE];                             ! DEFINE AS VECTOR OF BYTES
;  257     2677  2
;  258     2678  2 LOCAL
;  259     2679  2         FAB_ADDR : REF BLOCK[,BYTE],                            ! ADDRESS OF FAB
;  260     2680  2         NAM_BLK_ADDR : REF BLOCK[,BYTE];                        ! ADDRESS OF NAME BLOCK
;  261     2681  2
;  262     2682  2 !++
;  263     2683  2 ! First check if this write is going to the command file.  If so, add it
;  264     2684  2 ! to the text buffer.  Text buffers are allocated one block at a time.
;  265     2685  2 ! The first longword of the block points the next block of text, zero indicates
;  266     2686  2 ! no next block.  The commands are written into the buffer as ASCIC strings.
;  267     2687  2 ! They do NOT cross block boundaries.
;  268     2688  2 !--
```

```
;  269          2689  3 IF (.WRTRAB EQL PAT$GL_COMRAB)
;  270          2690  2 THEN
;  271          2691  3         BEGIN
;  272          2692  4         IF (.PAT$GL_TXTFREE+.MSGSIZ+1 GTRU .PAT$GL_TXTTAIL+A_PAGE)
;  273          2693  3         THEN
;  274          2694  4                 BEGIN
;  275          2695  4                 !++
;  276          2696  4                 ! Allocate a new block of command text and update pointers.
;  277          2697  4                 !--
;  278          2698  4                 PAT$GL_TXTFREE = PAT$FREEZ((A_PAGE + 3)/4);
;  279          2699  4                 PAT$GL_TXTTAIL[TXT$L_NXTBLK] = .PAT$GL_TXTFREE;
;  280          2700  4                 PAT$GL_TXTTAIL = .PAT$GL_TXTFREE;
;  281          2701  4                 PAT$GL_TXTFREE = .PAT$GL_TXTFREE + TXT$C_SIZE;
;  282          2702  3                 END;
;  283          2703  3         CH$MOVE(.MSGSIZ, .MSGADR, CH$PTR(.PAT$GL_TXTFREE, 1));
;  284          2704  3         PAT$GL_TXTFREE[0] = .MSGSIZ;
;  285          2705  3         PAT$GL_TXTFREE = CH$PTR(.PAT$GL_TXTFREE, .MSGSIZ+1);
;  286          2706  3         IF (.PAT$GL_FLAGS AND PAT$M_COMMAND) EQL 0
;  287          2707  3         THEN
;  288          2708  3                 RETURN;
;  289          2709  2         END;
;  290          2710  2
;  291          2711  2 !++
;  292          2712  2 ! SET UP THE RAB FOR OUTPUT.
;  293          2713  2 !--
;  294          2714  2 WRTRAB[RAB$W_RSZ]=.MSGSIZ;                                  ! SET SIZE OF MSG
;  295          2715  2 WRTRAB[RAB$L_RBF]=.MSGADR;                                  ! SET MESSAGE ADDRESS
;  296          2716  2 PAT$GL_ERRCODE = $PUT(RAB=.WRTRAB);                         ! OUTPUT MESSAGE
;  297          2717  2 IF NOT .PAT$GL_ERRCODE                                     ! IF ERROR,
;  298          2718  2 THEN                                                       ! OUTPUT AN
;  299          2719  3         BEGIN
;  300          2720  3         FAB_ADDR=.WRTRAB[RAB$L_FAB];                       ! GET ADDRESS OF FAB
;  301          2721  3         NAM_BLK_ADDR=.FAB_ADDR[FAB$L_NAM];                 ! GET ADDRESS OF NAME BLOCK
;  302          2722  3         SIGNAL(PAT$_WRITEERR,1,GETFI[DSC(.FAB_ADDR),.WRTRAB[RAB$L_STS],.WRTRAB[RAB$L_STV]);     ! APPROPRIAT
;  303          2723  2         END;
;  304          2724  1 END;                                                       ! END OF PAT$WRITEFILE


                                                                .TITLE  PATIO
                                                                .IDENT  \V04-000\

                                                                .PSECT  _PAT$PLIT,NOWRT,NOEXE,0

                                                00000 P.AAA:    .ASCII  <5>\'!AD'\
                                                00006 P.AAB:    .ASCII  <2>\,-\
                                                00009 P.AAC:    .ASCII  <5>\^X!XL\
                                                0000F P.AAD:    .ASCII  <11>\^X!XL:^X!XL\
                                                0001B P.AAE:    .ASCII  <9>\!AD+^X!XL\
                                                00025 P.AAF:    .ASCII  <19>\!AD+^X!XL:!AD+^X!XL\
                                                00034
                                                00039 P.AAG:    .ASCII  <5>\'!AD'\
                                                0003F P.AAH:    .ASCII  <3>\!AC\
                                                00043 P.AAI:    .ASCII  <3>\!AS\
                                                00047 P.AAJ:    .ASCII  <4>\EXIT\

                                                ISE$C_SIZE==        20
                                                TXT$C_SIZE==        4
```

```
;  PAL$C_SIZE==          16
;  ASD$C_SIZE==          9
;  FWR$C_SIZE==          24
;  ASC_STG=              P.AAA
;  COM_HYP_STG=          P.AAB
;  EXP_STG=              P.AAC
;  RAN_STG=              P.AAD
;  NAM_OFF_STG=          P.AAE
;  NAM_OFF_RAN_STG=      P.AAF
;  INS_STG=              P.AAG
;  MOD_STG=              P.AAH
;  NAM_STG=              P.AAI
;  EXIT_CMD=             P.AAJ
                .EXTRN  PAT$FIND_VAL, PAT$FAO_PUT
                .EXTRN  PAT$FREEZ, PAT$REDUCE_INS
                .EXTRN  GETFILDSC, PAT$GL_CHANUM
                .EXTRN  PAT$GL_SYMTBPTR
                .EXTRN  PAT$GL_SYMHEAD, PAT$GL_PAL_LHD
                .EXTRN  PAT$GL_LAST_LOC
                .EXTRN  PAT$GL_CONTEXT, PAT$GL_SEMAN1
                .EXTRN  PAT$CP_OUT_STR, PAT$GL_BUF_SIZ
                .EXTRN  PAT$GB_MOD_PTR, PAT$GL_HEAD_LST
                .EXTRN  PAT$GL_COMRAB, PAT$GL_ISVADDR
                .EXTRN  PAT$GL_EXPANDVA
                .EXTRN  PAT$GL_ERRCODE, PAT$GL_FLAGS
                .EXTRN  PAT$GL_OLDNBK, PAT$GL_NEWNBK
                .EXTRN  PAT$GL_JNLNBK, PAT$GL_COMNBK
                .EXTRN  PAT$GL_INPFAB, PAT$GL_OUTFAB
                .EXTRN  PAT$GL_ERRFAB, PAT$GL_OLDFAB
                .EXTRN  PAT$GL_NEWFAB, PAT$GL_JNLFAB
                .EXTRN  PAT$GL_COMFAB, PAT$GB_OLDNAME
                .EXTRN  PAT$GB_NEWNAME, PAT$GB_JNLNAME
                .EXTRN  PAT$GB_COMNAME, PAT$GL_TXTLHD
                .EXTRN  PAT$GL_TXTTAIL, PAT$GL_TXTFREE
                .EXTRN  PAT$_CLOSEIN, PAT$_CLOSEOUT
                .EXTRN  PAT$_OPENIN, PAT$_OPENOUT
                .EXTRN  PAT$_READERR, PAT$_SYSERROR
                .EXTRN  PAT$_WRITEERR, SYS$PUT
                .WEAK   ACCESS_CHECK

                .PSECT  _PAT$CODE,NOWRT,2

                07FC 00000      .ENTRY  PAT$WRITEFILE, Save R2,R3,R4,R5,R6,R7,R8,- ; 2632
                                        R9,R10
5A 00000000G    EF   9E 00002   MOVAB   PAT$GL_ERRCODE, R10                        ;
59 00000000G    EF   9E 00009   MOVAB   PAT$GL_TXTTAIL, R9                        ;
58 00000000G    EF   9E 00010   MOVAB   PAT$GL_TXTFREE, R8                        ;
57        0C    AC   D0 00017   MOVL    WRTRAB, R7                                ; 2689
50 00000000G    EF   9E 0001B   MOVAB   PAT$GL_COMRAB, R0                         ;
50              57   D1 00022   CMPL    R7, R0                                    ;
                4A   12 00025   BNEQ    2$                                        ;
50        68    04   AC   C1 00027   ADDL3   MSGSIZ, PAT$GL_TXTFREE, R0           ; 2692
                50   D6 0002C   INCL    R0                                        ;
51        69 00000200   8F   C1 0002E   ADDL3   #512, PAT$GL_TXTTAIL, R1          ;
51              50   D1 00036   CMPL    R0, R1                                    ;
                18   1B 00039   BLEQU   1$                                       ;
7E        80    8F   9A 0003B   MOVZBL  #128, -(SP)                               ; 2698
```

