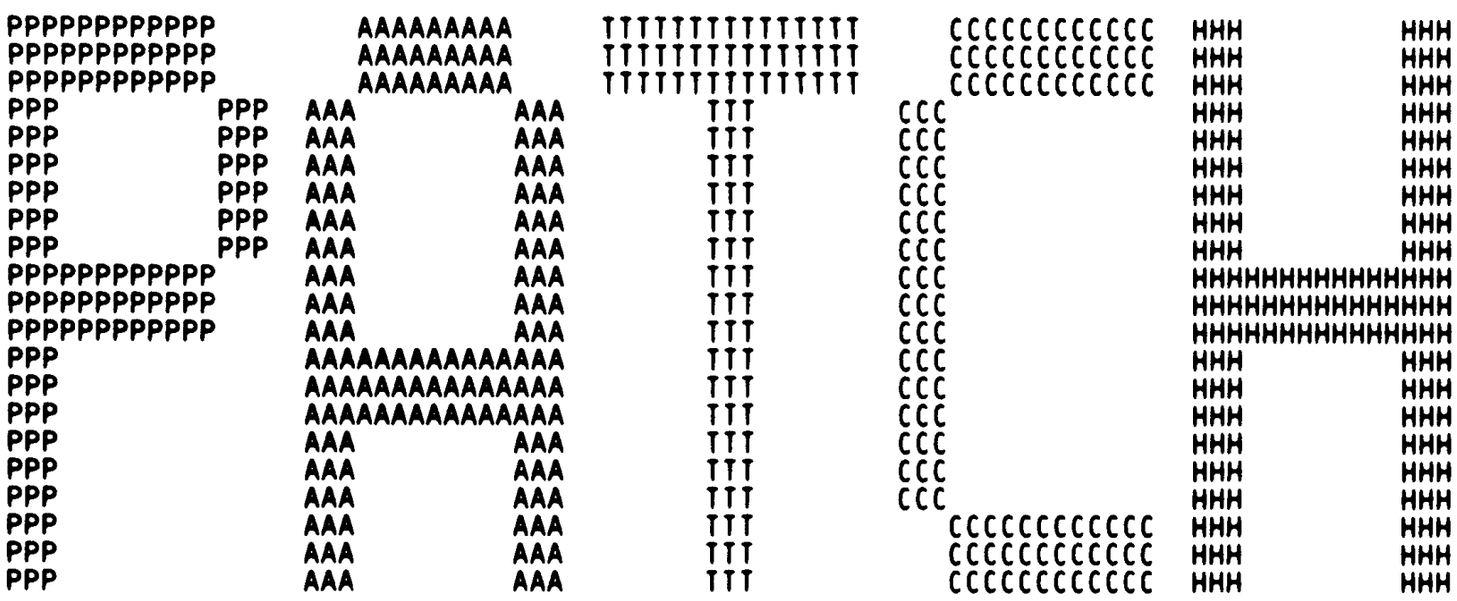
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPPPPPPPPPPP AAAAAAAAAAAAAAAA TTT CCC HHHHHHHHHHHHHHHH
PPP AAA AAA TTT CCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCCC HHH HHH

```
PPPPPPPP      AAAAAA    TTTTTTTTTT  WW      WW  RRRRRRRR    TTTTTTTTTT
PPPPPPPP      AAAAAA    TTTTTTTTTT  WW      WW  RRRRRRRR    TTTTTTTTTT
PP      PP   AA      AA     TT      WW      WW  RR      RR      TT
PP      PP   AA      AA     TT      WW      WW  RR      RR      TT
PP      PP   AA      AA     TT      WW      WW  RR      RR      TT
PP      PP   AA      AA     TT      WW      WW  RR      RR      TT
PPPPPPPP     AA      AA     TT      WW      WW  RRRRRRRR      TT
PPPPPPPP     AA      AA     TT      WW      WW  RRRRRRRR      TT
PP           AAAAAAAAAA     TT      WW  WW  WW  RR  RR          TT
PP           AAAAAAAAAA     TT      WW  WW  WW  RR  RR          TT
PP           AA      AA     TT      WWWW  WWWW  RR    RR        TT
PP           AA      AA     TT      WWWW  WWWW  RR    RR        TT
PP           AA      AA     TT      WW      WW  RR      RR      TT
PP           AA      AA     TT      WW      WW  RR      RR      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
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

```
;     1        L 0001  0 MODULE PATWRT (%IF %VARIANT EQL 1
;     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:     THIS MODULE CONTAINS THE ROUTINE TO WRITE THE PATCHED IMAGE FILE.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !
;    41          0041  1 ! AUTHOR: K.D. MORSE     , CREATION DATE: 3-NOV-77
;    42          0042  1 !
;    43          0043  1 ! MODIFIED BY:
;    44          0044  1 !
;    45          0045  1 !       V03-006 MCN0166         Maria del C. Nasr       23-Apr-1984
;    46          0046  1 !               Put the missing '.' in BYTES_TO_READ to load the buffer size
;    47          0047  1 !               in NEWRAB, so that we use the size and not the address.
;    48          0048  1 !
;    49          0049  1 !       V03-005 MCN0157         Maria del C. Nasr       20-Feb-1984
;    50          0050  1 !               Replace $READ to input file to $QIO since file is user opened.
;    51          0051  1 !               Add routine GET_IMAGE_BLOCK to do this.
;    52          0052  1 !
;    53          0053  1 !       V03-004 MTR0025         Mike Rhodes             8-Aug-1983
;    54          0054  1 !               Add routine WRITE_BINARY to support patching files in
;    55          0055  1 !               absolute mode.
;    56          0056  1 !
;    57          0057  1 !       V03-003 MTR0019         Mike Rhodes             5-Jan-1983
```

```
;    58          0058  1 !
;    59          0059  1 !               Add code to place the DMT (Debug Module/psect Table) between
;    60          0060  1 !               the DST and GST respectively in the image header in routine
;    61          0061  1 !               PAT$WRTIMG.
;    62          0062  1 !
;    63          0063  1 !       V03-002 MTR0014         Mike Rhodes             9-Sep-1982
;    64          0064  1 !               Correct the computation of the address of the next header block
;    65          0065  1 !               when the number of header blocks change.  Also, added code to
;    66          0066  1 !               preserve the last word of the image header (VBN 1) which is the
;    67          0067  1 !               image type identifier.
;    68          0068  1 !
;    69          0069  1 !       V03-001 MTR0007         Mike Rhodes             14-Jun-1982
;    70          0070  1 !               Use shared system messages.  Affected modules include:
;    71          0071  1 !               DYNMEM.B32, PATBAS.B32, PATCMD.B32, PATIHD.B32, PATINT.B32,
;    72          0072  1 !               PATIO.B32, PATMAI.B32, PATMSG.MSG, PATWRT.B32, and PATSPA.B32.
;    73          0073  1 !
;    74          0074  1 !               The shared messages are defined by DYNMEM.B32's invocation of
;    75          0075  1 !               SHRMSG.REQ and we simply link against these symbols.  They are
;    76          0076  1 !               declared as external literals below.
;    77          0077  1 !
;    78          0078  1 !       V0207   PCG0001         Peter George            02-FEB-1981
;    79          0079  1 !               Add require statement for LIB$:PATDEF.REQ
;    80          0080  1 !
;    81          0081  1 !       V0206   CNH0017         Chris Hume              10-Oct-1979     12:00
;    82          0082  1 !               Added OFP bit to PAT$GL_COMFAB.  Removed support for /COMMAND.
;    83          0083  1 !               Added support for /VOLUME.  (PATSTO.B32 01.17, PATPRE.REQ 01.03,
;    84          0084  1 !               PATMAI.B32 02.27, [VMSLIB]QUADEF.MAR 01.20)
;    85          0085  1 !
;    86          0086  1 !       V0205   KDM0008         KATHLEEN D. MORSE       16-OCT-1978     13:13
;    87          0087  1 !               Output updating message to SYS$OUTPUT. (V0205)
;    88          0088  1 !
;    89          0089  1 !       V0204   KDM0002         KATHLEEN D. MORSE       25-AUG-1978     13:13
;    90          0090  1 !               Check for global sections with no local copy in the
;    91          0091  1 !               image. (V0204)
;    92          0092  1 !
;    93          0093  1 ! MODIFICATIONS:
;    94          0094  1 !
;    95          0095  1 !  NO   DATE            PROGRAMMER              PURPOSE
;    96          0096  1 !  --   ----            ----------              -------
;    97          0097  1 !
;    98          0098  1 !  01   25-APR-78       K.D. MORSE              CONVERT TO NATIVE COMPILER.
;    99          0099  1 !  02   13-JUN-78       K.D. MORSE              ALLOW NON-CONTIGUOUS IMAGE FILES
;   100          0100  1 !                                              BUT TRY TO PRESERVE INPUT ATRIBS.
;   101          0101  1 !  03   13-JUN-78       K.D. MORSE              ADD FAO COUNTS TO SIGNALS.
;   102          0102  1 !
;   103          0103  1 !--
```

```
;  105          0104  1 !
;  106          0105  1 ! TABLE OF CONTENTS:
;  107          0106  1 !
;  108          0107  1
;  109          0108  1 FORWARD ROUTINE
;  110          0109  1         PAT$WRTIMG : NOVALUE,                   ! Writes out patched image
;  111          0110  1         GET_IMAGE_BLOCK,
;  112          0111  1         WRITE_BINARY : NOVALUE;                 ! Writes patched file.
;  113          0112  1
;  114          0113  1 !
;  115          0114  1 ! INCLUDE FILES:
;  116          0115  1 !
;  117          0116  1
;  118          0117  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  119          0118  1 REQUIRE 'SRC$:PATPCT.REQ';
;  120          0158  1 REQUIRE 'SRC$:PREFIX.REQ';
;  121          0346  1 REQUIRE 'SRC$:PATPRE.REQ';
;  122          0509  1 REQUIRE 'LIB$:PATDEF.REQ';                      ! Defines literals
;  123          0563  1 REQUIRE 'LIB$:PATMSG.REQ';
;  124          0737  1 REQUIRE 'SRC$:VXSMAC.REQ';
;  125          0802  1 REQUIRE 'SRC$:PATGEN.REQ';
;  126          1024  1 !
;  127          1025  1 ! MACROS:
;  128          1026  1 !
;  129          1027  1
;  130          1028  1 !
;  131          1029  1 ! EQUATED SYMBOLS:
;  132          1030  1 !
;  133          1031  1
;  134          1032  1 !
;  135          1033  1 ! OWN STORAGE:
;  136          1034  1 !
;  137          1035  1 OWN
;  138          1036  1         NUM_OF_UPDATES;                         ! Open vs create indicator
;  139          1037  1
;  140          1038  1 !
;  141          1039  1 ! EXTERNAL REFERENCES:
;  142          1040  1 !
;  143          1041  1
;  144          1042  1 EXTERNAL ROUTINE
;  145          1043  1         PAT$CREMAP : NOVALUE,                   ! Maps image sections
;  146          1044  1         PAT$WRITEFILE,                          ! Writes message to a file
;  147          1045  1         PAT$FAO_PUT,                            ! Formats message
;  148          1046  1         PAT$ALLOBLK : NOVALUE,                  ! Routine to allocate storage
;  149          1047  1         GETFILDSC,                              ! Returns the address of a file name descriptor
;  150          1048  1         IMG$DECODE_IHD : ADDRESSING_MODE(GENERAL),      ! Decode image header desc
;  151          1049  1         IMG$GET_NEXT_ISD : ADDRESSING_MODE(GENERAL);    ! Read next image header desc
;  152          1050  1
;  153          1051  1 EXTERNAL
;  154          1052  1         PAT$GB_ECOLVL : BYTE,                   ! Eco level for current patch
;  155          1053  1         PAT$GL_CHANUM,                          ! Channel number of input file
;  156          1054  1         PAT$GL_JNLRAB,                          ! RAB for journal file
;  157          1055  1         PAT$GL_NEWVBNMX,                        ! Max image section binary VBN in new file
;  158          1056  1         PAT$GL_OLDVBNMX,                        ! Max image section binary VBN in old file
;  159          1057  1         PAT$GL_IMGBLKS,                         ! Number of blocks in new image
;  160          1058  1         PAT$GW_IMGVOL : WORD,                   ! Relative Volume Number for new image
;  161          1059  1         PAT$GL_NEWXABALL : BLOCK [,BYTE],       ! New image file ALLocation XAB
```