```
                    00000000G  EF              01  FB 0003F          CALLS   #1, PAT$FREEZ                                    ;
                               68              50  D0 00046          MOVL    R0, PAT$GL_TXTFREE                              ;
                         00    B9              68  D0 00049          MOVL    PAT$GL_TXTFREE, @PAT$GL_TXTTAIL                 ; 2699
                               69              68  D0 0004D          MOVL    PAT$GL_TXTFREE, PAT$GL_TXTTAIL                  ; 2700
                               68              04  C0 00050          ADDL2   #4, PAT$GL_TXTFREE                              ; 2701
                               56              68  D0 00053 1$:      MOVL    PAT$GL_TXTFREE, R6                              ; 2703
                01  A6     08  BC        04    AC  28 00056          MOVC3   MSGSIZ, @MSGADR, 1(R6)                          ;
                               66        04    AC  90 0005D          MOVB    MSGSIZ, (R6)                                    ; 2704
                    50         68        04    AC  C1 00061          ADDL3   MSGSIZ, PAT$GL_TXTFREE, R0                      ; 2705
                               68        01    A0  9E 00066          MOVAB   1(R0), PAT$GL_TXTFREE                           ;
                               3F 00000000G    EF  E9 0006A          BLBC    PAT$GL_FLAGS, 3$                                ; 2706
                         22    A7        04    AC  B0 00071 2$:      MOVW    MSGSIZ, 34(R7)                                  ; 2714
                         28    A7        08    AC  D0 00076          MOVL    MSGADR, 40(R7)                                  ; 2715
                               57              DD  0007B             PUSHL   R7                                              ; 2716
                    00000000G  00              01  FB 0007D          CALLS   #1, SYS$PUT                                     ;
                               6A              50  D0 00084          MOVL    R0, PAT$GL_ERRCODE                              ;
                               26              6A  E8 00087          BLBS    PAT$GL_ERRCODE, 3$                              ; 2717
                               50        3C    A7  D0 0008A          MOVL    60(R7), FAB_ADDR                                ; 2720
                               51        28    A0  D0 0008E          MOVL    40(FAB_ADDR), NAM_BLK_ADDR                      ; 2721
                               7E        08    A7  7D 00092          MOVQ    8(R7), -(SP)                                    ; 2722
                               50              DD  00096             PUSHL   FAB_ADDR                                        ;
                    00000000G  EF              01  FB 00098          CALLS   #1, GETFILDSC                                   ;
                               50              DD  0009F             PUSHL   R0                                              ;
                               01              DD  000A1             PUSHL   #1                                              ;
                                   00000000G   8F  DD 000A3          PUSHL   #PAT$_WRITEERR                                  ;
                    00000000G  00              05  FB 000A9          CALLS   #5, LIB$SIGNAL                                  ;
                                               04  000B0 3$:        RET                                                     ; 2724

; Routine Size:  177 bytes,    Routine Base:  _PAT$CODE + 0000
```

```
;  306          2725  1 GLOBAL ROUTINE PAT$WRITE_EXP1 (SEMSP) : NOVALUE =
;  307          2726  1
;  308          2727  1 !++
;  309          2728  1 ! FUNCTIONAL DESCRIPTION:
;  310          2729  1 !
;  311          2730  1 !       THIS ROUTINE WRITES OUT A SERIES OF EXPRESSIONS TO THE COMMAND FILE.
;  312          2731  1 !       THE EXPRESSIONS ARE TAKEN FROM THE COMMAND ARGUMENT LIST.  IF THE
;  313          2732  1 !       CURRENT MODE IS ASCII, THEN THE ARGUMENTS ARE WRITTEN AS ASCII
;  314          2733  1 !       CHARACTERS WITHIN QUOTES.  ONE EXPRESSION IS WRITTEN PER LINE.
;  315          2734  1 !
;  316          2735  1 !       IF THE COMMAND IS THE "EXAMINE" COMMAND, THEN THE EXPRESSIONS MUST
;  317          2736  1 !       BE FOLLOWED BY A CONTINUATION CHARACTER AS THE "EXAMINE" COMMAND
;  318          2737  1 !       DOES NOT PROMPT.
;  319          2738  1 !
;  320          2739  1 ! FORMAL PARAMETERS:
;  321          2740  1 !
;  322          2741  1 !       SEMSP - PARSE STACK OFFSET TO COMMAND VERB TOKEN
;  323          2742  1 !
;  324          2743  1 ! IMPLICIT INPUTS:
;  325          2744  1 !
;  326          2745  1 !       THE ARGUMENT LIST FOR THE COMMAND POINTED TO BY PAT$GL_HEAD_LST.
;  327          2746  1 !
;  328          2747  1 ! IMPLICIT OUTPUTS:
;  329          2748  1 !
;  330          2749  1 !       THE EXPRESSION IS WRITTEN TO THE COMMAND FILE.  IN CASE OF ERROR, AN
;  331          2750  1 !       ERROR MESSAGE IS WRITTEN TO THE JOURNAL FILE AND 'SYS$ERROR'.
;  332          2751  1 !
;  333          2752  1 ! ROUTINE VALUE:
;  334          2753  1 !
;  335          2754  1 !       NONE
;  336          2755  1 !
;  337          2756  1 ! COMPLETION CODES:
;  338          2757  1 !
;  339          2758  1 !       NONE
;  340          2759  1 !
;  341          2760  1 ! SIDE EFFECTS:
;  342          2761  1 !
;  343          2762  1 !       PAT$CP_OUT_STR AND PAT$GL_BUF_SIZ ARE DESTROYED.
;  344          2763  1 !
;  345          2764  1 !--
;  346          2765  1
;  347          2766  2 BEGIN
;  348          2767  2
;  349          2768  2 LITERAL
;  350          2769  2         HYPHEN = %X'2D';                                ! CONTINUATION CHARACTER (HYPHEN)
;  351          2770  2
;  352          2771  2 LOCAL
;  353          2772  2         OUTPUT_BUFFER : BLOCK[TTY_OUT_WIDTH,BYTE],      ! OUTPUT BUFFER
;  354          2773  2         POINTER;                                        ! POINTER TO CURRENT ARGUMENT
;  355          2774  2
;  356          2775  2 !++
;  357          2776  2 ! INITIALIZE THE POINTER TO THE FIRST ARGUMENT.
;  358          2777  2 !--
;  359          2778  2 POINTER = .PAT$GL_HEAD_LST;
;  360          2779  2 IF (.PAT$GL_SEMANT[.SEMSP] EQL EXAMINE_TOKEN) AND
;  361          2780  3    (.POINTER EQLA 0)
;  362          2781  2 THEN
```

```
;  363          2782  3          BEGIN
;  364          2783  3          PAT$GL_BUF_SIZ = 0;
;  365          2784  3          PAT$CP_OUT_STR = CH$PTR(OUTPUT_BUFFER, 0);
;  366          2785  3          PAT$OUT_PAL_EXP(.PAT$GL_LAST_LOC, 0);
;  367          2786  3          PAT$WRITEFILE(.PAT$GL_BUF_SIZ, OUTPUT_BUFFER, PAT$GL_COMRAB);
;  368          2787  2          END;
;  369          2788  2
;  370          2789  2      !++
;  371          2790  2      ! LOOP, WRITING OUT EACH EXPRESSION.  IF THE CURRENT MODE IS ASCII, WRITE
;  372          2791  2      ! THE EXPRESSION AS AN ASCII STRING CONTAINED WITHIN QUOTES.
;  373          2792  2      !--
;  374          2793  2      WHILE .POINTER NEQA 0
;  375          2794  2      DO
;  376          2795  3          BEGIN
;  377          2796  3          PAT$GL_BUF_SIZ = 0;
;  378          2797  3          PAT$CP_OUT_STR = CH$PTR(OUTPUT_BUFFER, 0);
;  379          2798  3          IF .PAT$GB_MOD_PTR[MODE_ASCII]
;  380          2799  3          THEN
;  381          2800  3                  PAT$FAO_PUT(ASC_STG, .PAT$GB_MOD_PTR[MODE_LENGTH],
;  382          2801  3                              LIST_ELEM_EXP1(.POINTER))
;  383          2802  3          ELSE
;  384          2803  3                  IF (.LIST_ELEM_EXP2(.POINTER) EQL 0) OR
;  385          2804  4                     (.LIST_ELEM_EXP1(.POINTER) EQL .LIST_ELEM_EXP2(.POINTER))
;  386          2805  3                  THEN
;  387          2806  3                          PAT$OUT_PAL_EXP(.LIST_ELEM_EXP1(.POINTER), 0)
;  388          2807  3                  ELSE
;  389          2808  3                          PAT$OUT_PAL_EXP(.LIST_ELEM_EXP1(.POINTER),
;  390          2809  3                                          .LIST_ELEM_EXP2(.POINTER));
;  391          2810  3          IF (.PAT$GL_SEMAN1[.SEMSP] EQL EXAMINE_TOKEN) AND
;  392          2811  4             (.LIST_ELEM_FLINK(.POINTER) NEQA 0)
;  393          2812  3          THEN
;  394          2813  3                  PAT$FAO_PUT(COM_HYP_STG);
;  395          2814  3          PAT$WRITEFILE(.PAT$GL_BUF_SIZ, OUTPUT_BUFFER, PAT$GL_COMRAB);
;  396          2815  3          POINTER = .LIST_ELEM_FLINK(.POINTER);
;  397          2816  2          END;
;  398          2817  2  RETURN;
;  399          2818  2
;  400          2819  1  END;                                                    ! END OF PAT$WRITE_EXP1