```
;  162          1060  1          PAT$GL_NEWRAB : BLOCK[,BYTE],                    ! New image file RAB
;  163          1061  1          PAT$GL_NEWFAB : BLOCK[,BYTE],                    ! New image file FAB
;  164          1062  1          PAT$GL_NEWNBK : BLOCK[,BYTE],                    ! New image file name block
;  165          1063  1          PAT$GB_NEWNAME,                                      ! NAME OF NEW IMAGE FILE
;  166          1064  1          PAT$GL_OLDRAB : BLOCK[,BYTE],                        ! OLD IMAGE FILE RAB
;  167          1065  1          PAT$GL_OLDFAB : BLOCK[,BYTE],                        ! OLD IMAGE FILE FAB
;  168          1066  1          PAT$GL_OLDNBK : BLOCK[,BYTE],                        ! OLD IMAGE FILE NAME BLOCK
;  169          1067  1          PAT$GB_OLDNAME,                                      ! NAME OF OLD IMAGE FILE
;  170          1068  1          PAT$GB_INPBUF,
;  171          1069  1          PAT$GL_IMGHDR : REF BLOCK[,BYTE],                ! FIXED PART OF IMAGE HEADER
;  172          1070  1          PAT$GL_IHPPTR : REF BLOCK[,BYTE],                ! POINTER TO PATCH SECTION OF OLD IMAGE HEAD
;  173          1071  1          PAT$GL_ISELHD,                                   ! LIST HEAD OF IMAGE SECTION TABLE
;  174          1072  1          PAT$GL_ISETAIL,                                  ! TAIL OF THE IMAGE SECTION TABLE
;  175          1073  1          PAT$GL_TXTLHD : REF VECTOR[,BYTE],               ! LIST HEAD FOR COMMAND TEXT
;  176          1074  1          PAT$GL_FLAGS : BITVECTOR [32],                   ! PATCH FILE FLAGS
;  177          1075  1          PAT$GL_ERRCODE,                                  ! GLOBAL ERROR CODE
;  178          1076  1          PAT$GW_IMGTYP : WORD;                            ! IMAGE TYPE IDENTIFIER
;  179          1077  1
;  180          1078  1  EXTERNAL LITERAL
;  181          1079  1  !
;  182          1080  1  ! Define shared message references.  (resolved @ link time)
;  183          1081  1  !
;  184          1082  1          PAT$_CLOSEIN,                                    ! Error closing input file.
;  185          1083  1          PAT$_CLOSEOUT,                                   ! Error closing output file.
;  186          1084  1          PAT$_OPENIN,                                     ! Error opening input file.
;  187          1085  1          PAT$_OPENOUT,                                    ! Error opening output file.
;  188          1086  1          PAT$_OVERLAY,                                    ! !AS being overwritten. (/ABS/NONEW)
;  189          1087  1          PAT$_READERR,                                    ! Error reading from file.
;  190          1088  1          PAT$_SYSERROR,                                   ! System Service error.
;  191          1089  1          PAT$_WRITEERR;                                   ! Error writing to file.
;  192          1090  1
```

```
;  194          1091  1 GLOBAL ROUTINE PAT$WRTIMG : NOVALUE =                         ! WRITES OUT NEW IMAGE FILE
;  195          1092  1
;  196          1093  1 !++
;  197          1094  1 ! FUNCTIONAL DESCRIPTION:
;  198          1095  1 !
;  199          1096  1 !       THIS ROUTINE WRITES OUT THE NEW IMAGE FILE.
;  200          1097  1 !       IT PERFORMS THE FOLLOWING STEPS:
;  201          1098  1 !               1) Copies the fixed portion of the image header
;  202          1099  1 !               2) Moves in the image section descriptors
;  203          1100  1 !               3) Updates the header if necessary
;  204          1101  1 !               4) WRITES THE IMAGE HEADER
;  205          1102  1 !               5) WRITES OUT IMAGE BINARY, READING IT FROM OLD FILE IF NECESSARY
;  206          1103  1 !               6) WRITES OUT THE SYMBOL TABLE
;  207          1104  1 !               7) WRITES OUT THE APPENDED PATCH COMMANDS
;  208          1105  1 !               8) APPENDS THE PATCH COMMANDS FOR THIS PATCH SESSION
;  209          1106  1 !
;  210          1107  1 !       FIRST IT BUILDS THE IMAGE HEADER.  AT PRESENT THE IMAGE HEADER MUST FIT
;  211          1108  1 !       WITHIN THE ORIGINAL SIZE PLUS TWO BLOCKS AS THIS IS THE SIZE OF THE BUFFER.
;  212          1109  1 !       THIS IS ACCOMPLISHED BY MOVING IN THE FIXED PART OF THE HEADER AND THEN
;  213          1110  1 !       THE IMAGE SECTION DESCRIPTORS FROM THE IMAGE SECTION TABLE.
;  214          1111  1 !       IMAGE SECTION DESCRIPTORS DO NOT CROSS BLOCK BOUNDARIES AND SO
;  215          1112  1 !       THE REMAINING BYTES IN A BLOCK ARE FILLED WITH NEGATIVE ONES.
;  216          1113  1 !       AFTER THE HEADER IS COMPLETELY BUILT, THE NUMBER OF HEADER BLOCKS IS
;  217          1114  1 !       SET AND THE IMAGE SECTION DESCRIPTORS ARE UPDATED IF NECESSARY.
;  218          1115  1 !       NOW THE HEADER IS WRITTEN.
;  219          1116  1 !
;  220          1117  1 !       NEXT A SECOND PASS IS MADE THROUGH THE IMAGE SECTION TABLE TO WRITE
;  221          1118  1 !       OUT THE IMAGE BINARY.  THE IMAGE SECTION ENTRIES WITH ZEROS FOR
;  222          1119  1 !       STARTING AND ENDING MAPPED ADDRESSES MUST FIRST BE MAPPED IN FROM THE
;  223          1120  1 !       OLD IMAGE FILE AS THEY WERE NOT MAPPED IN DURING THE PATCH SESSION.
;  224          1121  1 !
;  225          1122  1 !       NOW THE SYMBOL TABLE IS WRITTEN INTO THE IMAGE FILE.  THIS SECTION
;  226          1123  1 !       IS VARIABLE LENGTH RECORDS, SO THE FILE ATTRIBUTES MUST BE CHANGED.
;  227          1124  1 !
;  228          1125  1 !       LASTLY THE PATCH COMMAND INFORMATION, ALSO VARIABLE LENGTH RECORDS IS
;  229          1126  1 !       IS COPIED FROM THE OLD IMAGE FILE.  THE PATCH COMMAND TEXT FROM THIS
;  230          1127  1 !       PATCH SESSION IS APPENDED TO THE IMAGE FILE.
;  231          1128  1 !
;  232          1129  1 ! FORMAL PARAMETERS:
;  233          1130  1 !
;  234          1131  1 !       NONE
;  235          1132  1 !
;  236          1133  1 ! IMPLICIT INPUTS:
;  237          1134  1 !
;  238          1135  1 !       THE IMAGE SECTION TABLE AND IMAGE HEADER MUST BE SET UP.
;  239          1136  1 !       APPENDED COMMAND TEXT BLOCKS MUST BE SET UP.
;  240          1137  1 !       SYMBOL TABLE IS IN MEMORY.
;  241          1138  1 !
;  242          1139  1 ! IMPLICIT OUTPUTS:
;  243          1140  1 !
;  244          1141  1 !       NONE
;  245          1142  1 !
;  246          1143  1 ! ROUTINE VALUE:
;  247          1144  1 !
;  248          1145  1 !       NONE
;  249          1146  1 !
;  250          1147  1 ! COMPLETION CODES:
```

```
;  251          1148  1 !
;  252          1149  1 !       NONE
;  253          1150  1 !
;  254          1151  1 ! SIDE EFFECTS:
;  255          1152  1 !
;  256          1153  1 !       THE PATCHED IMAGE FILE IS OUTPUT.
;  257          1154  1 !
;  258          1155  1 !--
;  259          1156  1
;  260          1157  2 BEGIN
;  261          1158  2
;  262          1159  2 LITERAL
;  263          1160  2         OUT_BUF_BLKS = 10,                              ! NUMBER OF BLOCKS IN OUTPUT BUFFER
;  264          1161  2         OUT_BUF_SIZ = OUT_BUF_BLKS * A_PAGE,            ! OUTPUT BUFFER SIZE
;  265          1162  2         NO_MORE = 0,                                    ! INDICATOR OF NO MORE ISD'S
;  266          1163  2         FILL_CHAR = %X'FFFF';                                   ! HEADER FILL CHARACTER
;  267          1164  2
;  268          1165  2 ! Temporary FAB and RAB blocks to be used when we have to do record I/O on the
;  269          1166  2 ! input file instead of $QIO's.
;  270          1167  2 !
;  271          1168  2
;  272          1169  2 LOCAL
;  273          1170  2
;  274        P 1171  2         TMPFAB  : $FAB (NAM = PAT$GL_OLDNBK,
;  275          1172  2                         FAC = GET),
;  276          1173  2
;  277        P 1174  2         TMPRAB  : $RAB (RBF = PAT$GB_INPBUF,
;  278        P 1175  2                         RSZ = 512,
;  279        P 1176  2                         UBF = PAT$GB_INPBUF,
;  280        P 1177  2                         USZ = 512,
;  281          1178  2                         FAB = TMPFAB);
;  282          1179  2 LOCAL
;  283          1180  2         CUR_VBN,                                        ! VBN TO READ
;  284          1181  2         BYTES_TO_READ,                                  ! NUMBER OF BYTES TO READ
;  285          1182  2         MAX_VBN_WRITTEN,                                ! VBN OF NEXT BLOCK TO BE WRITTEN INTO NEW I
;  286          1183  2         ECO_LEVEL_PTR : REF BITVECTOR,                  ! POINTER TO ECO LEVEL BITS IN IMAGE HEADER
;  287          1184  2         BLK_DIFF,                                       ! DIFFERENCE BETWEEN OLD AND NEW HEADER BLOC
;  288          1185  2         COUNTER,                                        ! NUMBER OF GLOBAL SYMBOLS
;  289          1186  2         NUM_HDR_BLKS,                                   ! NUMBER OF BLOCKS USED IN HEADER
;  290          1187  2         REM_BYTE_SIZ,                                   ! NUMBER OF UNUSED BYTES IN CURREN OUTPUT BL
;  291          1188  2         NEW_IHD_MAX,                                    ! MAXIMUM SIZE IN BYTES OF NEW IMAGE HEADER
;  292          1189  2         OUT_BUF_PTR,                                    ! POINTER TO OUTPUT BUFFER
;  293          1190  2         NXT_BYTE_PTR : REF VECTOR[,BYTE],               ! POINTER TO NEXT BYTE OF OUTPUT BUFFER
;  294          1191  2         ISD_PTR : REF BLOCK[,BYTE],                     ! POINTER TO CURRENT ISD
;  295          1192  2         ISE_PTR : REF BLOCK[,BYTE],                     ! POINTER TO CURRENT ISE
;  296          1193  2         NEW_IHD_PTR : REF BLOCK[,BYTE],                 ! POINTER TO NEW IMAGE HEADER
;  297          1194  2         NEW_IHSYM_PTR : REF BLOCK[,BYTE],               ! POINTER TO NEW IMAGE HEADER SYMBOL SECTION
;  298          1195  2         NEW_IHPAT_PTR : REF BLOCK[,BYTE],               ! POINTER TO NEW IMAGE HEADER PATCH SECTION
;  299          1196  2         NEW_ISD_PTR : REF BLOCK[,BYTE],                 ! POINTER TO NEW IMAGE SECTION DESCRIPTOR
;  300          1197  2         COM_TXT_PTR : REF BLOCK[,BYTE],                 ! POINTER TO PATCH COMMAND TEXT BLOCK
;  301          1198  2         COM_PTR : REF VECTOR[,BYTE],                    ! POINTER TO NEXT PATCH COMMAND
;  302          1199  2         OLD_IHSYM_PTR : REF BLOCK[,BYTE];               ! POINTER TO OLD IMAGE HEADER SYMBOL SECTION
;  303          1200  2
;  304          1201  2
;  305          1202  2 IF .PAT$GL_FLAGS [PAT$S_ABSOLUTE]                      ! If the file has been patched in absolute m
;  306          1203  2 THEN                                                    ! write the file as a simple binary file.
;  307          1204  3     BEGIN                                              ! The file is extracted from the 'ISE/ISD' l
```