                                07FC 00000          .ENTRY  PAT$WRITE_EXP1, Save R2,R3,R4,R5,R6,R7,R8.- ; 2725
                                                            R9,R10                                      ;
5A 00000000G   EF   9E 00002        MOVAB   PAT$FAO_PUT, R10                                            ;
59      FF42   CF   9E 00009        MOVAB   PAT$WRITEFILE, R9                                           ;
58 00000000G   EF   9E 0000E        MOVAB   PAT$GL_COMRAB, R8                                           ;
57 00000000V   EF   9E 00015        MOVAB   PAT$OUT_PAL_EXP, R7                                         ;
56 00000000G   EF   9E 0001C        MOVAB   PAT$CP_OUT_STR, R6                                          ;
55 00000000G   EF   9E 00023        MOVAB   PAT$GL_SEMAN1, R5                                           ;
54 00000000G   EF   9E 0002A        MOVAB   PAT$GL_BUF_SIZ, R4                                          ;
5E      FF7C   CE   9E 00031        MOVAB   -132(SP), SP                                                ;
52 00000000G   EF   D0 00036        MOVL    PAT$GL_HEAD_LST, POINTER                                    ; 2778
53         04   AC   D0 0003D        MOVL    SEMSP, R3                                                   ; 2779
09       6543   D1 00041        CMPL    PAT$GL_SEMAN1[R3], #9                                       ;
         1E   12 00045        BNEQ    1$                                                          ;
```

```
                 52   D5 00047          TSTL    POINTER                                 ; 2780
                 1A   12 00049          BNEQ    1$
                 64   D4 0004B          CLRL    PAT$GL_BUF_SIZ                          ; 2783
66               6E   9E 0004D          MOVAB   OUTPUT_BUFFER, PAT$CP_OUT_STR           ; 2784
                 7E   D4 00050          CLRL    -(SP)                                   ; 2785
   00000000G     EF   DD 00052          PUSHL   PAT$GL_LAST_LOC
67               02   FB 00058          CALLS   #2, PAT$OUT_PAL_EXP
                 58   DD 0005B          PUSHL   R8                                      ; 2786
         04      AE   9F 0005D          PUSHAB  OUTPUT_BUFFER
                 64   DD 00060          PUSHL   PAT$GL_BUF_SIZ
69               03   FB 00062          CALLS   #3, PAT$WRITEFILE
                 52   D5 00065 1$:      TSTL    POINTER                                 ; 2793
                 5C   13 00067          BEQL    8$
                 64   D4 00069          CLRL    PAT$GL_BUF_SIZ                          ; 2796
66               6E   9E 0006B          MOVAB   OUTPUT_BUFFER, PAT$CP_OUT_STR           ; 2797
50 00000000G     EF   D0 0006E          MOVL    PAT$GB_MOD_PTR, R0                      ; 2798
12         04    A0   E9 00075          BLBC    4(R0), 2$
         04      A2   9F 00079          PUSHAB  4(POINTER)                              ; 2801
7E       01      A0   9A 0007C          MOVZBL  1(R0), -(SP)
   00000000'     EF   9F 00080          PUSHAB  ASC_STG                                 ; 2800
6A               03   FB 00086          CALLS   #3, PAT$FAO_PUT                         ; 2801
                 18   11 00089          BRB     6$
50         08    A2   D0 0008B 2$:      MOVL    8(POINTER), R0                          ; 2803
                 06   13 0008F          BEQL    3$
50         04    A2   D1 00091          CMPL    4(POINTER), R0                          ; 2804
                 04   12 00095          BNEQ    4$
                 7E   D4 00097 3$:      CLRL    -(SP)                                   ; 2806
                 02   11 00099          BRB     5$
                 50   DD 0009B 4$:      PUSHL   R0                                      ; 2809
         04      A2   DD 0009D 5$:      PUSHL   4(POINTER)                              ; 2808
67               02   FB 000A0          CALLS   #2, PAT$OUT_PAL_EXP
09             6543   D1 000A3 6$:      CMPL    PAT$GL_SEMAN1[R3], #9                   ; 2810
                 0D   12 000A7          BNEQ    7$
                 62   D5 000A9          TSTL    (POINTER)                               ; 2811
                 09   13 000AB          BEQL    7$
   00000000'     EF   9F 000AD          PUSHAB  COM_HYP_STG                             ; 2813
6A               01   FB 000B3          CALLS   #1, PAT$FAO_PUT
                 58   DD 000B6 7$:      PUSHL   R8                                      ; 2814
         04      AE   9F 000B8          PUSHAB  OUTPUT_BUFFER
                 64   DD 000BB          PUSHL   PAT$GL_BUF_SIZ
69               03   FB 000BD          CALLS   #3, PAT$WRITEFILE
52               62   D0 000C0          MOVL    (POINTER), POINTER                      ; 2815
                 A0   11 000C3          BRB     1$                                      ; 2793
                      04 000C5 8$:      RET                                             ; 2819

; Routine Size:  198 bytes,    Routine Base:  _PAT$CODE + 00B1
```

```
;  402     2820  1 GLOBAL ROUTINE PAT$WRITE_NAME (SEMSP) : NOVALUE =
;  403     2821  1
;  404     2822  1 !++
;  405     2823  1 ! FUNCTIONAL DESCRIPTION:
;  406     2824  1 !
;  407     2825  1 !       THIS ROUTINE WRITES OUT A SERIES OF NAMES TO THE COMMAND FILE.
;  408     2826  1 !       THE NAMES ARE TAKEN FROM THE COMMAND ARGUMENT LIST.  ONE NAME IS
;  409     2827  1 !       WRITTEN PER LINE.
;  410     2828  1 !
;  411     2829  1 ! FORMAL PARAMETERS:
;  412     2830  1 !
;  413     2831  1 !       SEMSP - OFFSET TO VERB TOKEN ON PARSE STACK
;  414     2832  1 !
;  415     2833  1 ! IMPLICIT INPUTS:
;  416     2834  1 !
;  417     2835  1 !       THE COMMAND ARGUMENT LIST POINTED TO BY PAT$GL_HEAD_LST.
;  418     2836  1 !
;  419     2837  1 ! IMPLICIT OUTPUTS:
;  420     2838  1 !
;  421     2839  1 !       THE NAME IS WRITTEN TO THE COMMAND FILE.  IN CASE OF ERROR, AN
;  422     2840  1 !       ERROR MESSAGE IS WRITTEN TO THE JOURNAL FILE AND 'SYS$ERROR'.
;  423     2841  1 !
;  424     2842  1 ! ROUTINE VALUE:
;  425     2843  1 !
;  426     2844  1 !       NONE
;  427     2845  1 !
;  428     2846  1 ! COMPLETION CODES:
;  429     2847  1 !
;  430     2848  1 !       NONE
;  431     2849  1 !
;  432     2850  1 ! SIDE EFFECTS:
;  433     2851  1 !
;  434     2852  1 !       PAT$CP_OUT_STR AND PAT$GL_BUF_SIZ ARE DESTROYED.
;  435     2853  1 !
;  436     2854  1 !--
;  437     2855  1
;  438     2856  2 BEGIN
;  439     2857  2
;  440     2858  2 LOCAL
;  441     2859  2         MC_PTR : REF MC_RECORD,                         ! POINTER TO MODULE CHAIN ENTRY
;  442     2860  2         OUTPUT_BUFFER : VECTOR[TTY_OUT_WIDTH,BYTE],     ! OUTPUT BUFFER
;  443     2861  2         POINTER;                                        ! POINTER TO CURRENT NAME
;  444     2862  2
;  445     2863  2 !++
;  446     2864  2 ! INITIALIZE THE POINTER TO THE FIRST ARGUMENT.
;  447     2865  2 !--
;  448     2866  2 POINTER = .PAT$GL_HEAD_LST;
;  449     2867  2
;  450     2868  2 !++
;  451     2869  2 ! LOOP, WRITING OUT EACH EXPRESSION.  IF THE CURRENT MODE IS ASCII, WRITE
;  452     2870  2 ! THE EXPRESSION AS AN ASCII STRING CONTAINED WITHIN QUOTES.
;  453     2871  2 !--
;  454     2872  2 WHILE .POINTER NEQA 0
;  455     2873  2 DO
;  456     2874  3         BEGIN
;  457     2875  3         PAT$GL_BUF_SIZ = 0;
;  458     2876  3         PAT$CP_OUT_STR = CH$PTR(OUTPUT_BUFFER, 0);
```