```
;   308          1205  3     WRITE_BINARY ();                                    ! and written to the new file in typical pat
;   309          1206  3     RETURN;                                             ! Upon completion of this task return for fu
;   310          1207  2     END;                                                ! patch activity (normal course of action).
;   311          1208  2
;   312          1209  2 !++
;   313          1210  2 ! ALLOCATE A BUFFER FOR THE HEADER AND INITIALIZE POINTERS.
;   314          1211  2 !--
;   315          1212  2 IF .PAT$GL_IMGHDR[IHD$W_SYMDBGOFF] NEQ 0
;   316          1213  2 THEN
;   317          1214  2         OLD_IHSYM_PTR=CH$PTR(.PAT$GL_IMGHDR, .PAT$GL_IMGHDR[IHD$W_SYMDBGOFF])
;   318          1215  2 ELSE
;   319          1216  2         OLD_IHSYM_PTR=0;
;   320          1217  2 NEW_IHD_MAX=(.PAT$GL_IMGHDR[IHD$B_HDRBLKCNT] +2) * A_PAGE;       ! GET BUFFER SIZE
;   321          1218  2 PAT$ALLOBLK(.NEW_IHD_MAX, NEW_IHD_PTR);                          ! ALLOCATE NEW IMAGE HEADER BUFFER
;   322          1219  2 PAT$ALLOBLK(OUT_BUF_SIZ, OUT_BUF_PTR);                         ! ALLOCATE OUTPUT BUFFER
;   323          1220  2 NXT_BYTE_PTR=CH$PTR(.NEW_IHD_PTR,0);                          ! INITIALIZE OUTPUT BUFFER POINTER
;   324          1221  2
;   325          1222  2 !++
;   326          1223  2 ! NOW MOVE IN THE FIXED SIZE PORTION OF THE IMAGE HEADER.
;   327          1224  2 ! INITIALIZE POINTERS TO PIECES OF THE NEW HEADER.
;   328          1225  2 !--
;   329          1226  2 NXT_BYTE_PTR=CH$MOVE(.PAT$GL_IMGHDR[IHD$W_SIZE], .PAT$GL_IMGHDR, .NXT_BYTE_PTR);
;   330          1227  2 REM_BYTE_SIZ=A_PAGE - .PAT$GL_IMGHDR[IHD$W_SIZE];
;   331          1228  2 IF .NEW_IHD_PTR[IHD$W_SYMDBGOFF] NEQ 0
;   332          1229  2 THEN
;   333          1230  2         NEW_IHSYM_PTR=CH$PTR(.NEW_IHD_PTR, .NEW_IHD_PTR[IHD$W_SYMDBGOFF])
;   334          1231  2 ELSE
;   335          1232  2         NEW_IHD_PTR = 0;
;   336          1233  2 NEW_IHPAT_PTR=CH$PTR(.NEW_IHD_PTR, .NEW_IHD_PTR[IHD$W_PATCHOFF]);
;   337          1234  2
;   338          1235  2 !++
;   339          1236  2 ! NOW MOVE IN THE IMAGE SECTION DESCRIPTORS.
;   340          1237  2 !--
;   341          1238  2 NUM_HDR_BLKS = 1;                                           ! COUNT THE FIRST HEADER BLOCK
;   342          1239  2 ISE_PTR=CH$PTR(.PAT$GL_ISELHD,0);                          ! INITIALIZE POINTER TO FIRST ISE
;   343          1240  2 REM_BYTE_SIZ = .REM_BYTE_SIZ - A_WORD;                     ! LEAVE ROOM FOR IMAGE TYPE IDENTIFIER -
;   344          1241  2 WHILE .ISE_PTR NEQA 0                                     ! WORD AT END OF BLOCK.
;   345          1242  2 DO
;   346          1243  3         BEGIN
;   347          1244  3         IF .NUM_HDR_BLKS GTR .PAT$GL_IMGHDR[IHD$B_HDRBLKCNT]+2  ! CHECK IF HEADER OVERFLOWED BUFFER
;   348          1245  3         THEN
;   349          1246  3                 SIGNAL(PAT$_HDRBLK);
;   350          1247  3         ISD_PTR=CH$PTR(.ISE_PTR, ISE$C_SIZE);               ! POINT TO ISD
;   351          1248  4         IF (.REM_BYTE_SIZ GTR                               ! CHECK IF THERE IS ENOUGH ROOM FOR:
;   352          1249  5                         (IF .ISE_PTR[ISE$L_NXTISE] EQLA 0   ! (IF THIS IS LAST ISD, THEN
;   353          1250  5                          THEN .ISD_PTR[ISD$W_SIZE] + A_WORD ! THE ISD AND A NO-MORE INDICATOR,
;   354          1251  4                          ELSE .ISD_PTR[ISD$W_SIZE]))        ! OTHERWISE, JUST THE ISD.)
;   355          1252  3         THEN
;   356          1253  4                 BEGIN                                       ! YES, THEN MOVE IT IN
;   357          1254  4                 NXT_BYTE_PTR=CH$MOVE(.ISD_PTR[ISD$W_SIZE], .ISD_PTR, .NXT_BYTE_PTR);
;   358          1255  4                 REM_BYTE_SIZ=.REM_BYTE_SIZ - .ISD_PTR[ISD$W_SIZE];
;   359          1256  4                 ISE_PTR = .ISE_PTR[ISE$L_NXTISE];           ! POINT TO NEXT ISE
;   360          1257  4                 END
;   361          1258  3         ELSE
;   362          1259  4                 BEGIN
;   363          1260  4                 NXT_BYTE_PTR = CH$FILL(FILL_CHAR, .REM_BYTE_SIZ+A_WORD, .NXT_BYTE_PTR); ! INSERT FILL CHARAC
;   364          1261  4                 IF .NUM_HDR_BLKS EQL 1                      ! IF THIS IS THE FIRST HEADER BLOCK, RESTORE
```

```
;  365     1262  4                 THEN .NXT_BYTE_PTR - 2 = .PAT$GW_IMGTYP;        ! THE IMAGE TYPE IDENTIFIER WORD.
;  366     1263  4                 NUM_HDR_BLKS = .NUM_HDR_BLKS + 1;               ! ONE MORE HEADER BLOCK
;  367     1264  4                 REM_BYTE_SIZ = A_PAGE - A_WORD;                 ! SET NEW BLOCK BYTE COUNT
;  368     1265  3                 END;
;  369     1266  2         END;
;  370     1267  2 NXT_BYTE_PTR = CH$FILL(NO_MORE, A_WORD, .NXT_BYTE_PTR);
;  371     1268  2 NXT_BYTE_PTR = CH$FILL(FILL_CHAR, .REM_BYTE_SIZ, .NXT_BYTE_PTR);
;  372     1269  2 IF .NUM_HDR_BLKS EQL 1                                          ! IF THERE IS ONLY ONE HEADER BLOCK, RESTORE
;  373     1270  2 THEN .NXT_BYTE_PTR - 2 = .PAT$GW_IMGTYP;                       ! THE IMAGE TYPE IDENTIFIER WORD HERE.
;  374     1271  2
;  375     1272  2 !++
;  376     1273  2 ! NOW UPDATE THE IMAGE HEADER IF THE NUMBER OF HEADER BLOCKS CHANGED.
;  377     1274  2 !--
;  378     1275  2 IF (.PAT$GL_IMGHDR[IHD$B_HDRBLKCNT] NEQ .NUM_HDR_BLKS) OR
;  379     1276  3    (.PAT$GL_NEWVBNMX NEQ .PAT$GL_OLDVBNMX)
;  380     1277  2 THEN
;  381     1278  3         BEGIN
;  382     1279  3         !++
;  383     1280  3         ! FILE HEADER EXPANDED.  ALL THE VBN'S IN THE ISD'S MUST BE CHANGED.
;  384     1281  3         ! THAT IS, THE DIFFERENCE IN THE SIZE OF THE HEADERS MUST BE ADDED
;  385     1282  3         ! TO EVERY VBN IN THE HEADER.
;  386     1283  3         !--
;  387     1284  3         BLK_DIFF = .NUM_HDR_BLKS - .NEW_IHD_PTR[IHD$B_HDRBLKCNT];
;  388     1285  3         NEW_ISD_PTR = CH$PTR(.NEW_IHD_PTR, .NEW_IHD_PTR[IHD$W_SIZE]);
;  389     1286  3         WHILE .NEW_ISD_PTR[ISD$W_SIZE] NEQ 0
;  390     1287  3         DO
;  391     1288  4                 BEGIN
;  392     1289  5                 IF (NOT .NEW_ISD_PTR[ISD$V_DZRO]) AND (NOT .NEW_ISD_PTR[ISD$V_GBL])
;  393     1290  4                 THEN
;  394     1291  4                         NEW_ISD_PTR[ISD$L_VBN] = .NEW_ISD_PTR[ISD$L_VBN] + .BLK_DIFF;
;  395     1292  4                 NEW_ISD_PTR = CH$PTR(.NEW_ISD_PTR, .NEW_ISD_PTR[ISD$W_SIZE]);
;  396     1293  4                 IF .NEW_ISD_PTR[ISD$W_SIZE] EQL FILL_CHAR
;  397     1294  4                 THEN
;  398     1295  4                         !
;  399     1296  4                         ! Compute the beginning address of the next header block by rounding
;  400     1297  4                         ! up to the next page boundary. then adding the amount of scew into the page.
;  401     1298  4                         !
;  402     1299  4                         NEW_ISD_PTR = ((.NEW_ISD_PTR + A_PAGE - 1)/ A_PAGE * A_PAGE) +
;  403     1300  4                                       .NEW_IHD_PTR MOD A_PAGE;
;  404     1301  3                 END;
;  405     1302  3         NEW_IHD_PTR[IHD$B_HDRBLKCNT] = .NUM_HDR_BLKS;           ! RESET NUMBER OF HEADER BLOCKS
;  406     1303  3         BLK_DIFF = .BLK_DIFF + .PAT$GL_NEWVBNMX - .PAT$GL_OLDVBNMX;
;  407     1304  3         IF .NEW_IHSYM_PTR NEQA 0
;  408     1305  3         THEN
;  409     1306  4                 BEGIN
;  410     1307  4                 IF .NEW_IHSYM_PTR[IHS$L_DSTVBN] NEQ 0
;  411     1308  4                 THEN
;  412     1309  4                         NEW_IHSYM_PTR[IHS$L_DSTVBN] = .NEW_IHSYM_PTR[IHS$L_DSTVBN] + .BLK_DIFF;
;  413     1310  4
;  414     1311  4                 IF .NEW_IHSYM_PTR[IHS$L_DMTVBN] NEQ 0
;  415     1312  4                 THEN
;  416     1313  4                         NEW_IHSYM_PTR[IHS$L_DMTVBN] = .NEW_IHSYM_PTR[IHS$L_DMTVBN] + .BLK_DIFF;
;  417     1314  4
;  418     1315  4                 IF .NEW_IHSYM_PTR[IHS$L_GSTVBN] NEQ 0
;  419     1316  4                 THEN
;  420     1317  4                         NEW_IHSYM_PTR[IHS$L_GSTVBN] = .NEW_IHSYM_PTR[IHS$L_GSTVBN] + .BLK_DIFF;
;  421     1318  3                 END;
```

```
;  422     1319  3          IF .NEW_IHPAT_PTR[IHP$L_PATCOMTXT] NEQ 0
;  423     1320  3          THEN
;  424     1321  3                  NEW_IHPAT_PTR[IHP$L_PATCOMTXT] = .NEW_IHPAT_PTR[IHP$L_PATCOMTXT] + .BLK_DIFF;
;  425     1322  3          END
;  426     1323  2 ELSE
;  427     1324  2          BLK_DIFF = 0;
;  428     1325  2
;  429     1326  2 !++
;  430     1327  2 ! NOW COMPUTE THE SIZE OF THE NEW IMAGE FILE.  THEN OPEN IT, ALLOCATING THE
;  431     1328  2 ! NEEDED CONTIGUOUS SPACE.
;  432     1329  2 !--
;  433     1330  2 COM_TXT_PTR = CH$PTR(.PAT$GL_TXTLHD, 0);                    ! POINT TO FIRST COMMAND TEXT BLOCK
;  434     1331  2 WHILE .COM_TXT_PTR NEQA 0                                   ! INCREMENT IMAGE SIZE FOR EACH
;  435     1332  2 DO                                                         ! BLOCK OF COMMAND TEXT
;  436     1333  3          BEGIN
;  437     1334  3          PAT$GL_IMGBLKS = .PAT$GL_IMGBLKS + 1;
;  438     1335  3          COM_TXT_PTR = .COM_TXT_PTR[TXT$L_NXTBLK];          ! POINT TO NEXT COMMAND TEXT BLOCK
;  439     1336  2          END;
;  440     1337  2 PAT$GL_IMGBLKS = .PAT$GL_IMGBLKS + .BLK_DIFF;               ! ADD IN DIFFERENCE IN HEADER SIZES
;  441     1338  2 PAT$GL_NEWXABALL[XAB$L_ALQ] = .PAT$GL_IMGBLKS;             ! INITIALIZE NUMBER OF BLOCKS TO ALLOCATE
;  442     1339  2
;  443     1340  2 IF .PAT$GL_FLAGS [PAT$S_VOLUME]                           ! The VOLUME qualifier specifies the RVN
;  444     1341  2 THEN
;  445     1342  3          BEGIN
;  446     1343  3          PAT$GL_NEWXABALL[XAB$W_VOL] = .PAT$GW_IMGVOL;     ! The Relative Volume Number
;  447     1344  3          PAT$GL_NEWXABALL[XAB$B_ALN] = XAB$C_LBN;          ! To enable XAB$W_VOL
;  448     1345  2          END;
;  449     1346  2
;  450     1347  3 IF (.NUM_OF_UPDATES EQL 0)                                ! Check number of updates done
;  451     1348  2 THEN
;  452     1349  3          BEGIN
;  453     1350  3          !++
;  454     1351  3          ! Create the output image file.  Try to make it a contiguous file if
;  455     1352  3          ! the input image file was contiguous, i.e., first try a create with
;  456     1353  3          ! the same attributes.  If the file cannot be created with the same
;  457     1354  3          ! attributes, then attempt a second try with contiguous-best-try.  If
;  458     1355  3          ! this succeeds, then print an informational message.
;  459     1356  3          !--
;  460     1357  3          PAT$GL_NEWXABALL[XAB$V_CTG] = .PAT$GL_OLDFAB[FAB$V_CTG];
;  461     1358  3          PAT$GL_NEWXABALL[XAB$V_CBT] = .PAT$GL_OLDFAB[FAB$V_CBT];
;  462     1359  4          IF NOT (PAT$GL_ERRCODE=$CREATE(FAB=PAT$GL_NEWFAB))
;  463     1360  3          THEN
;  464     1361  4                  BEGIN                                     ! Attempt a contiguous best try
;  465     1362  4                  PAT$GL_NEWXABALL[XAB$V_CBT] = TRUE;
;  466     1363  4                  PAT$GL_ERRCODE = $CREATE(FAB=PAT$GL_NEWFAB);
;  467     1364  4                  IF .PAT$GL_ERRCODE
;  468     1365  4                  THEN
;  469     1366  4                          SIGNAL(PAT$_NONCONTIG+MSG$K_INFO,.PAT$GL_ERRCODE,.PAT$GL_NEWRAB[RAB$L_STV]),
;  470     1367  3                  END;
;  471     1368  3          END
;  472     1369  2 ELSE
;  473     1370  2          PAT$GL_ERRCODE=$OPEN(FAB=PAT$GL_NEWFAB);          ! OPEN OUTPUT FILE
;  474     1371  2 IF NOT .PAT$GL_ERRCODE                                     ! SUCCESS ON OPEN OR CREATE?
;  475     1372  2 THEN
;  476     1373  2          SIGNAL(PAT$_OPENOUT,1,GETFILDSC(PAT$GL_NEWFAB),.PAT$GL_NEWFAB[FAB$L_STS],.PAT$GL_NEWRAB[RAB$L_STV])
;  477     1374  2 ELSE
;  478     1375  3          BEGIN
```

```
;  479          1376  3          PAT$GL_ERRCODE=$CONNECT(RAB=PAT$GL_NEWRAB);                     ! CONNECT INPUT FILE
;  480          1377  3          IF NOT .PAT$GL_ERRCODE                                          ! SUCCESS ON CONNECT?
;  481          1378  3          THEN
;  482          1379  3          SIGNAL(PAT$_OPENOUT,1,GETFILDSC(PAT$GL_NEWFAB),.PAT$GL_NEWRAB[RAB$L_STS],.PAT$GL_NEWRAB[RAB$L_STV])
;  483          1380  3          ELSE
;  484          1381  4                  BEGIN
;  485          1382  4                  NUM_OF_UPDATES = 1;                                     ! SET INDICATOR FOR ALREADY CREATED
;  486          1383  4                  PAT$GL_FLAGS [PAT$S_OUTPUT] = 1;                        ! SET FLAG FILE NOT OPEN
;  487          1384  4                  PAT$GL_NEWFAB[FAB$V_ESC] = TRUE;
;  488          1385  4                  PAT$GL_NEWFAB[FAB$L_CTX] = RME$C_SETRFM;                ! SET MODIFY CODE
;  489          1386  4                  PAT$GL_NEWFAB[FAB$B_RFM] = FAB$C_VAR;                   ! SET VARIABLE LENGTH RECORDS
;  490          1387  4                  PAT$GL_ERRCODE = $MODIFY(FAB=PAT$GL_NEWFAB);
;  491          1388  4                  IF NOT .PAT$GL_ERRCODE
;  492          1389  4                  THEN
;  493          1390  4                          SIGNAL(PAT$_MODIFYERR, 3, .PAT$GL_ERRCODE,
;  494          1391  4                                  .PAT$GL_NEWNBK[NAM$B_RSL], PAT$GB_NEWNAME,
;  495          1392  4                                  .PAT$GL_ERRCODE, .PAT$GL_NEWRAB[RAB$L_STV]);
;  496          1393  4                  END
;  497          1394  2          END;
;  498          1395  2
;  499          1396  2 !++
;  500          1397  2 ! REPORT FILE BEING WRITTEN.
;  501          1398  2 !--
;  502          1399  2 SIGNAL(PAT$_WRTFIL+MSG$K_INFO, 2, .PAT$GL_NEWNBK[NAM$B_RSL], PAT$GB_NEWNAME);
;  503          1400  2
;  504          1401  2 !++
;  505          1402  2 ! MAKE SURE THE FILE IS OPEN.
;  506          1403  2 !--
;  507          1404  2 IF NOT .PAT$GL_FLAGS [PAT$S_OUTPUT]
;  508          1405  2 OR NOT .PAT$GL_FLAGS [PAT$S_INPUT]
;  509          1406  2 THEN    RETURN;                                                           ! CAN'T GET AT FILES, GIVE UP
;  510          1407  2
;  511          1408  2 !++
;  512          1409  2 ! Now write out the image binary.
;  513          1410  2 !--
;  514          1411  2 ISE_PTR=CH$PTR(.PAT$GL_ISELHD,0);                                        ! POINT TO FIRST ISE
;  515          1412  2 NEW_ISD_PTR = CH$PTR(.NEW_IHD_PTR, .NEW_IHD_PTR[IHD$W_SIZE]);           ! POINT TO FIRST NEW ISD
;  516          1413  2 MAX_VBN_WRITTEN = 0;                                                     ! NO VBN WRITTEN YET
;  517          1414  2 WHILE .ISE_PTR NEQA 0                                                   ! LOOP UNTIL ISE'S ARE EXHAUSTED
;  518          1415  2 DO
;  519          1416  3          BEGIN
;  520          1417  3          ISD_PTR = CH$PTR(.ISE_PTR, ISE$C_SIZE);                         ! FIND OLD ISD ADDRESS
;  521          1418  3          COUNTER = 0;                                                    ! SET COUNT OF BLOCKS WRITTEN FOR IMAGE SECT
;  522          1419  3          IF (NOT .ISD_PTR[ISD$V_DZRO]) AND                               ! CHECK FOR NO IMAGE BINARY
;  523          1420  3             (.ISD_PTR[ISD$B_TYPE] NEQ ISD$K_USRSTACK) AND                ! AND FOR IMAGE STACK
;  524          1421  4             (    (NOT .ISD_PTR[ISD$V_GBL])  OR                           ! AND FOR GLOBAL SECTIONS WITH
;  525          1422  4                  (.ISD_PTR[ISD$V_GBL] AND (.ISD_PTR[ISD$L_VBN] NEQ 0))) ! NO LOCAL COPY.
;  526          1423  3          THEN
;  527          1424  4                  BEGIN
;  528          1425  4                  !++
;  529          1426  4                  ! SET VBN OF IMAGE SECTION IN OLD AND NEW FILES.
;  530          1427  4                  !--
;  531          1428  4                  CUR_VBN = .ISD_PTR[ISD$L_VBN];
;  532          1429  4                  PAT$GL_NEWRAB[RAB$L_BKT] = .NEW_ISD_PTR[ISD$L_VBN];
;  533          1430  4
;  534          1431  4                  !++
;  535          1432  4                  ! NOW LOOP TO OUTPUT ALL OF THIS IMAGE SECTION.
```

```
;  536     1433  4          ! (THE LOOP IS IN CASE THE IMAGE SECTION IS LARGER THAN THE OUTPUT BUFFER.)
;  537     1434  4          !--
;  538     1435  4          WHILE .COUNTER NEQ .ISD_PTR[ISD$W_PAGCNT]
;  539     1436  4          DO
;  540     1437  5                  BEGIN
;  541     1438  5                  IF .ISE_PTR[ISE$L_MAPVEND] EQLA 0       ! IF THE IMAGE SECTION IS NOT MAPPED
;  542     1439  5                  THEN                                    ! THEN READ IT
;  543     1440  6                          BEGIN
;  544     1441  6                          !++
;  545     1442  6                          ! CHECK THAT THE BUFFER IS LARGE ENOUGH TO HOLD ALL OF IMAGE SECTION.
;  546     1443  6                          !--
;  547     1444  7                          IF OUT_BUF_SIZ GTR ((.ISD_PTR[ISD$W_PAGCNT]-.COUNTER)*A_PAGE)
;  548     1445  6                          THEN
;  549     1446  7                                  BEGIN
;  550     1447  7                                  !++
;  551     1448  7                                  ! BUFFER WAS LARGE ENOUGH.  SET UP TO READ ENTIRE IMAGE SECTION.
;  552     1449  7                                  !--
;  553     1450  7                                  BYTES_TO_READ = A_PAGE * (.ISD_PTR[ISD$W_PAGCNT]-.COUNTER);
;  554     1451  7                                  PAT$GL_NEWRAB[RAB$W_RSZ] = .BYTES_TO_READ;
;  555     1452  7                                  COUNTER = .ISD_PTR[ISD$W_PAGCNT];
;  556     1453  7                                  END
;  557     1454  6                          ELSE
;  558     1455  7                                  BEGIN
;  559     1456  7                                  !++
;  560     1457  7                                  ! BUFFER WAS NOT LARGE ENOUGH TO READ ENTIRE IMAGE SECTION.
;  561     1458  7                                  ! THEREFORE, SET COUNTER TO READ TEN BLOCKS.
;  562     1459  7                                  !--
;  563     1460  7                                  BYTES_TO_READ = OUT_BUF_SIZ;
;  564     1461  7                                  PAT$GL_NEWRAB[RAB$W_RSZ] = OUT_BUF_SIZ;
;  565     1462  7                                  COUNTER = .COUNTER + 10;
;  566     1463  6                                  END;
;  567     1464  6
;  568     1465  6                          !++
;  569     1466  6                          ! NOW READ IMAGE SECTION.  IF IMAGE SECTION IS TOO LARGE
;  570     1467  6                          ! FOR BUFFER, READ TEN BLOCKS OF IT.
;  571     1468  6                          !--
;  572     1469  6                          PAT$GL_ERRCODE = GET_IMAGE_BLOCK ( .CUR_VBN,
;  573     1470  6                                                             .OUT_BUF_PTR,
;  574     1471  6                                                             .BYTES_TO_READ );
;  575     1472  6                          IF NOT .PAT$GL_ERRCODE
;  576     1473  6                          THEN
;  577     1474  6                                  SIGNAL(PAT$_READERR, 1, GETFILDSC(PAT$GL_OLDFAB),
;  578     1475  6                                          .PAT$GL_ERRCODE, 0);
;  579     1476  6
;  580     1477  6                          !++
;  581     1478  6                          ! INITIALIZE THE OUTPUT BUFFER ADDRESS.
;  582     1479  6                          !--
;  583     1480  6                          PAT$GL_NEWRAB[RAB$L_RBF] = .OUT_BUF_PTR;
;  584     1481  6                          END
;  585     1482  5                  ELSE
;  586     1483  6                          BEGIN
;  587     1484  6                          !++
;  588     1485  6                          ! THIS IMAGE SECTION WAS MAPPED.  CHECK IF
;  589     1486  6                          ! THE ENTIRE SECTION SHOULD BE WRITTEN OR TEN
;  590     1487  6                          ! BLOCKS AT A TIME.
;  591     1488  6                          !--
;  592     1489  7                          IF OUT_BUF_SIZ GTR ((.ISD_PTR[ISD$W_PAGCNT]-.COUNTER)*A PAGE)
```

```
;  593     1490  6                    THEN
;  594     1491  6                        !++
;  595     1492  6                        ! SET THE OUTPUT BUFFER ADDRESS EQUAL
;  596     1493  6                        ! TO THE STARTING MAPPED ADDRESS PLUS
;  597     1494  6                        ! AN OFFSET FOR THE BLOCKS ALREADY WRITTEN.
;  598     1495  6                        ! OUTPUT THE REST OF THE IMAGE SECTION.
;  599     1496  6                        !--
;  600     1497  7                        BEGIN
;  601     1498  7                        PAT$GL_NEWRAB[RAB$W_RSZ] = A_PAGE *
;  602     1499  7                                (.NEW_ISD_PTR[ISD$W_PAGCNT] - .COUNTER);
;  603     1500  7                        PAT$GL_NEWRAB[RAB$L_RBF] = .ISE_PTR[ISE$L_MAPVST] +
;  604     1501  7                                (A_PAGE * .COUNTER);
;  605     1502  7                        COUNTER = .ISD_PTR[ISD$W_PAGCNT];
;  606     1503  7                        END
;  607     1504  6                    ELSE
;  608     1505  7                        BEGIN
;  609     1506  7                        !++
;  610     1507  7                        ! BUFFER WAS NOT LARGE ENOUGH TO READ
;  611     1508  7                        ! ENTIRE IMAGE SECTION. THEREFORE, SET
;  612     1509  7                        ! COUNTER TO WRITE TEN BLOCKS.
;  613     1510  7                        !--
;  614     1511  7                        PAT$GL_NEWRAB[RAB$W_RSZ] = OUT_BUF_SIZ;
;  615     1512  7                        PAT$GL_NEWRAB[RAB$L_RBF] =
;  616     1513  7                                .ISE_PTR[ISE$L_MAPVST] + (A_PAGE * .COUNTER);
;  617     1514  7                        COUNTER = .COUNTER + 10;
;  618     1515  6                        END;
;  619     1516  5                    END;
;  620     1517  5
;  621     1518  5                !++
;  622     1519  5                ! NOW WRITE OUT THE IMAGE SECTION (OR PART OF IT).
;  623     1520  5                !--
;  624     1521  5                PAT$GL_ERRCODE = $WRITE(RAB=PAT$GL_NEWRAB);
;  625     1522  5                IF NOT .PAT$GL_ERRCODE
;  626     1523  5                THEN
;  627     1524  5                        SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB),
;  628     1525  5                                .PAT$GL_NEWRAB[RAB$L_STS],
;  629     1526  5                                .PAT$GL_NEWRAB[RAB$L_STV]);
;  630     1527  5
;  631     1528  5                !++
;  632     1529  5                ! CHECK TO SEE THAT THE ENTIRE IMAGE SECTION HAS BEEN WRITTEN.
;  633     1530  5                ! IF NOT, RESET THE VBN'S FOR THE NEXT TEN BLOCKS OF IT.
;  634     1531  5                !--
;  635     1532  5                IF .COUNTER NEQ .ISD_PTR[ISD$W_PAGCNT]
;  636     1533  5                THEN
;  637     1534  6                        BEGIN
;  638     1535  6                        CUR_VBN = .ISD_PTR[ISD$L_VBN] + .COUNTER;
;  639     1536  6                        PAT$GL_NEWRAB[RAB$L_BKT] = .NEW_ISD_PTR[ISD$L_VBN] + .COUNTER;
;  640     1537  5                        END;
;  641     1538  4                END;
;  642     1539  4            !++
;  643     1540  4            ! Now update the pointer to the next highest VBN to be write.
;  644     1541  4            !--
;  645     1542  5            IF (.MAX_VBN_WRITTEN LSSU (.NEW_ISD_PTR[ISD$L_VBN] + .NEW_ISD_PTR[ISD$W_PAGCNT]))
;  646     1543  4            THEN
;  647     1544  4                    MAX_VBN_WRITTEN = .NEW_ISD_PTR[ISD$L_VBN] + .NEW_ISD_PTR[ISD$W_PAGCNT];
;  648     1545  3            END;
;  649     1546  3        ISE_PTR = .ISE_PTR[ISE$L_NXTISE];
```