```
;  459     2877  3          IF .PAT$GL_CONTEXT[MODULE_BIT]
;  460     2878  3          THEN
;  461     2879  4                  BEGIN
;  462     2880  4                  MC_PTR = .LIST_ELEM_EXP1(.POINTER);
;  463     2881  4                  PAT$FAO_PUT(MOD_STG, MC_PTR[MC_NAME_CS]);
;  464     2882  4                  END
;  465     2883  3          ELSE
;  466     2884  3                  PAT$FAO_PUT(NAM_STG, .LIST_ELEM_EXP1(.POINTER));
;  467     2885  3          PAT$WRITEFILE(.PAT$GL_BUF_SIZ, OUTPUT_BUFFER, PAT$GL_COMRAB);
;  468     2886  3          IF .PAT$GL_SEMAN1[.SEMSP] EQL DEFINE_TOKEN
;  469     2887  3          THEN
;  470     2888  4                  BEGIN
;  471     2889  4                  PAT$GL_BUF_SIZ = 0;
;  472     2890  4                  PAT$CP_OUT_STR = CH$PTR(OUTPUT_BUFFER, 0);
;  473     2891  4                  PAT$OUT_PAC_EXP(.LIST_ELEM_EXP2(.POINTER), 0);
;  474     2892  4                  PAT$WRITEFILE(.PAT$GL_BUF_SIZ, OUTPUT_BUFFER, PAT$GL_COMRAB);
;  475     2893  3                  END;
;  476     2894  3          POINTER = .LIST_ELEM_FLINK(.POINTER);
;  477     2895  2          END;
;  478     2896  2 RETURN;
;  479     2897  2
;  480     2898  1 END;                                                    ! END OF PAT$WRITE_NAME


                                                        .EXTRN  PAT$GL_RST_BEGN

                                    01FC 00000          .ENTRY  PAT$WRITE_NAME, Save R2,R3,R4,R5,R6,R7,R8   ; 2820
                            58   FE83   CF   9E 00002           MOVAB   PAT$WRITEFILE, R8
                            57 00000000G EF   9E 00007          MOVAB   PAT$GL_COMRAB, R7
                            56 00000000G EF   9E 0000E          MOVAB   PAT$CP_OUT_STR, R6
                            55 00000000G EF   9E 00015          MOVAB   PAT$GL_BUF_SIZ, R5
                            5E   FF7C   CE   9E 0001C          MOVAB   -132(SP), SP
                            53 00000000G EF   D0 00021          MOVL    PAT$GL_HEAD_LST, POINTER            ; 2866
                            54     04   AC   D0 00028          MOVL    SEMSP, R4                           ; 2886
                                        53   D5 0002C 1$:      TSTL    POINTER                             ; 2872
                                        68   13 0002E          BEQL    5$
                                        65   D4 00030          CLRL    PAT$GL_BUF_SIZ                      ; 2875
                            66          6E   9E 00032          MOVAB   OUTPUT_BUFFER, PAT$CP_OUT_STR       ; 2876
                               00000000G EF   95 00035          TSTB    PAT$GL_CONTEXT                      ; 2877
                                        17   18 0003B          BGEQ    2$
                            52     04   A3   D0 0003D          MOVL    4(POINTER), MC_PTR                  ; 2880
                        50  52 00000000G EF   C1 00041          ADDL3   PAT$GL_RST_BEGN, MC_PTR, R0         ; 2881
                                  0C   A0   9F 00049          PUSHAB  12(R0)
                               00000000' EF   9F 0004C          PUSHAB  MOD_STG
                                        09   11 00052          BRB     3$
                                  04   A3   DD 00054 2$:      PUSHL   4(POINTER)                          ; 2884
                               000000(;' EF   9F 00057          PUSHAB  NAM_STG
                    00000000G  EF                02   FB 0005D 3$:      CALLS   #2, PAT$FAO_PUT
                                        57   DD 00064          PUSHL   R7                                  ; 2885
                                  04   AE   9F 00066          PUSHAB  OUTPUT_BUFFER
                                        65   DD 00069          PUSHL   PAT$GL_BUF_SIZ
                            68                  03   FB 0006B          CALLS   #3, PAT$WRITEFILE
                            05 00000000GEF      44   D1 0006E          CMPL    PAT$GL_SEMAN1[R4], #5       ; 2886
                                        1B   12 00076          BNEQ    4$
                                        65   D4 00078          CLRL    PAT$GL_BUF_SIZ                      ; 2889
                            66          6E   9E 0007A          MOVAB   OUTPUT_BUFFER, PAT$CP_OUT_STR       ; 2890
```

```
                                 7E  D4 0007D          CLRL    -(SP)                                   ; 2891
                              08 A3  DD 0007F          PUSHL   8(POINTER)                              ;
                00000000V     EF     02 FB 00082       CALLS   #2, PAT$OUT_PAL_EXP                     ;
                                     57 DD 00089       PUSHL   R7                                      ; 2892
                              04 AE  9F 0008B          PUSHAB  OUTPUT_BUFFER                           ;
                                     65 DD 0008E       PUSHL   PAT$GL_BUF_SIZ                          ;
                           68        03 FB 00090       CALLS   #3, PAT$WRITEFILE                       ;
                           53        63 D0 00093 4$:   MOVL    (POINTER), POINTER                      ; 2894
                                     94 11 00096       BRB     1$                                      ; 2872
                                        04 00098 5$:   RET                                             ; 2898

; Routine Size:  153 bytes,    Routine Base:  _PAT$CODE + 0177
```

```
;  482          2899  1 GLOBAL ROUTINE PAT$WRITE_INS (SEMSP) : NOVALUE =
;  483          2900  1
;  484          2901  1 !++
;  485          2902  1 ! FUNCTIONAL DESCRIPTION:
;  486          2903  1 !
;  487          2904  1 !       THIS ROUTINE WRITES OUT THE ARGUMENTS TO THE COMMAND FILE FOR THE
;  488          2905  1 !       FOLLOWING COMMAND VERBS, "DEPOSIT", "INSERT", "REPLACE", AND
;  489          2906  1 !       "VERIFY".  THE ARGUMENTS CONSIST OF AN ADDRESS, FOLLOWED BY ONE OR TWO
;  490          2907  1 !       LISTS OF INSTRUCTIONS (DEPENDING UPON THE COMMAND).
;  491          2908  1 !       THE INSTRUCTIONS ARE TAKEN FROM THE COMMAND ARGUMENT LIST.
;  492          2909  1 !       ONE INSTRUCTION IS WRITTEN PER LINE.
;  493          2910  1 !
;  494          2911  1 ! FORMAL PARAMETERS:
;  495          2912  1 !
;  496          2913  1 !       SEMSP - OFFSET ONTO PARSE STACK FOR COMMAND VERB TOKEN
;  497          2914  1 !
;  498          2915  1 ! IMPLICIT INPUTS:
;  499          2916  1 !
;  500          2917  1 !       THE COMMAND ARGUMENT LIST POINTED TO BY PAT$GL_HEAD_LST.
;  501          2918  1 !       THE PARSE STACK, PAT$GL_SEMAN1, HOLDS THE COMMAND VERB.
;  502          2919  1 !
;  503          2920  1 ! IMPLICIT OUTPUTS:
;  504          2921  1 !
;  505          2922  1 !       THE COMMAND ARGUMENTS ARE WRITTEN TO THE COMMAND FILE.  IN CASE OF
;  506          2923  1 !       ERROR, AN ERROR MESSAGE IS WRITTEN TO THE JOURNAL FILE AND 'SYS$ERROR'.
;  507          2924  1 !
;  508          2925  1 ! ROUTINE VALUE:
;  509          2926  1 !
;  510          2927  1 !       NONE
;  511          2928  1 !
;  512          2929  1 ! COMPLETION CODES:
;  513          2930  1 !
;  514          2931  1 !       NONE
;  515          2932  1 !
;  516          2933  1 ! SIDE EFFECTS:
;  517          2934  1 !
;  518          2935  1 !       PAT$CP_OUT_STR AND PAT$GL_BUF_SIZ ARE DESTROYED.
;  519          2936  1 !
;  520          2937  1 !--
;  521          2938  1
;  522          2939  2 BEGIN
;  523          2940  2
;  524          2941  2 LOCAL
;  525          2942  2         EDITED_INS_DESC : BLOCK[12,BYTE],                       ! DESCRIPTOR FOR EDITED INSTRUCTIONS
;  526          2943  2         EDITED_INS_BUF : VECTOR[TTY_OUT_WIDTH,BYTE],            ! BUFFER FOR EDITED INSTRUCTION
;  527          2944  2         OUTPUT_BUFFER : VECTOR[TTY_OUT_WIDTH,BYTE],             ! OUTPUT BUFFER
;  528          2945  2         POINTER;                                                ! POINTER TO CURRENT NAME
;  529          2946  2
;  530          2947  2 !++
;  531          2948  2 ! INITIALIZE THE POINTER TO THE FIRST ARGUMENT.
;  532          2949  2 !--
;  533          2950  2 POINTER = .PAT$GL_HEAD_LST;
;  534          2951  2
;  535          2952  2 !++
;  536          2953  2 ! FIRST WRITE OUT THE LOCATION ADDRESS.  THIS IS ALWAYS THE FIRST ARGUMENT.
;  537          2954  2 !--
;  538          2955  2 PAT$GL_BUF_SIZ = 0;
```