```
;  650          1547  3          NEW_ISD_PTR = CH$PTR(.NEW_ISD_PTR, .NEW_ISD_PTR[ISD$W_SIZE]);
;  651          1548  3          !++
;  652          1549  3          ! Now position to the next block of ISD's.
;  653          1550  3          !--
;  654          1551  3          IF .NEW_ISD_PTR[ISD$W_SIZE] EQL FILL_CHAR
;  655          1552  3          THEN
;  656          1553  3                  NEW_ISD_PTR = .NEW_IHD_PTR +
;  657          1554  3                                ((.NEW_ISD_PTR - .NEW_IHD_PTR + A_PAGE-1) / A_PAGE) * A_PAGE);
;  658          1555  2          END;
;  659          1556  2
;  660          1557  2  !++
;  661          1558  2  ! INITIALIZE THE OUTPUT FILE TO TRUNCATE THE FILE AFTER A PUT.  OTHERWISE, PATCH
;  662          1559  2  ! GETS AN NEF ERROR (NOT AT END OF FILE).
;  663          1560  2  !--
;  664          1561  2  PAT$GL_NEWRAB[RAB$V_TPT] = TRUE;
;  665          1562  2
;  666          1563  2  !++
;  667          1564  2  ! NOW WRITE OUT THE DEBUG SYMBOL TABLE BLOCKS.
;  668          1565  2  !**** THIS CODE WILL BE CHANGE WHEN PATCH HANDLES SYMBOLS. ****
;  669          1566  2  !--
;  670          1567  3  IF (.OLD_IHSYM_PTR NEQA 0)
;  671          1568  2  THEN
;  672          1569  3          BEGIN
;  673          1570  4          IF (.OLD_IHSYM_PTR[IHS$W_DSTBLKS] NEQ 0) AND (.OLD IHSYM_PTR[IHS$L_DSTVBN] GTR 2)
;  674          1571  3          THEN
;  675          1572  4                  BEGIN
;  676          1573  4                  COUNTER = 0;
;  677          1574  4                  NEW_IHSYM_PTR[IHS$L_DSTVBN] = .MAX_VBN_WRITTEN;
;  678          1575  4                  PAT$GL_NEWRAB[RAB$L_BKT] = .MAX_VBN_WRITTEN;
;  679          1576  4                  CUR_VBN = .OLD_IHSYM_PTR[IHS$L_DSTVBN];
;  680          1577  4                  WHILE .COUNTER NEQ .OLD_IHSYM_PTR[IHS$W_DSTBLKS]
;  681          1578  4                  DO
;  682          1579  5                          BEGIN
;  683          1580  5                          !++
;  684          1581  5                          ! CHECK THAT THE BUFFER IS LARGE ENOUGH TO HOLD ALL OF IMAGE SECTION.
;  685          1582  5                          !--
;  686          1583  6                          IF OUT_BUF_SIZ GTR ((.OLD_IHSYM_PTR[IHS$W_DSTBLKS]-.COUNTER)*A_PAGE)
;  687          1584  5                          THEN
;  688          1585  6                                  BEGIN
;  689          1586  6                                  !++
;  690          1587  6                                  ! BUFFER WAS LARGE ENOUGH.  SET UP TO READ ENTIRE IMAGE SECTION.
;  691          1588  6                                  !--
;  692          1589  6                                  BYTES_TO_READ = A_PAGE * (.OLD_IHSYM_PTR[IHS$W_DSTBLKS]-.COUNTER);
;  693          1590  6                                  PAT$GL_NEWRAB[RAB$W_RSZ] = .BYTES_TO_READ;
;  694          1591  6                                  COUNTER = .OLD_IHSYM_PTR[IHS$W_DSTBLKS];
;  695          1592  6                                  END
;  696          1593  5                          ELSE
;  697          1594  6                                  BEGIN
;  698          1595  6                                  !++
;  699          1596  6                                  ! BUFFER WAS NOT LARGE ENOUGH TO READ ENTIRE IMAGE SECTION.
;  700          1597  6                                  ! THEREFORE, SET COUNTER TO READ TEN BLOCKS.
;  701          1598  6                                  !--
;  702          1599  6                                  BYTES_TO_READ = OUT_BUF_SIZ;
;  703          1600  6                                  PAT$GL_NEWRAB[RAB$W_RSZ] = OUT_BUF_SIZ;
;  704          1601  6                                  COUNTER = .COUNTER + 10;
;  705          1602  5                                  END;
;  706          1603  5
```

```
;  707     1604  5                !++
;  708     1605  5                ! NOW READ IMAGE SECTION.  IF IMAGE SECTION IS TOO LARGE
;  709     1606  5                ! FOR BUFFER, READ TEN BLOCKS OF IT.
;  710     1607  5                !--
;  711     1608  5                PAT$GL_ERRCODE = GET_IMAGE_BLOCK ( .CUR_VBN,
;  712     1609  5                                                   .OUT_BUF_PTR,
;  713     1610  5                                                   .BYTES_TO_READ );
;  714     1611  5                IF NOT .PAT$GL_ERRCODE
;  715     1612  5                THEN
;  716     1613  5                        SIGNAL(PAT$_READERR, 1, GETFILDSC(PAT$GL_OLDFAB), .PAT$GL_ERRCODE, 0);
;  717     1614  5
;  718     1615  5                !++
;  719     1616  5                ! INITIALIZE THE OUTPUT BUFFER ADDRESS.
;  720     1617  5                !--
;  721     1618  5                PAT$GL_NEWRAB[RAB$L_RBF] = .OUT_BUF_PTR;
;  722     1619  5
;  723     1620  5                !++
;  724     1621  5                ! NOW WRITE OUT THE DST (OR PART OF IT).
;  725     1622  5                !--
;  726     1623  5                PAT$GL_ERRCODE = $WRITE(RAB=PAT$GL_NEWRAB);
;  727     1624  5                IF NOT .PAT$GL_ERRCODE
;  728     1625  5                THEN
;  729     1626  5                        SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS]
;  730     1627  5
;  731     1628  5                !++
;  732     1629  5                ! CHECK TO SEE THAT THE ENTIRE DST HAS BEEN WRITTEN.
;  733     1630  5                ! IF NOT, RESET THE VBN'S FOR THE NEXT TEN BLOCKS OF IT.
;  734     1631  5                !--
;  735     1632  5                IF .COUNTER NEQ .OLD_IHSYM_PTR[IHS$W_DSTBLKS]
;  736     1633  5                THEN
;  737     1634  6                        BEGIN
;  738     1635  6                        CUR_VBN = .OLD_IHSYM_PTR[IHS$L_DSTVBN] + .COUNTER;
;  739     1636  6                        PAT$GL_NEWRAB[RAB$L_BKT] = .NEW_IHSYM_PTR[IHS$L_DSTVBN] + .COUNTER;
;  740     1637  5                        END;
;  741     1638  4                END;
;  742     1639  3        END;
;  743     1640  4        IF (.MAX_VBN_WRITTEN LSSU (.NEW_IHSYM_PTR[IHS$L_DSTVBN] + .NEW_IHSYM_PTR[IHS$W_DSTBLKS]))
;  744     1641  3        THEN
;  745     1642  3                MAX_VBN_WRITTEN = .NEW_IHSYM_PTR[IHS$L_DSTVBN] + .NEW_IHSYM_PTR[IHS$W_DSTBLKS];
;  746     1643  3
;  747     1644  3        !++
;  748     1645  3        !
;  749     1646  3        ! NOW WRITE OUT THE DEBUG MODULE/PSECT TABLE (DMT) BLOCKS.
;  750     1647  3        !
;  751     1648  3        !--
;  752     1649  3        IF .PAT$GL_IMGHDR [IHD$V_DBGDMT]                                        ! Did the latest linker gene
;  753     1650  3        THEN                                                                    ! If so, then ...
;  754     1651  3            IF .OLD_IHSYM_PTR [IHS$L_DMTBYTES] NEQ 0                            ! ...are there any DMT entri
;  755     1652  3            THEN                                                                ! Yes, propagate the DMT to
;  756     1653  4                BEGIN                                                           ! NOTE: At this point it is
;  757     1654  4                BLK_DIFF = (.OLD_IHSYM_PTR[IHS$L_DMTBYTES] + A_PAGE) / A_PAGE;  ! Number of pages used for t
;  758     1655  4                COUNTER = 0;                                                    ! Reset local transfer count
;  759     1656  4                NEW_IHSYM_PTR [IHS$L_DMTVBN] = .MAX_VBN_WRITTEN;                ! Set the new starting VBN o
;  760     1657  4                PAT$GL_NEWRAB[RAB$L_BKT] = .MAX_VBN_WRITTEN;                    ! Point to the DMT's VBN in
;  761     1658  4                CUR_VBN = .OLD_IHSYM_PTR[IHS$L_DMTVBN];                         ! Point to the old starting
;  762     1659  4
;  763     1660  4                WHILE .COUNTER LSS .BLK_DIFF DO                                 ! Copy the entire DMT.
```

```
;  764          1661  5                BEGIN
;  765          1662  5                IF OUT_BUF_SIZ GEQ .OLD_IHSYM_PTR[IHS$L_DMTBYTES] -          ! Is the buffer large enough
;  766          1663  6                                  (.COUNTER * A_PAGE)                        ! the entire (or remaining)
;  767          1664  5                THEN
;  768          1665  6                    BEGIN                                                    ! Yes,
;  769          1666  6                    BYTES_TO_READ = (.BLK_DIFF - .COUNTER) * A_PAGE;         ! Set the number of DMT byte
;  770          1667  6                    COUNTER = .COUNTER + (.BLK_DIFF - .COUNTER);             ! Set number of blocks trans
;  771          1668  6                    END
;  772          1669  5                ELSE
;  773          1670  6                    BEGIN                                                    ! No,
;  774          1671  6                    BYTES_TO_READ = OUT_BUF_SIZ;                             ! Copy a buffer full at a ti
;  775          1672  6                    COUNTER = .COUNTER + OUT_BUF_BLKS;                       ! Update number of blocks tr
;  776          1673  5                    END;
;  777          1674  5
;  778          1675  5                PAT$GL_NEWRAB[RAB$W_RSZ] = .BYTES_TO_READ;                   ! Propagate the byte count t
;  779          1676  5
;  780          1677  5                !++
;  781          1678  5                ! NOW READ IMAGE SECTION.  IF IMAGE SECTION IS TOO LARGE
;  782          1679  5                ! FOR BUFFER, READ TEN BLOCKS OF IT.
;  783          1680  5                !--
;  784          1681  5                PAT$GL_ERRCODE = GET_IMAGE_BLOCK ( .CUR_VBN,
;  785          1682  5                                                   .OUT_BUF_PTR,
;  786          1683  5                                                   .BYTES_TO_READ );
;  787          1684  5                IF NOT .PAT$GL_ERRCODE
;  788          1685  5                THEN
;  789          1686  5                   SIGNAL(PAT$_READERR, 1, GETFILDSC(PAT$GL_OLDFAB), .PAT$GL_ERRCODE, 0);
;  790          1687  5
;  791          1688  5                !++
;  792          1689  5                ! INITIALIZE THE OUTPUT BUFFER ADDRESS AND WRITE THE DMT (OR PART THERE OF...)
;  793          1690  5                !--
;  794          1691  5                PAT$GL_NEWRAB[RAB$L_RBF] = .OUT_BUF_PTR;
;  795          1692  5                PAT$GL_ERRCODE = $WRITE(RAB=PAT$GL_NEWRAB);
;  796          1693  5                IF NOT .PAT$GL_ERRCODE
;  797          1694  5                THEN
;  798          1695  5                    SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS], .PAT$G
;  799          1696  5
;  800          1697  5                !++
;  801          1698  5                ! CHECK TO SEE THAT THE ENTIRE DMT HAS BEEN WRITTEN.
;  802          1699  5                ! IF NOT, RESET THE VBN'S FOR THE NEXT TEN BLOCKS OF IT.
;  803          1700  5                !--
;  804          1701  5                IF .COUNTER NEQ .BLK_DIFF
;  805          1702  5                THEN
;  806          1703  6                    BEGIN
;  807          1704  6                    CUR_VBN  = .OLD_IHSYM_PTR[IHS$L_DMTVBN] + .COUNTER;
;  808          1705  6                    PAT$GL_NEWRAB[RAB$L_BRT] = .NEW_IHSYM_PTR[IHS$L_DMTVBN] + .COUNTER;
;  809          1706  5                    END;
;  810          1707  5
;  811          1708  4                END; ! WHILE
;  812          1709  4
;  813          1710  5             IF (.MAX_VBN_WRITTEN LSSU (.NEW_IHSYM_PTR[IHS$L_DMTVBN] + .BLK_DIFF))
;  814          1711  4             THEN
;  815          1712  4                MAX_VBN_WRITTEN = .NEW_IHSYM_PTR[IHS$L_DMTVBN] + .BLK_DIFF;
;  816          1713  4
;  817          1714  3             END; ! IF (DMT present)
;  818          1715  3
;  819          1716  2       END; ! of DST/DMT processing.
```