```
;  539          2956  2 PAT$CP_OUT_STR = CH$PTR(OUTPUT_BUFFER, 0);
;  540          2957  2 PAT$OUT_PAL_EXP(.LIST_ELEM_EXP1(.POINTER), 0);
;  541          2958  2 PAT$WRITEFILE(.PAT$GL_BUF_SIZ, OUTPUT_BUFFER, PAT$GL_COMRAB);
;  542          2959  2
;  543          2960  2 !++
;  544          2961  2 ! NOW LOOP, WRITING OUT EACH ARGUMENT.
;  545          2962  2 !--
;  546          2963  2 WHILE (POINTER = .LIST_ELEM_FLINK(.POINTER)) NEQA 0
;  547          2964  2 DO
;  548          2965  3         BEGIN
;  549          2966  3         PAT$GL_BUF_SIZ = 0;
;  550          2967  3         PAT$CP_OUT_STR = CH$PTR(OUTPUT_BUFFER, 0);
;  551          2968  3         IF .PAT$GB_MOD_PTR[MODE_INSTRUC]
;  552          2969  3         THEN
;  553          2970  4                 BEGIN
;  554          2971  4                 EDITED_INS_DESC[DSC$W_LENGTH] = 0;
;  555          2972  4                 EDITED_INS_DESC[DSC$A_POINTER] = EDITED_INS_BUF;
;  556          2973  4                 EDITED_INS_DESC[DSC$W_MAXLEN] = TTY_OUT_WIDTH;
;  557          2974  4                 PAT$REDUCE_INS(.LIST_ELEM_EXP1(.POINTER), EDITED_INS_DESC);
;  558          2975  4                 PAT$GL_BUF_SIZ = 0;
;  559          2976  4                 PAT$CP_OUT_STR = CH$PTR(OUTPUT_BUFFER, 0);
;  560          2977  4                 PAT$FAO_PUT(INS_STG, .EDITED_INS_DESC[DSC$W_LENGTH], EDITED_INS_BUF);
;  561          2978  4                 END
;  562          2979  3         ELSE
;  563          2980  3                 IF .PAT$GB_MOD_PTR[MODE_ASCII]
;  564          2981  3                 THEN
;  565          2982  3                         PAT$FAO_PUT(ASC_STG, .PAT$GB_MOD_PTR[MODE_LENGTH],
;  566          2983  3                                 LIST_ELEM_EXP1(.POINTER))
;  567          2984  3                 ELSE
;  568          2985  3                         PAT$OUT_PAL_EXP(.LIST_ELEM_EXP1(.POINTER), 0);
;  569          2986  3         PAT$WRITEFILE(.PAT$GL_BUF_SIZ, OUTPUT_BUFFER, PAT$GL_COMRAB);
;  570          2987  3         IF .PAT$GL_SEMAN1[.SEMSP] EQL REPLACE_TOKEN
;  571          2988  3         THEN
;  572          2989  3                 IF .LIST_ELEM_EXP2(.POINTER) EQL EXIT_TOKEN
;  573          2990  3                 THEN
;  574          2991  3                         PAT$WRITEFILE(.EXIT_CMD[0], EXIT_CMD[1], PAT$GL_COMRAB);
;  575          2992  2         END;
;  576          2993  2 RETURN;
;  577          2994  2
;  578          2995  1 END;                                                    ! END OF PAT$WRITE_INS


                                  03FC 00000          .ENTRY  PAT$WRITE_INS, Save R2,R3,R4,R5,R6,R7,R8,R9 ; 2899
              59 00000000V    EF   9E 00002          MOVAB   PAT$OUT_PAL_EXP, R9                         ;
              58      FDE3    CF   9E 00009          MOVAB   PAT$WRITEFILE, R8                           ;
              57 000000C0G    EF   9E 0000E          MOVAB   PAT$GL_COMRAB, R7                           ;
              56 00000000G    EF   9E 00015          MOVAB   PAT$CP_OUT_STR, R6                          ;
              55 00000000'    EF   9E 0001C          MOVAB   INS_STG, R5                                 ;
              54 00000000G    EF   9E 00023          MOVAB   PAT$GL_BUF_SIZ, R4                          ;
              5E      FEEC    CE   9E 0002A          MOVAB   -276(SP), SP                                ;
              52 00000000G    EF   D0 0002F          MOVL    PAT$GL_HEAD_LST, POINTER                    ; 2950
                              64   D4 00036          CLRL    PAT$GL_BUF_SIZ                              ; 2955
              66              6E   9E 00038          MOVAB   OUTPUT_BUFFER, PAT$CP_OUT_STR               ; 2956
                              7E   D4 0003B          CLRL    -(SP)                                       ; 2957
```

```
                        04   A2  DD 0003D          PUSHL   4(POINTER)                                                    ;
                    69       02  FB 00040          CALLS   #2, PAT$OUT_PAL_EXP                                          ;
                             57  DD 00043          PUSHL   R7                                                           ; 2958
                        04   AE  9F 00045          PUSHAB  OUTPUT_BUFFER                                                ;
                             64  DD 00048          PUSHL   PAT$GL_BUF_SIZ                                               ;
                    68       03  FB 0004A          CALLS   #3, PAT$WRITEFILE                                            ;
                    53   04  AC  D0 0004D          MOVL    SEMSP, R3                                                    ; 2987
                    52       62  D0 00051 1$:      MOVL    (POINTER), POINTER                                           ; 2963
                             01  12 00054          BNEQ    2$                                                           ;
                                 04 00056          RET                                                                  ;
                             64  D4 00057 2$:      CLRL    PAT$GL_BUF_SIZ                                               ; 2966
                    66       6E  9E 00059          MOVAB   OUTPUT_BUFFER, PAT$CP_OUT_STR                                ; 2967
                    50 00000000G EF D0 0005C       MOVL    PAT$GB_MOD_PTR, R0                                           ; 2968
                    2C   03  A0  E9 00063          BLBC    3(R0), 3$                                                    ;
                        F4   AD  B4 00067          CLRW    EDITED_INS_DESC                                              ; 2971
                F8  AD   0084   CE  9E 0006A       MOVAB   EDITED_INS_BUF, EDITED_INS_DESC+4                            ; 2972
                FC  AD     84   8F  9B 00070       MOVZBW  #132, EDITED_INS_DESC+8                                      ; 2973
                        F4   AD  9F 00075          PUSHAB  EDITED_INS_DESC                                              ; 2974
                        04   A2  DD 00078          PUSHL   4(POINTER)                                                   ;
        00000000G   EF           02  FB 0007B      CALLS   #2, PAT$REDUCE_INS                                           ;
                             64  D4 00082          CLRL    PAT$GL_BUF_SIZ                                               ; 2975
                    66       6E  9E 00084          MOVAB   OUTPUT_BUFFER, PAT$CP_OUT_STR                                ; 2976
                        0084 CE  9F 00087          PUSHAB  EDITED_INS_BUF                                               ; 2977
                    7E       F4  AD  3C 0008B      MOVZWL  EDITED_INS_DESC, -(SP)                                       ;
                             55  DD 0008F          PUSHL   R5                                                           ;
                             0E  11 00091          BRB     4$                                                           ;
                    13       04  A0  E9 00093 3$:  BLBC    4(R0), 5$                                                    ; 2980
                             04  A2  9F 00097      PUSHAB  4(POINTER)                                                   ; 2983
                    7E       01  A0  9A 0009A      MOVZBL  1(R0), -(SP)                                                 ;
                             C7  A5  9F 0009E      PUSHAB  ASC_STG                                                      ; 2982
        00000000G   EF           03  FB 000A1 4$:  CALLS   #3, PAT$FAO_PUT                                              ; 2983
                             08  11 000A6          BRB     6$                                                           ;
                             7E  D4 000AA 5$:      CLRL    -(SP)                                                        ; 2985
                        04   A2  DD 000AC          PUSHL   4(POINTER)                                                   ;
                    69       02  FB 000AF          CALLS   #2, PAT$OUT_PAL_EXP                                          ;
                             57  DD 000B2 6$:      PUSHL   R7                                                           ; 2986
                        04   AE  9F 000B4          PUSHAB  OUTPUT_BUFFER                                                ;
                             64  DD 000B7          PUSHL   PAT$GL_BUF_SIZ                                               ;
                    68       03  FB 000B9          CALLS   #3, PAT$WRITEFILE                                            ;
                    0D 00000000GEF43  D1 000BC     CMPL    PAT$GL_SEMAN1[R3], #13                                       ; 2987
                             8B  12 000C4          BNEQ    1$                                                           ;
                    0A       08  A2  D1 000C6      CMPL    8(POINTER), #10                                              ; 2989
                             85  12 000CA          BNEQ    1$                                                           ;
                             57  DD 000CC          PUSHL   R7                                                           ; 2991
                        0F   A5  9F 000CE          PUSHAB  EXIT_CMD+1                                                   ;
                    7E       0E  A5  9A 000D1      MOVZBL  EXIT_CMD, -(SP)                                              ;
                    68           03  FB 000D5      CALLS   #3, PAT$WRITEFILE                                            ;
                             FF76  31 000D8          BRW     1$                                                           ; 2963
                                   04 000DB          RET                                                                  ; 2995

; Routine Size:  220 bytes,    Routine Base:  _PAT$CODE + 0210
```

```
;  580          2996  1 GLOBAL ROUTINE PAT$OUT_PAL_EXP (EXPR1, EXPR2) : NOVALUE =
;  581          2997  1
;  582          2998  1 !++
;  583          2999  1 ! FUNCTIONAL DESCRIPTION:
;  584          3000  1 !
;  585          3001  1 !       THIS ROUTINE FORMATS EXPRESSIONS FOR OUTPUT TO THE COMMAND FILE.
;  586          3002  1 !       IF AN EXPRESSION FALLS WITHIN A PATCH AREA, THEN THE EXPRESSION
;  587          3003  1 !       MUST BE WRITTEN AS A SYMBOLIC NAME PLUS OFFSET.  OTHERWISE IT MAY
;  588          3004  1 !       BE WRITTEN AS AN ABSOLUTE VALUE.  THE PATCH AREA LIST (PAL) IS
;  589          3005  1 !       SEARCHED TO DETERMINE IF THE EXPRESSION FALLS WITHIN A PATCH AREA.
;  590          3006  1 !
;  591          3007  1 !       IF AN EXPRESSION RANGE IS PROVIDED AS INPUT (I.E., EXPR2 IS NOT 0),
;  592          3008  1 !       THEN THE UPPER LIMIT OF THE RANGE IS TREATED IN THE SAME MANNER AS
;  593          3009  1 !       THE LOWER LIMIT WAS TREATED.
;  594          3010  1 !
;  595          3011  1 ! FORMAL PARAMETERS:
;  596          3012  1 !
;  597          3013  1 !       EXPR1 - EXPRESSION TO OUTPUT (IF EXPR2 NEQ 0, THEN LOWER LIMIT OF RANGE)
;  598          3014  1 !       EXPR2 - IF NEQ 0, THEN LOWER LIMIT OF RANGE
;  599          3015  1 !               IF EQL 0, THEN INDICATES SINGLE EXPRESSION NOT RANGE
;  600          3016  1 !
;  601          3017  1 ! IMPLICIT INPUTS:
;  602          3018  1 !
;  603          3019  1 !       THE PATCH AREA LIST (PAL).
;  604          3020  1 !       THE USER DEFINED SYMBOL TABLE.
;  605          3021  1 !       PAT$GL_BUF_SIZ AND PAT$CP_OUT_STR MUST BE INITIALIZED.
;  606          3022  1 !
;  607          3023  1 ! IMPLICIT OUTPUTS:
;  608          3024  1 !
;  609          3025  1 !       THE EXPRESSION IS WRITTEN INTO THE OUTPUT BUFFER SPECIFIED BY
;  610          3026  1 !       PAT$GL_BUF_SIZ AND PAT$CP_OUT_STR.
;  611          3027  1 !
;  612          3028  1 ! ROUTINE VALUE:
;  613          3029  1 !
;  614          3030  1 !       NONE
;  615          3031  1 !
;  616          3032  1 ! COMPLETION CODES:
;  617          3033  1 !
;  618          3034  1 !       NONE
;  619          3035  1 !
;  620          3036  1 ! SIDE EFFECTS:
;  621          3037  1 !
;  622          3038  1 !       PAT$CP_OUT_STR AND PAT$GL_BUF_SIZ ARE UPDATED TO INCLUDE THE EXPRESSION.
;  623          3039  1 !
;  624          3040  1 !--
;  625          3041  1
;  626          3042  2 BEGIN
;  627          3043  2
;  628          3044  2 LOCAL
;  629          3045  2         TEMP_PTR : REF BLOCK[,BYTE],                    ! POINTER TO CURRENT PAL ENTRY
;  630          3046  2         SYM_PTR;                                        ! POINTER TO SYMBOL TABLE ENTRY
;  631          3047  2
;  632          3048  2 !++
;  633          3049  2 ! INITIALIZE THE POINTER TO THE FIRST PATCH AREA IN THE LIST.
;  634          3050  2 !--
;  635          3051  2 TEMP_PTR = .PAT$GL_PAL_LHD;
;  636          3052  2
```

```
;  637          3053  2 !++
;  638          3054  2 ! SEARCH THE PATCH AREA LIST FOR A PATCH AREA WHICH INCLUDES THE EXPRESSION.
;  639          3055  2 ! IF FOUND, THEN SEARCH THE SYMBOL TABLE FOR A CORRESPONDING SYMBOLIC NAME.
;  640          3056  2 ! IF THE EXPRESSION IS NOT WITHIN A PATCH AREA, THEN OUTPUT IT AS AN ABSOLUTE
;  641          3057  2 ! VALUE.  OTHERWISE, OUTPUT IT AS A SYMBOLIC NAME PLUS OFFSET.
;  642          3058  2 !--
;  643          3059  3 IF (.EXPR1 NEQ 0)                                          ! PREVENT NULL PATCH AREA ADDRESS REPLACING
;  644          3060  2 THEN
;  645          3061  3         BEGIN
;  646          3062  3         WHILE .TEMP_PTR NEQA 0
;  647          3063  3         DO
;  648          3064  4                 BEGIN
;  649          3065  4                 IF (.EXPR1 LEQU .TEMP_PTR[PAL$L_END_ADR]) AND
;  650          3066  5                    (.EXPR1 GEQU .TEMP_PTR[PAL$L_START_ADR])
;  651          3067  4                 THEN
;  652          3068  5                         BEGIN
;  653          3069  5                         PAT$GL_SYMTBPTR = .PAT$GL_SYMHEAD;
;  654          3070  5                         SYM_PTR = PAT$FIND_VAL(.EXPR1, FALSE);
;  655          3071  5                         IF .EXPR2 EQL 0
;  656          3072  5                         THEN
;  657          3073  5                                 PAT$FAO_PUT(NAM_OFF_STG,
;  658          3074  5                                         .SYM_CHCOUNT(.SYM_PTR), SYM_NAME(.SYM_PTR),
;  659          3075  5                                         (.EXPR1-.SYM_VALUE(.SYM_PTR)))
;  660          3076  5                         ELSE
;  661          3077  5                                 PAT$FAO_PUT(NAM_OFF_RAN_STG,
;  662          3078  5                                         .SYM_CHCOUNT(.SYM_PTR), SYM_NAME(.SYM_PTR),
;  663          3079  5                                         (.EXPR1-.SYM_VALUE(.SYM_PTR)),
;  664          3080  5                                         .SYM_CHCOUNT(.SYM_PTR), SYM_NAME(.SYM_PTR),
;  665          3081  5                                         (.EXPR2-.SYM_VALUE(.SYM_PTR)));
;  666          3082  5                         RETURN;
;  667          3083  4                         END;
;  668          3084  4                 TEMP_PTR = .TEMP_PTR[PAL$L_FLINK];
;  669          3085  3                 END;
;  670          3086  2         END;
;  671          3087  2
;  672          3088  2 !++
;  673          3089  2 ! EXPRESSION DID NOT FALL WITHIN A PATCH AREA.  THEREFORE, OUTPUT IT AS
;  674          3090  2 ! AN ABSOLUTE VALUE.
;  675          3091  2 !--
;  676          3092  2 IF .EXPR2 EQL 0
;  677          3093  2 THEN
;  678          3094  2         PAT$FAO_PUT(EXP_STG, .EXPR1)
;  679          3095  2 ELSE
;  680          3096  2         PAT$FAO_PUT(RAN_STG, .EXPR1, .EXPR2);
;  681          3097  2
;  682          3098  2 RETURN;
;  683          3099  2
;  684          3100  1 END;                                                      ! END OF PAT$OUT_PAL_EXP