```
;   821          1717  2
;   822          1718  2 ! Now reopen the input file with another FAB so that we can change
;   823          1719  2 ! the processing from QIO's to record I/O.  We need to do record I/O in the
;   824          1720  2 ! symbol table and patch commands since information needs to be appended to
;   825          1721  2 ! these in the new file.
;   826          1722  2 !
;   827          1723  2
;   828          1724  2 TMPFAB [FAB$L_FNA] = PAT$GB_OLDNAME;
;   829          1725  2 TMPFAB [FAB$B_FNS] = .PAT$G[_OLDNBK [NAM$B_RSL];
;   830          1726  2 PAT$GL_ERRCODE = $OPEN (FAB=TMPFAB);
;   831          1727  2 IF NOT .PAT$GL_ERRCODE
;   832          1728  2 THEN
;   833          1729  2     SIGNAL (PAT$_OPENIN, 1, GETFILDSC(TMPFAB),
;   834          1730  2             .TMPFAB [FAB$L_STS], .TMPFAB [FAB$L_STV]);
;   835          1731  2
;   836          1732  2 PAT$GL_ERRCODE = $CONNECT (RAB=TMPRAB);
;   837          1733  2 IF NOT .PAT$GL_ERRCODE
;   838          1734  2 THEN
;   839          1735  2     SIGNAL (PAT$_OPENIN, 1, GETFILDSC(TMPFAB),
;   840          1736  2             .TMPRAB [RAB$L_STS], .TMPRAB [RAB$L_STV]);
;   841          1737  2
;   842          1738  2 ! Change attributes to be able to do record I/O.
;   843          1739  2 !
;   844          1740  2 TMPFAB [FAB$V_ESC] = TRUE;
;   845          1741  2 TMPFAB [FAB$L_CTX] = RME$C_SETRFM;
;   846          1742  2 TMPFAB [FAB$B_RFM] = FAB$C_VAR;
;   847          1743  2
;   848          1744  3 IF NOT (PAT$GL_ERRCODE = $MODIFY (FAB=TMPFAB))
;   849          1745  2 THEN
;   850          1746  2     SIGNAL (PAT$_OPENIN, 1, GETFILDSC(TMPFAB),
;   851          1747  2             .TMPRAB [RAB$L_STS], .TMPRAB [RAB$L_STV]);
;   852          1748  2
;   853          1749  2 ! THE FIRST GET/PUT MUST BE BY RFA AND THE REST SEQUENTIALLY.
;   854          1750  2 !
;   855          1751  2 PAT$GL_NEWRAB[RAB$W_USZ] = A_PAGE;
;   856          1752  2 PAT$GL_NEWRAB[RAB$L_UBF] = .OUT_BUF_PTR;
;   857          1753  3 IF (.NEW_IHSYM_PTR[IHS$L_GSTVBN] NEQ 0)
;   858          1754  2 THEN
;   859          1755  2         NEW_IHSYM_PTR[IHS$L_GSTVBN] = .MAX_VBN_WRITTEN;
;   860          1756  3 IF (.MAX_VBN_WRITTEN NEQ 0)
;   861          1757  2 THEN
;   862          1758  2         PAT$GL_NEWRAB[RAB$L_BKT] = .MAX_VBN_WRITTEN - 1
;   863          1759  2 ELSE
;   864          1760  2         SIGNAL(PAT$_PATERR);
;   865          1761  2
;   866          1762  2 PAT$GL_ERRCODE = $READ(RAB=PAT$GL_NEWRAB);
;   867          1763  2 IF NOT .PAT$GL_ERRCODE
;   868          1764  2 THEN
;   869          1765  2         SIGNAL(PAT$_READERR, 3, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS], .PAT$GL_NEWRAB[RAB$L_ST
;   870          1766  2
;   871          1767  2 ! Initialize to read the global symbol table.  The first record is read by RFA.
;   872          1768  2 !
;   873          1769  2 TMPRAB [RAB$L_RFA0] = .OLD_IHSYM_PTR[IHS$L_GSTVBN];
;   874          1770  2 TMPRAB [RAB$W_RFA4] = 0;
;   875          1771  2 TMPRAB [RAB$B_RAC] = RAB$C_RFA;
;   876          1772  2
;   877          1773  2 ! Initialize buffer addresses.
```

```
;  878          1774  2 !
;  879          1775  2 PAT$GL_NEWRAB[RAB$L_RBF] = .OUT_BUF_PTR;
;  880          1776  2 TMPRAB[RAB$L_UBF] = .OUT_BUF_PTR;
;  881          1777  2
;  882          1778  2 !++
;  883          1779  2 ! Now write out the variable length global symbol records.
;  884          1780  2 ! **** THIS CODE WILL CHANGE WHEN PATCH HANDLES SYMBOLS. ****
;  885          1781  2 ! **** IT WOULD BE VERY EASY TO EXPAND THE GLOBAL SYMBOL TABLE. ****
;  886          1782  2 !--
;  887          1783  2 IF (.OLD_IHSYM_PTR NEQA 0) AND (.OLD_IHSYM_PTR[IHS$W_GSTRECS] NEQ 0) AND
;  888          1784  3    (.OLD_IHSYM_PTR[IHS$L_GSTVBN] GTR 2)
;  889          1785  2 THEN
;  890          1786  3         BEGIN
;  891          1787  3         COUNTER = .OLD_IHSYM_PTR[IHS$W_GSTRECS];                ! COUNT THE RECORDS AS READ
;  892          1788  3         WHILE .COUNTER GTR 0
;  893          1789  3         DO
;  894          1790  4                 BEGIN
;  895          1791  4                 PAT$GL_ERRCODE = $GET(RAB=TMPRAB);
;  896          1792  4                 IF NOT .PAT$GL_ERRCODE
;  897          1793  4                 THEN
;  898          1794  4                     SIGNAL (PAT$_READERR, 1, GETFILDSC(TMPFAB),
;  899          1795  4                             .TMPRAB[RAB$L_STS], .TMPRAB[RAB$L_STV]);
;  900          1796  4                 TMPRAB[RAB$B_RAC] = RAB$C_SEQ;                  ! SET FOR SEQUENTIAL I/O
;  901          1797  4                 PAT$GL_NEWRAB[RAB$W_RSZ] = .TMPRAB[RAB$W_RSZ];
;  902          1798  4                 PAT$GL_ERRCODE = $PUT(RAB=PAT$GL_NEWRAB);
;  903          1799  4                 IF NOT .PAT$GL_ERRCODE
;  904          1800  4                 THEN
;  905          1801  4                         SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS], .PAT$G
;  906          1802  4                 COUNTER = .COUNTER - 1;
;  907          1803  3                 END;
;  908          1804  3
;  909          1805  3         !++
;  910          1806  3         ! NOW WRITE A RECORD TO FILL THE REST OF THE BLOCK WITH A FILL CHARACTER.
;  911          1807  3         ! THE SIZE OF THE FILLER RECORD IS THE NUMBER OF BYTES IN A BLOCK MINUS
;  912          1808  3         ! THE LAST RECORD SIZE, THE LAST RECORD OFFSET INTO THE BLOCK, AND FOUR
;  913          1809  3         ! BYTES FOR THE LAST RECORD LENGTH AND THE FILLER RECORD LENGTH.
;  914          1810  3         !--
;  915          1811  3         PAT$GL_NEWRAB[RAB$W_RSZ] = A_PAGE - .PAT$GL_NEWRAB[RAB$W_RSZ] - .PAT$GL_NEWRAB[RAB$W_RFA4] - 4;
;  916          1812  4         IF (.PAT$GL_NEWRAB[RAB$W_RSZ]  GTR 0) AND (.PAT$GL_NEWRAB[RAB$W_RSZ] LSS A_PAGE)
;  917          1813  3         THEN
;  918          1814  4                 BEGIN
;  919          1815  4                 CH$FILL(FILL_CHAR, .PAT$GL_NEWRAB[RAB$W_RSZ], .OUT_BUF_PTR);
;  920          1816  4                 PAT$GL_ERRCODE = $PUT(RAB=PAT$GL_NEWRAB);
;  921          1817  4                 IF NOT .PAT$GL_ERRCODE
;  922          1818  4                 THEN
;  923          1819  4                 SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS], .PAT$GL_NEWRAB
;  924          1820  4                 END
;  925          1821  2 END;
;  926          1822  2
;  927          1823  2 !++
;  928          1824  2 ! SET THE VBN OF THE PATCH COMMAND TEXT IN THE NEW IMAGE HEADER, TO NEXT BLOCK.
;  929          1825  2 !--
;  930          1826  2 NEW_IHPAT_PTR[IHP$L_PATCOMTXT] = .PAT$GL_NEWRAB[RAB$L_RFA0] + 1,
;  931          1827  2
;  932          1828  2 !++
;  933          1829  2 ! NOW WRITE OUT THE OLD APPENDED PATCH COMMANDS.
;  934          1830  2 ! THEY ARE VARIABLE LENGTH, SEQUENTIAL RECORDS, ENDED BY EOF.
```

```
;  935          1831  2 !--
;  936          1832  2 IF .PAT$GL_IHPPTR[IHP$L_PATCOMTXT] NEQ 0
;  937          1833  2 THEN
;  938          1834  3         BEGIN
;  939          1835  3         TMPRAB[RAB$B_RAC] = RAB$C_RFA;                      ! FIND THE FIRST BY RFA
;  940          1836  3         TMPRAB[RAB$L_RFA0] = .PAT$GL_IHPPTR[IHP$L_PATCOMTXT]; ! SET VBN
;  941          1837  3         TMPRAB[RAB$W_RFA4] = 0;                             ! SET BYTE OFFSET WITHIN BLOCK
;  942          1838  3         REPEAT
;  943          1839  4                 BEGIN
;  944          1840  4                 !++
;  945          1841  4                 ! THIS LOOP READS AND WRITES ALL THE PREVIOUS APPENDED PATCH
;  946          1842  4                 ! COMMANDS AND FINISHES WHEN EOF IS ENCOUNTERED.
;  947          1843  4                 !--
;  948          1844  4                 PAT$GL_ERRCODE = $GET(RAB=TMPRAB);
;  949          1845  4                 IF .PAT$GL_ERRCODE EQL RMS$_EOF
;  950          1846  4                 THEN
;  951          1847  4                         EXITLOOP;
;  952          1848  4                 IF NOT .PAT$GL_ERRCODE
;  953          1849  4                 THEN
;  954          1850  4                         SIGNAL(PAT$_READERR, 1, GETFILDSC(TMPFAB),
;  955          1851  4                          .TMPRAB[RAB$L_STS], .TMPRAB[RAB$L_STV]);
;  956          1852  4                 TMPRAB[RAB$B_RAC] = RAB$C_SEQ;              ! ALL THE REMAINING IS SEQUENTIAL I/O
;  957          1853  4                 PAT$GL_NEWRAB[RAB$W_RSZ] = .TMPRAB[RAB$W_RSZ];
;  958          1854  4                 PAT$GL_ERRCODE = $PUT(RAB=PAT$GL_NEWRAB);
;  959          1855  4                 IF NOT .PAT$GL_ERRCODE
;  960          1856  4                 THEN
;  961          1857  4                         SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS], .PAT$G
;  962          1858  3                 END;
;  963          1859  2         END;
;  964          1860  2
;  965          1861  2 !++
;  966          1862  2 ! NOW APPEND THE PATCH COMMANDS FOR THIS SESSION.
;  967          1863  2 ! THE PATCH COMMANDS ARE STORED AS ASCIC STRINGS IN BLOCKS THAT ARE
;  968          1864  2 ! SINGULARLY LINKED TOGETHER.  THE COMMANDS DO NOT SPAN BLOCK BOUNDARIES.
;  969          1865  2 ! THE LAST COMMAND IN A BLOCK IS FOLLOWED BY A ZERO COUNT.
;  970          1866  2 !--
;  971          1867  2 COM_TXT_PTR = .PAT$GL_TXTLHD;
;  972          1868  2 WHILE .COM_TXT_PTR NEQA 0
;  973          1869  2 DO
;  974          1870  3         BEGIN
;  975          1871  3         COM_PTR = .COM_TXT_PTR + TXT$C_SIZE;                ! POINT TO FIRST COMMAND IN BLOCK
;  976          1872  3         WHILE .COM_PTR[0] NEQ 0
;  977          1873  3         DO
;  978          1874  4                 BEGIN
;  979          1875  4                 PAT$GL_NEWRAB[RAB$W_RSZ] = .COM_PTR[0];     ! SET LENGTH OF COMMAND
;  980          1876  4                 PAT$GL_NEWRAB[RAB$L_RBF] = .COM_PTR + 1;    ! SET ADDRESS OF COMMAND
;  981          1877  4                 PAT$GL_ERRCODE = $PUT(RAB=PAT$GL_NEWRAB);   ! WRITE ONE COMMAND
;  982          1878  4                 IF NOT .PAT$GL_ERRCODE
;  983          1879  4                 THEN
;  984          1880  4                         SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS], .PAT$G
;  985          1881  4                 COM_PTR = CH$PTR(.COM_PTR, .COM_PTR[0] + 1); ! POINT TO NEXT COMMAND
;  986          1882  3                 END;
;  987          1883  3         COM_TXT_PTR = .COM_TXT_PTR[TXT$L_NXTBLK];           ! POINT TO NEXT COMMAND TEXT BLOCK
;  988          1884  2         END;
```