                                00FC 00000          .ENTRY  PAT$OUT_PAL_EXP, Save R2,R3,R4,R5,R6,R7      ; 2996
                57 00000000G   EF   9E 00002        MOVAB   PAT$FAO_PUT, R7                              ;
                56 00000000'   EF   9E 00009        MOVAB   NAM_OFF_STG, R6                              ;
                54 00000000G   EF   D0 00010        MOVL    PAT$GL_PAL_LHD, TEMP_PTR                     ; 3051
```

```
                        55        04    AC  D0 00017          MOVL    EXPR1, R5                                       ; 3059
                                        66  13 0001B          BEQL    4$                                              ;
                                        54  D5 0001D 1$:      TSTL    TEMP_PTR                                        ; 3062
                                        62  13 0001F          BEQL    4$                                              ;
                              08    A4    55  D1 00021          CMPL    R5, 8(TEMP_PTR)                                 ; 3065
                                        57  1A 00025          BGTRU   3$                                              ;
                              04    A4    55  D1 00027          CMPL    R5, 4(TEMP_PTR)                                 ; 3066
                                        51  1F 0002B          BLSSU   3$                                              ;
                     00000000G  EF 00000000G  EF  D0 0002D          MOVL    PAT$GL_SYMHEAD, PAT$GL_SYMTBPTR         ; 3069
                                        7E  D4 00038          CLRL    -(SP)                                           ; 3070
                                        55  DD 0003A          PUSHL   R5                                              ;
                     00000000G  EF      02  FB 0003C          CALLS   #2, PAT$FIND_VAL                                ;
                                52      50  D0 00043          MOVL    R0, SYM_PTR                                     ;
                                53        0D  A2  9E 00046          MOVAB   13(R2), R3                              ; 3074
                                          08  AC  D5 0004A          TSTL    EXPR2                                   ; 3071
                                              11  12 0004D          BNEQ    2$                                      ;
                        7E        55        08  A2  C3 0004F          SUBL3   8(SYM_PTR), R5, -(SP)                   ; 3075
                                            53  DD 00054          PUSHL   R3                                      ; 3074
                                  7E        0C  A2  9A 00056          MOVZBL  12(SYM_PTR), -(SP)                      ;
                                            56  DD 0005A          PUSHL   R6                                      ; 3073
                                  67            04  FB 0005C          CALLS   #4, PAT$FAO_PUT                         ; 3074
                                                04  0005F          RET                                             ;
                        7E  08    AC        08  A2  C3 00060 2$:      SUBL3   8(SYM_PTR), EXPR2, -(SP)                ; 3081
                                            53  DD 00066          PUSHL   R3                                      ; 3080
                                  7E        0C  A2  9A 00068          MOVZBL  12(SYM_PTR), -(SP)                      ;
                        7E        55        08  A2  C3 0006C          SUBL3   8(SYM_PTR), R5, -(SP)                   ; 3079
                                            53  DD 00071          PUSHL   R3                                      ; 3080
                                  7E        0C  A2  9A 00073          MOVZBL  12(SYM_PTR), -(SP)                      ;
                                            0A  A6  9F 00077          PUSHAB  NAM_OFF_RAN_STG                         ; 3077
                                  67            07  FB 0007A          CALLS   #7, PAT$FAO_PUT                         ; 3080
                                                04  0007D          RET                                             ; 3068
                                  54        64  D0 0007E 3$:      MOVL    (TEMP_PTR), TEMP_PTR                    ; 3084
                                            9A  11 00081          BRB     1$                                      ; 3062
                                      08    AC  D5 00083 4$:      TSTL    EXPR2                                   ; 3092
                                            09  12 00086          BNEQ    5$                                      ;
                                            55  DD 00088          PUSHL   R5                                      ; 3094
                                      EE    A6  9F 0008A          PUSHAB  EXP_STG                                 ;
                                  67        02  FB 0008D          CALLS   #2, PAT$FAO_PUT                         ;
                                                04  00090          RET                                             ;
                                      08    AC  DD 00091 5$:      PUSHL   EXPR2                                   ; 3096
                                            55  DD 00094          PUSHL   R5                                      ;
                                      F4    A6  9F 00096          PUSHAB  RAN_STG                                 ;
                                  67        03  FB 00099          CALLS   #3, PAT$FAO_PUT                         ;
                                                04  0009C          RET                                             ; 3100

; Routine Size:  157 bytes,    Routine Base:  _PAT$CODE + 02EC
```

```
;  686     3101  1 GLOBAL ROUTINE PAT$CLOSEFILES : NOVALUE =       ! CLOSE ALL OPEN FILES
;  687     3102  1
;  688     3103  1 !++
;  689     3104  1 ! FUNCTIONAL DESCRIPTION:
;  690     3105  1 !
;  691     3106  1 !       THIS ROUTINE CLOSES ALL THE OPEN FILES BEING USED BY PATCH.
;  692     3107  1 !
;  693     3108  1 ! FORMAL PARAMETERS:
;  694     3109  1 !
;  695     3110  1 !       NONE
;  696     3111  1 !
;  697     3112  1 ! IMPLICIT INPUTS:
;  698     3113  1 !
;  699     3114  1 !       ALL FABS AND RABS MUST HAVE BEEN INITIALIZED.
;  700     3115  1 !
;  701     3116  1 ! IMPLICIT OUTPUTS:
;  702     3117  1 !
;  703     3118  1 !       NONE
;  704     3119  1 !
;  705     3120  1 ! ROUTINE VALUE:
;  706     3121  1 !
;  707     3122  1 ! COMPLETION CODES:
;  708     3123  1 !
;  709     3124  1 !       NONE
;  710     3125  1 !
;  711     3126  1 ! SIDE EFFECTS:
;  712     3127  1 !
;  713     3128  1 !       ALL FILES ARE CLOSED.
;  714     3129  1 !       IF ERRORS OCCUR, APPROPRIATE ERROR MESSAGES ARE OUTPUT TO THE
;  715     3130  1 !       JOURNAL FILE AND 'SYS$ERROR'.
;  716     3131  1 !
;  717     3132  1 !--
;  718     3133  1
;  719     3134  2 BEGIN
;  720     3135  2
;  721     3136  2 LITERAL
;  722     3137  2         START_OFF = 0,                                  ! START VIR ADDR OFFSET
;  723     3138  2         END_OFF = 1;                                    ! END VIR ADDR OFFSET
;  724     3139  2
;  725     3140  2 !++
;  726     3141  2 ! CLOSE OUTPUT COMMAND FILE, IF IT WAS OPEN.
;  727     3142  2 !--
;  728     3143  2 IF (.PAT$GL_FLAGS AND PAT$M_COMMAND) NEQ 0              ! IF /COMMAND WAS SPECIFIED
;  729     3144  2 THEN
;  730     3145  3         BEGIN
;  731     3146  3         PAT$GL_ERRCODE = $CLOSE(FAB=PAT$GL_COMFAB);     ! CLOSE COMMAND FILE
;  732     3147  3         IF NOT .PAT$GL_ERRCODE
;  733     3148  3         THEN
;  734     3149  3                 SIGNAL(PAT$_CLOSEOUT,1,GETFILDSC(PAT$GL_COMFAB),.PAT$GL_COMFAB[FAB$L_STS],.PAT$GL_COMFAB[FAB
;  735     3150  2         END;
;  736     3151  2
;  737     3152  2 !++
;  738     3153  2 ! CLOSE JOURNAL FILE, IF OPEN.
;  739     3154  2 !--
;  740     3155  2 IF (.PAT$GL_FLAGS AND PAT$M_JOURNAL) NEQ 0              ! IF JOURNAL IS OPEN
;  741     3156  2 THEN
;  742     3157  3         BEGIN
```

```
;  743          3158  3          PAT$GL_ERRCODE = $CLOSE(FAB=PAT$GL_JNLFAB);             ! CLOSE JOURNAL FILE
;  744          3159  3          IF NOT .PAT$GL_ERRCODE
;  745          3160  3          THEN
;  746          3161  3                  SIGNAL(PAT$_CLOSEOUT,1,GETFILDSC(PAT$GL_JNLFAB),.PAT$GL_JNLFAB[FAB$L_STS],.PAT$GL_JNLFAB[FAB
;  747          3162  2          END;
;  748          3163  2
;  749          3164  2  !++
;  750          3165  2  ! CLOSE INPUT IMAGE FILE, IF OPEN.
;  751          3166  2  !--
;  752          3167  2  IF (.PAT$GL_FLAGS AND PAT$M_INPUT) NEQ 0                       ! IF INPUT IMAGE FILE IS OPEN
;  753          3168  2  THEN
;  754          3169  3          BEGIN
;  755          3170  3          PAT$GL_ERRCODE = $DASSGN(CHAN=.PAT$GL_CHANUM);
;  756          3171  3          IF NOT .PAT$GL_ERRCODE
;  757          3172  3          THEN
;  758          3173  3                  SIGNAL(PAT$_CLOSEIN,1,GETFILDSC(PAT$GL_OLDFAB),.PAT$GL_ERRCODE,0);
;  759          3174  2          END;
;  760          3175  2
;  761          3176  2  !++
;  762          3177  2  ! CLOSE OUTPUT IMAGE FILE, IF OPEN.
;  763          3178  2  !--
;  764          3179  2  IF (.PAT$GL_FLAGS AND PAT$M_OUTPUT) NEQ 0                      ! IF OUTPUT IMAGE FILE IS OPEN
;  765          3180  2  THEN
;  766          3181  3          BEGIN
;  767          3182  3          PAT$GL_ERRCODE = $CLOSE(FAB=PAT$GL_NEWFAB);             ! CLOSE OUTPUT IMAGE FILE
;  768          3183  3          IF NOT .PAT$GL_ERRCODE
;  769          3184  3          THEN
;  770          3185  3                  SIGNAL(PAT$_CLOSEOUT,1,GETFILDSC(PAT$GL_NEWFAB),.PAT$GL_NEWFAB[FAB$L_STS],.PAT$GL_NEWFAB[FAB
;  771          3186  2          END;
;  772          3187  2
;  773          3188  2  !++
;  774          3189  2  ! CLOSE OTHER MESSAGE CHANNELS.
;  775          3190  2  --!
;  776          3191  2  $CLOSE(FAB=PAT$GL_INPFAB);                                    ! CLOSE INPUT COMMAND FILE
;  777          3192  2  $CLOSE(FAB=PAT$GL_OUTFAB);                                    ! CLOSE OUTPUT INFORMATION FILE
;  778          3193  2  $CLOSE(FAB=PAT$GL_ERRFAB);                                    ! CLOSE ERROR FILE
;  779          3194  2
;  780          3195  1  END;                                                          ! END OF PAT$CLOSEFILES


                                                        .EXTRN  SYS$CLOSE, SYS$DASSGN

                                        07FC 00000      .ENTRY  PAT$CLOSEFILES, Save R2,R3,R4,R5,R6,R7,R8,- ; 3101
                                                                R9,R10
                5A 00000000G   EF   9E 00002            MOVAB   PAT$GL_NEWFAB, R10                          ;
                59 00000000G   EF   9E 00009            MOVAB   PAT$GL_JNLFAB, R9                           ;
                58 00000000G   8F   D0 00010            MOVL    #PAT$_CLOSEOUT, R8                          ;
                57 00000000G   EF   9E 00017            MOVAB   PAT$GL_COMFAB, R7                           ;
                56 00000000G   EF   9E 0001E            MOVAB   PAT$GL_FLAGS, R6                            ;
                55 00000000G   00   9E 00025            MOVAB   LIB$SIGNAL, R5                              ;
                54 00000000G   EF   9E 0002C            MOVAB   GETFILDSC, R4                               ;
                53 00000000G   00   9E 00033            MOVAB   SYS$CLOSE, R3                               ;
                52 00000000G   EF   9E 0003A            MOVAB   PAT$GL_ERRCODE, R2                          ;
                1D                  66   E9 00041      BLBC    PAT$GL_FLAGS, 1$                            ; 3143
                                    57   DD 00044      PUSHL   R7                                          ; 3146
                63                  01   FB 00046      CALLS   #1, SYS$CLOSE                               ;
```

```
                62              50  D0 00049          MOVL    R0, PAT$GL_ERRCODE
                12              62  E8 0004C          BLBS    PAT$GL_ERRCODE, 1$                      ;3147
                7E        08    A7  7D 0004F          MOVQ    PAT$GL_COMFAB+8, -(SP)                  ;3149
                                  57  DD 00053          PUSHL   R7
                64                01  FB 00055          CALLS   #1, GETFILDSC
                                  50  DD 00058          PUSHL   R0
                                  01  DD 0005A          PUSHL   #1
                                  58  DD 0005C          PUSHL   R8
                65                05  FB 0005E          CALLS   #5, LIB$SIGNAL
          1D    66                01  E1 00061 1$:      BBC     #1, PAT$GL_FLAGS, 2$                    ;3155
                                  59  DD 00065          PUSHL   R9                                      ;3158
                63                01  FB 00067          CALLS   #1, SYS$CLOSE
                62                50  D0 0006A          MOVL    R0, PAT$GL_ERRCODE
                12                62  E8 0006D          BLBS    PAT$GL_ERRCODE, 2$                      ;3159
                7E        08    A9  7D 00070          MOVQ    PAT$GL_JNLFAB+8, -(SP)                  ;3161
                                  59  DD 00074          PUSHL   R9
                64                01  FB 00076          CALLS   #1, GETFILDSC
                                  50  DD 00079          PUSHL   R0
                                  01  DD 0007B          PUSHL   #1
                                  58  DD 0007D          PUSHL   R8
                65                05  FB 0007F          CALLS   #5, LIB$SIGNAL
          2D    66                02  E1 00082 2$:      BBC     #2, PAT$GL_FLAGS, 3$                    ;3167
                    00000000G     EF  DD 00086          PUSHL   PAT$GL_CHANUM                           ;3170
    00000000G   00                01  FB 0008C          CALLS   #1, SYS$DASSGN
                62                50  D0 00093          MOVL    R0, PAT$GL_ERRCODE
                1A                62  E8 00096          BLBS    PAT$GL_ERRCODE, 3$                      ;3171
                                  7E  D4 00099          CLRL    -(SP)                                   ;3173
                                  62  DD 0009B          PUSHL   PAT$GL_ERRCODE
                    00000000G     EF  9F 0009D          PUSHAB  PAT$GL_OLDFAB
                64                01  FB 000A3          CALLS   #1, GETFILDSC
                                  50  DD 000A6          PUSHL   R0
                                  01  DD 000A8          PUSHL   #1
                    00000000G     8F  DD 000AA          PUSHL   #PAT$_CLOSEIN
                65                05  FB 000B0          CALLS   #5, LIB$SIGNAL
          1D    66                03  E1 000B3 3$:      BBC     #3, PAT$GL_FLAGS, 4$                    ;3179
                                  5A  DD 000B7          PUSHL   R10                                     ;3182
                63                01  FB 000B9          CALLS   #1, SYS$CLOSE
                62                50  D0 000BC          MOVL    R0, PAT$GL_ERRCODE
                12                62  E8 000BF          BLBS    PAT$GL_ERRCODE, 4$                      ;3183
                7E        08    AA  7D 000C2          MOVQ    PAT$GL_NEWFAB+8, -(SP)                  ;3185
                                  5A  DD 000C6          PUSHL   R10
                64                01  FB 000C8          CALLS   #1, GETFILDSC
                                  50  DD 000CB          PUSHL   R0
                                  01  DD 000CD          PUSHL   #1
                                  58  DD 000CF          PUSHL   R8
                65                05  FB 000D1          CALLS   #5, LIB$SIGNAL
                    00000000G     EF  9F 000D4 4$:      PUSHAB  PAT$GL_INPFAB                           ;3191
                63                01  FB 000DA          CALLS   #1, SYS$CLOSE
                    00000000G     EF  9F 000DD          PUSHAB  PAT$GL_OUTFAB                           ;3192
                63                01  FB 000E3          CALLS   #1, SYS$CLOSE
                    00000000G     EF  9F 000E6          PUSHAB  PAT$GL_ERRFAB                           ;3193
                63                01  FB 000EC          CALLS   #1, SYS$CLOSE
                                      04 000EF          RET                                             ;3195

; Routine Size:  240 bytes,    Routine Base:  _PAT$CODE + 0389
```

```
;  782          3196  1 END
;  783          3197  0 ELUDOM                                ! END OF PATIO


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  _PAT$PLIT                      76  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(0)
;  _PAT$CODE                    1145  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                       0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                             Library Statistics

;                                      -------- Symbols --------      Pages       Processing
;       File                           Total    Loaded   Percent      Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1       18619        21         0      1000          00:01.9



;                                       COMMAND QUALIFIERS

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

; Size:          1145 code + 76 data bytes
; Run Time:         00:36.8
; Elapsed Time:     02:07.7
; Lines/CPU Min:     5219
; Lexemes/CPU-Min: 38275
; Memory Used:  225 pages
; Compilation Complete
```

PATPAR
LIS

PATMAC
LIS

PATMAI
LIS

PATLST
LIS

PATIO
LIS

PATLEX
LIS

PATMOD
LIS

PATMSG
LIS