```
;  990          1885  2
;  991          1886  2 !++
;  992          1887  2 ! NOW WRITE OUT THE IMAGE HEADER.  THE IMAGE HEADER IS WRITTEN LAST BECAUSE
;  993          1888  2 ! THE VBN FOR THE PATCH COMMAND TEXT MUST BE FOUND FIRST.  THE ECO LEVEL BIT
;  994          1889  2 ! CORRESPONDING TO THIS PATCH MAY NOW BE SET IN THE IMAGE HEADER.  IF THE
;  995          1890  2 ! HEADER IS WRITTEN SUCCESSFULLY, THEN THE ECO LEVEL INDICATOR, PAT$GB_ECOLVL,
;  996          1891  2 ! IS RE-INITIALIZED TO ZERO.  THIS WILL ENABLE ANOTHER "SET ECO" COMMAND TO
;  997          1892  2 ! SPECIFY A NEW PATCH.
;  998          1893  2 !--
;  999          1894  3 IF (.PAT$GB_ECOLVL NEQ 0)                                   ! DON'T TRY TO SET A LEVEL IF NONE SPECIFIED
; 1000          1895  2 THEN
; 1001          1896  3         BEGIN
; 1002          1897  3         ECO_LEVEL_PTR = CH$PTR(NEW_IHPAT_PTR[IHP$L_ECO1], 0);   ! SET POINTER TO ECO LEVEL BITVECTOR
; 1003          1898  3         ECO_LEVEL_PTR[.PAT$GB_ECOLVL-1] = 1;                    ! SET ECO BIT
; 1004          1899  3         ECO_LEVEL_PTR = CH$PTR(PAT$GL_IHPPTR[IHP$L_ECO1], 0);   ! SET POINTER TO ECO LEVEL BITVECTOR
; 1005          1900  3         ECO_LEVEL_PTR[.PAT$GB_ECOLVL-1] = 1;                    ! SET ECO BIT
; 1006          1901  2         END;
; 1007          1902  2 PAT$GL_NEWRAB[RAB$L_BKT] = 1;                               ! SET NUMBER OF BLOCK TO OUTPUT
; 1008          1903  2 PAT$GL_NEWRAB[RAB$W_RSZ] = .NUM_HDR_BLKS * A_PAGE;          ! SET NUMBER OF BYTES TO WRITE
; 1009          1904  2 PAT$GL_NEWRAB[RAB$L_RBF] = .NEW_IHD_PTR;                    ! SET BUFFER ADDRESS
; 1010          1905  2
; 1011          1906  2 !++
; 1012          1907  2 ! NOW CLEAR THE TRUNCATE BIT BEFORE PATCH WRITES THE IMAGE HEADER.  IF THIS
; 1013          1908  2 ! IS NOT DONE, THE REST OF THE FILE IS LOST.
; 1014          1909  2 !--
; 1015          1910  2 PAT$GL_NEWRAB[RAB$V_TPT] = FALSE;
; 1016          1911  2 PAT$GL_ERRCODE=$WRITE(RAB=PAT$GL_NEWRAB);                   ! OUTPUT HEADER BLOCKS
; 1017          1912  2 IF NOT .PAT$GL_ERRCODE
; 1018          1913  2 THEN
; 1019          1914  2         SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWRAB[RAB$L_STS], .PAT$GL_NEWRAB[RAB$L_S
; 1020          1915  2 PAT$GB_ECOLVL = 0;                                          ! ALLOW NEW PATCH ECO LEVEL
; 1021          1916  2
; 1022          1917  2 !++
; 1023          1918  2 ! NOW RESET FILE ATTRIBUTES.
; 1024          1919  2 !--
; 1025          1920  2 PAT$GL_NEWFAB[FAB$V_ESC] = TRUE;
; 1026          1921  2 PAT$GL_NEWFAB[FAB$L_CTX] = RME$C_SETRFM;                    ! SET MODIFY CODE
; 1027          1922  2 PAT$GL_NEWFAB[FAB$B_RFM] = FAB$C_FIX;                       ! SET VARIABLE LENGTH RECORDS
; 1028          1923  2 PAT$GL_ERRCODE = $MODIFY(FAB=PAT$GL_NEWFAB);
; 1029          1924  2 IF NOT .PAT$GL_ERRCODE
; 1030          1925  2 THEN
; 1031          1926  2         SIGNAL(PAT$_MODIFYERR, 3, .PAT$GL_ERRCODE, .PAT$GL_NEWNBK[NAM$B_RSL],
; 1032          1927  2                 PAT$GB_NEWNAME, .PAT$GL_ERRCODE, .PAT$GL_NEWRAB[RAB$L_STV]);
; 1033          1928  2
; 1034          1929  2 !++
; 1035          1930  2 ! NOW CLOSE THE OUTPUT IMAGE FILE.  THIS IS DONE HERE SO THAT THE "UPDATE"
; 1036          1931  2 ! COMMAND CAN REWRITE THE FILE IF UPDATE IS SPECIFIED MORE THAN ONCE.
; 1037          1932  2 !--
; 1038          1933  2 PAT$GL_ERRCODE = $CLOSE(FAB=PAT$GL_NEWFAB);
; 1039          1934  2 IF NOT .PAT$GL_ERRCODE
; 1040          1935  2 THEN
; 1041          1936  2         SIGNAL(PAT$_CLOSEOUT, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWFAB[FAB$L_STS], .PAT$GL_NEWFAB[FAB$L_S
; 1042          1937  2 ELSE
; 1043          1938  2         PAT$GL_FLAGS [PAT$S_OUTPUT] = 0;
; 1044          1939  2
; 1045          1940  2 ! Close the input file, we are all done with record I/O processing.
; 1046          1941  2 !
```

```
; 1047        1942  2
; 1048        1943  2 PAT$GL_ERRCODE = $CLOSE (FAB=TMPFAB);
; 1049        1944  2 IF NOT .PAT$GL_ERRCODE
; 1050        1945  2 THEN
; 1051        1946  2     SIGNAL (PAT$_CLOSEIN, 1, GETFILDSC(TMPFAB), .TMPFAB[FAB$L_STS], .TMPFAB[FAB$L_STV]);
; 1052        1947  2
; 1053        1948  2 RETURN;
; 1054        1949  1 END;                                            ! END OF PAT$WRTIMG


                                      .TITLE  PATWRT
                                      .IDENT  \V04-000\

                                      .PSECT  _PAT$PLIT,NOWRT,NOEXE,0

              03  00000 P.AAA:        .BYTE   3
              50  00001               .BYTE   80
            0000  00002               .WORD   0
        00000000  00004               .LONG   0
        00000000  00008               .LONG   0
        00000000  0000C               .LONG   0
        00000000  00010               .LONG   0
            0000  00014               .WORD   0
              02  00016               .BYTE   2
              00  00017               .BYTE   0
        00000000  00018               .LONG   0
              00  0001C               .BYTE   0
              00  0001D               .BYTE   0
              00  0001E               .BYTE   0
              02  0001F               .BYTE   2
        00000000  00020               .LONG   0
        00000000  00024               .LONG   0
       00000000G  00028               .ADDRESS PAT$GL_OLDNBK
        00000000  0002C               .LONG   0
        00000000  00030               .LONG   0
              00  00034               .BYTE   0
              00  00035               .BYTE   0
            0000  00036               .WORD   0
        00000000  00038               .LONG   0
            0000  0003C               .WORD   0
              00  0003E               .BYTE   0
              00  0003F               .BYTE   0
        00000000  00040               .LONG   0
        00000000  00044               .LONG   0
            0000  00048               .WORD   0
              00  0004A               .BYTE   0
              00  0004B               .BYTE   0
        00000000  0004C               .LONG   0
              01  00050 P.AAB:        .BYTE   1
              44  00051               .BYTE   68
            0000  00052               .WORD   0
        00000000  00054               .LONG   0
        00000000  00058               .LONG   0
        00000000  0005C               .LONG   0
           0000#  00060               .WORD   0[3]
            0000  00066               .WORD   0
        00000000  00068               .LONG   0
```

```
                0000  0006C          .WORD   0                                                       ;
                  00  0006E          .BYTE   0                                                       ;
                  00  0006F          .BYTE   0                                                       ;
                0200  00070          .WORD   512                                                     ;
                0200  00072          .WORD   512                                                     ;
00000000G       00074          .ADDRESS PAT$GB_INPBUF                                                ;
00000000G       00078          .ADDRESS PAT$GB_INPBUF                                                ;
00000000        0007C          .LONG   0                                                       ;
00000000        00080          .LONG   0                                                       ;
                  00  00084          .BYTE   0                                                       ;
                  00  00085          .BYTE   0                                                       ;
                  00  00086          .BYTE   0                                                       ;
                  00  00087          .BYTE   0                                                       ;
00000000        00088          .LONG   0                                                       ;
00000000        0008C          .LONG   0                                                       ;
00000000        00090          .LONG   0                                                       ;

                               .PSECT  _PAT$OWN,NOEXE,2

                00000 NUM_OF_UPDATES:
                               .BLKB   4

                      ISE$C_SIZE==        20
                      TXT$C_SIZE==        4
                      PAL$C_SIZE==        16
                      ASD$C_SIZE==        9
                      FWR$C_SIZE==        24
                               .EXTRN  PAT$CREMAP, PAT$WRITEFILE
                               .EXTRN  PAT$FAO_PUT, PAT$ALLOBLK
                               .EXTRN  GETFILDSC, IMG$DECODE_IHD
                               .EXTRN  IMG$GET_NEXT_ISD
                               .EXTRN  PAT$GB_ECOLVC, PAT$GL_CHANUM
                               .EXTRN  PAT$GL_JNLRAB, PAT$GL_NEWVBNMX
                               .EXTRN  PAT$GL_OLDVBNMX
                               .EXTRN  PAT$GL_IMGBLKS, PAT$GW_IMGVOL
                               .EXTRN  PAT$GL_NEWXABALL
                               .EXTRN  PAT$GL_NEWRAB, PAT$GL_NEWFAB
                               .EXTRN  PAT$GL_NEWNBK, PAT$GB_NEWNAME
                               .EXTRN  PAT$GL_OLDRAB, PAT$GL_OLDFAB
                               .EXTRN  PAT$GL_OLDNBK, PAT$GB_OLDNAME
                               .EXTRN  PAT$GB_INPBUF, PAT$GL_IMGHDR
                               .EXTRN  PAT$GL_IHPPTR, PAT$GL_ISELHD
                               .EXTRN  PAT$GL_ISETAIL, PAT$GL_TXTLHD
                               .EXTRN  PAT$GL_FLAGS, PAT$GL_ERRCODE
                               .EXTRN  PAT$GW_IMGTYP, PAT$_CLOSEIN
                               .EXTRN  PAT$_CLOSEOUT, PAT$_OPENIN
                               .EXTRN  PAT$_OPENOUT, PAT$_OVERLAY
                               .EXTRN  PAT$_READERR, PAT$_SYSERROR
                               .EXTRN  PAT$_WRITEERR, SYS$CREATE
                               .EXTRN  SYS$OPEN, SYS$CONNECT
                               .EXTRN  SYS$MODIFY, SYS$WRITE
                               .EXTRN  SYS$READ, SYS$GET
                               .EXTRN  SYS$PUT, SYS$CLOSE

                               .PSECT  _PAT$CODE,NOWRT,2

                0FFC 00000     .ENTRY  PAT$WRTIMG, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 1091
```

```
                                                              R10,R11                                  ;
                    5E        FF54  CE  9E 00002            MOVAB   -172(SP), SP                       ;
  5C  AE 00000000'  EF        0050  8F  28 00007            MOVC3   #80, P.AAA, TMPFAB                 ; 1172
  18  AE 00000000'  EF        0044  8F  28 00012            MOVC3   #68, P.AAB, TMPRAB                 ; 1178
             54     AE          5C  AE  9E 0001D            MOVAB   TMPFAB, TMPRAB+60                  ; 1172
      08 00000000G  EF              06  E1 00022            BBC     #6, PAT$GL_FLAGS, 1$               ; 1202
         00000000V  EF              00  FB 0002A            CALLS   #0, WRITE_BINARY                   ; 1205
                                        04 00031            RET                                        ; 1204
                    50 00000000G    EF  D0 00032 1$:        MOVL    PAT$GL_IMGHDR, R0                  ; 1212
                                04  A0  B5 00039            TSTW    4(R0)                              ;
                                09  13 0003C                BEQL    2$                                 ;
                    5B          04  A0  3C 0003E            MOVZWL  4(R0), OLD_IHSYM_PTR               ; 1214
                    5B              50  C0 00042            ADDL2   R0, OLD_IHSYM_PTR                  ;
                                    02  11 00045            BRB     3$                                 ;
                                    5B  D4 00047 2$:        CLRL    OLD_IHSYM_PTR                      ; 1216
                    50          10  A0  9A 00049 3$:        MOVZBL  16(R0), R0                         ; 1217
      50            50              09  78 0004D            ASHL    #9, R0, R0                         ;
                    50        0400  C0  9E 00051            MOVAB   1024(R0), NEW_IHD_MAX              ;
                                10  AE  9F 00056            PUSHAB  NEW_IHD_PTR                        ; 1218
                                    50  DD 00059            PUSHL   NEW_IHD_MAX                        ;
         00000000G  EF              02  FB 0005B            CALLS   #2, PAT$ALLOBLK                    ;
                                14  AE  9F 00062            PUSHAB  OUT_BUF_PTR                        ; 1219
                    7E        1400  8F  3C 00065            MOVZWL  #5120, -(SP)                       ;
         00000000G  EF              02  FB 0006A            CALLS   #2, PAT$ALLOBLK                    ;
                    56          10  AE  D0 00071            MOVL    NEW_IHD_PTR, R6                    ; 1220
                    53              56  D0 00075            MOVL    R6, NXT_BYTE_PTR                   ;
      63 00000000G  FF 00000000G    FF  28 00078            MOVC3   @PAT$GL_IMGHDR, @PAT$GL_IMGHDR, -  ; 1226
                                                                    (NXT_BYTE_PTR)                     ;
                    59 00000000G    FF  3C 00084            MOVZWL  @PAT$GL_IMGHDR, REM_BYTE_SIZ       ; 1227
      59 00000200   8F              59  C3 0008B            SUBL3   REM_BYTE_SIZ, #512, REM_BYTE_SIZ   ;
                                04  A6  B5 00093            TSTW    4(R6)                              ; 1228
                                    09  13 00096            BEQL    4$                                 ;
                    50          04  A6  3C 00098            MOVZWL  4(R6), R0                          ; 1230
                    56              50  C0 0009C            ADDL2   R0, NEW_IHSYM_PTR                  ;
                                    03  11 0009F            BRB     5$                                 ;
                                10  AE  D4 000A1 4$:        CLRL    NEW_IHD_PTR                        ; 1232
      51       10   AE              08  C1 000A4 5$:        ADDL3   #8, NEW_IHD_PTR, R1                ; 1233
                    50              61  3C 000A9            MOVZWL  (R1), R0                           ;
                    6E       10 BE40    9E 000AC            MOVAB   @NEW_IHD_PTR[R0], NEW_IHPAT_PTR    ;
               0C   AE              01  D0 000B1            MOVL    #1, NUM_HDR_BLKS                   ; 1238
                    58 00000000G    EF  D0 000B5            MOVL    PAT$GL_ISELHD, ISE_PTR             ; 1239
                    59              02  C2 000BC            SUBL2   #2, REM_BYTE_SIZ                   ; 1240
                                    58  D5 000BF 6$:        TSTL    ISE_PTR                            ; 1241
                                    6C  13 000C1            BEQL    12$                                ;
                    50 00000000G    EF  D0 000C3            MOVL    PAT$GL_IMGHDR, R0                  ; 1244
                    50          10  A0  9A 000CA            MOVZBL  16(R0), R0                         ;
                    50              02  C0 000CE            ADDL2   #2, R0                             ;
                    50          0C  AE  D1 000D1            CMPL    NUM_HDR_BLKS, R0                   ;
                                    0D  15 000D5            BLEQ    7$                                 ;
                         006D817C   8F  DD 000D7            PUSHL   #7176572                           ; 1246
         00000000G  00              01  FB 000DD            CALLS   #1, LIB$SIGNAL                     ;
                    57          14  A8  9E 000E4 7$:        MOVAB   20(R8), ISD_PTR                    ; 1247
                                    68  D5 000E8            TSTL    (ISE_PTR)                          ; 1249
                                    09  12 000EA            BNEQ    8$                                 ;
                    5A              67  3C 000EC            MOVZWL  (ISD_PTR), R10                     ; 1250
                    50          02  AA  9E 000EF            MOVAB   2(R10), R0                         ;
                                    06  11 000F3            BRB     9$                                 ;
```

```
                          5A           67  3C 000F5 8$:     MOVZWL  (ISD_PTR), R10                                ; 1251
                          50           5A  D0 000F8         MOVL    R10, R0                                       ;
                          50           59  D1 000FB 9$:     CMPL    REM_BYTE_SIZ, R0                              ; 1249
                                       0C  15 000FE         BLEQ    10$                                           ;
              63          67           5A  28 00100         MOVC3   R10, (ISD_PTR), (NXT_BYTE_PTR)                ; 1254
                          59           5A  C2 00104         SUBL2   R10, REM_BYTE_SIZ                             ; 1255
                          58           68  D0 00107         MOVL    (ISE_PTR), ISE_PTR                            ; 1256
                                       B3  11 0010A         BRB     6$                                            ; 1248
                          50       02  A9  9E 0010C 10$:    MOVAB   2(R9), R0                                     ; 1260
    50        FF      8F           6E  00  2C 00110         MOVC5   #0, (SP), #255, R0, (NXT_BYTE_PTR)            ;
                                       63     00116                                                           ;
                          01       0C  AE  D1 00117         CMPL    NUM_HDR_BLKS, #1                              ; 1261
                                       08  12 0011B         BNEQ    11$                                           ;
                      FE  A3 00000000G EF  3C 0011D         MOVZWL  PAT$GW_IMGTYP, -2(NXT_BYTE_PTR)               ; 1262
                                   0C  AE  D6 00125 11$:    INCL    NUM_HDR_BLKS                                  ; 1263
                          59     01FE  8F  3C 00128         MOVZWL  #510, REM_BYTE_SIZ                            ; 1264
                                       90  11 0012D         BRB     6$                                            ; 1241
                                       83  B4 0012F 12$:    CLRW    (NXT_BYTE_PTR)+                               ; 1267
    59        FF      8F           6E  00  2C 00131         MOVC5   #0, (SP), #255, REM_BYTE_SIZ, -               ; 1268
                                       63     00137                 (NXT_BYTE_PTR)                                ;
                          01       0C  AE  D1 00138         CMPL    NUM_HDR_BLKS, #1                              ; 1269
                                       08  12 0013C         BNEQ    13$                                           ;
                      FE  A3 00000000G EF  3C 0013E         MOVZWL  PAT$GW_IMGTYP, -2(NXT_BYTE_PTR)               ; 1270
                          50 00000000G EF  D0 00146 13$:    MOVL    PAT$GL_IMGHDR, R0                             ; 1275
 0C   AE          10  A0           08  00  ED 0014D         CMPZV   #0, #8, 16(R0), NUM_HDR_BLKS                  ;
                                       10  12 00154         BNEQ    14$                                           ;
                  00000000G EF 00000000G EF  D1 00156       CMPL    PAT$GL_NEWVBNMX, PAT$GL_OLDVBNMX              ; 1276
                                       03  12 00161         BNEQ    14$                                           ;
                                     009F  31 00163         BRW     21$                                           ;
              50          10  AE           10  C1 00166 14$: ADDL3  #16, NEW_IHD_PTR, R0                          ; 1284
                          54           60  9A 0016B         MOVZBL  (R0), BLK_DIFF                                ;
              54          0C  AE           54  C3 0016E     SUBL3   BLK_DIFF, NUM_HDR_BLKS, BLK_DIFF              ;
                          52       10  BE  3C 00173         MOVZWL  @NEW_IHD_PTR, NEW_ISD_PTR                     ; 1285
                          52       10  AE  C0 00177         ADDL2   NEW_IHD_PTR, NEW_ISD_PTR                      ;
                                       62  B5 0017B 15$:    TSTW    (NEW_ISD_PTR)                                 ; 1286
                                       3F  13 0017D         BEQL    17$                                           ;
              08          08  A2           02  E0 0017F     BBS     #2, 8(NEW_ISD_PTR), 16$                       ; 1289
                          04       08  A2  E8 00184         BLBS    8(NEW_ISD_PTR), 16$                           ;
                          0C  A2           54  C0 00188     ADDL2   BLK_DIFF, 12(NEW_ISD_PTR)                     ; 1291
                          50           62  3C 0018C 16$:    MOVZWL  (NEW_ISD_PTR), R0                             ; 1292
                          52           50  C0 0018F         ADDL2   R0, NEW_ISD_PTR                               ;
                    FFFF  8F           62  B1 00192         CMPW    (NEW_ISD_PTR), #65535                         ; 1293
                                       E2  12 00197         BNEQ    15$                                           ;
                          50     01FF  C2  9E 00199         MOVAB   511(R2), R0                                   ; 1299
                          50 00000200  8F  C6 0019E         DIVL2   #512, R0                                      ;
              50          50           09  78 001A5         ASHL    #9, R0, R0                                    ;
        7E    00          10  AE           01  7A 001A9     EMUL    #1, NEW_IHD_PTR, #0, -(SP)                    ; 1300
        51    51          8E 00000200  8F  7B 001AF         EDIV    #512, (SP)+, R1, R1                           ;
              52          50           51  C1 001B8         ADDL3   R1, R0, NEW_ISD_PTR                           ;
                                       BD  11 001BC         BRB     15$                                           ; 1286
              50          10  AE           10  C1 001BE 17$: ADDL3  #16, NEW_IHD_PTR, R0                          ; 1302
                          60       0C  AE  90 001C3         MOVB    NUM_HDR_BLKS, (R0)                            ;
              50          54 00000000G EF  C1 001C7         ADDL3   PAT$GL_NEWVBNMX, BLK_DIFF, R0                 ; 1303
              54          50 00000000G EF  C3 001CF         SUBL3   PAT$GL_OLDVBNMX, R0, BLK_DIFF                 ;
                                       56  D5 001D7         TSTL    NEW_IHSYM_PTR                                 ; 1304
                                       19  13 001D9         BEQL    20$                                           ;
                                       66  D5 001DB         TSTL    (NEW_IHSYM_PTR)                               ; 1307
```

```
                                                03  13 001DD        BEQL    18$                                            ;
                                 66             54  C0 001DF        ADDL2   BLK_DIFF, (NEW_IHSYM_PTR)                      ; 1309
                                          0C    A6  D5 001E2 18$:   TSTL    12(NEW_IHSYM_PTR)                              ; 1311
                                                04  13 001E5        BEQL    19$                                            ;
                           0C    A6             54  C0 001E7        ADDL2   BLK_DIFF, 12(NEW_IHSYM_PTR)                    ; 1313
                                          04    A6  D5 001EB 19$:   TSTL    4(NEW_IHSYM_PTR)                               ; 1315
                                                04  13 001EE        BEQL    20$                                            ;
                           04    A6             54  C0 001F0        ADDL2   BLK_DIFF, 4(NEW_IHSYM_PTR)                     ; 1317
                     50          6E             20  C1 001F4 20$:   ADDL3   #32, NEW_IHPAT_PTR, R0                         ; 1319
                                                60  D5 001F8        TSTL    (R0)                                           ;
                                                0B  13 001FA        BEQL    22$                                            ;
                     50          6E             20  C1 001FC        ADDL3   #32, NEW_IHPAT_PTR, R0                         ; 1321
                                 60             54  C0 00200        ADDL2   BLK_DIFF, (R0)                                 ;
                                                02  11 00203        BRB     22$                                            ; 1275
                                                54  D4 00205 21$:   CLRL    BLK_DIFF                                       ; 1324
                                 5A 00000000G   EF  D0 00207 22$:   MOVL    PAT$GL_TXTLHD, COM_TXT_PTR                     ; 1330
                                                0B  13 0020E 23$:   BEQL    24$                                            ; 1331
                                    00000000G   EF  D6 00210        INCL    PAT$GL_IMGBLKS                                 ; 1334
                                 5A             6A  D0 00216        MOVL    (COM_TXT_PTR), COM_TXT_PTR                     ; 1335
                                                F3  11 00219        BRB     23$                                            ; 1331
                        00000000G EF            54  C0 0021B 24$:   ADDL2   BLK_DIFF, PAT$GL_IMGBLKS                       ; 1337
                        00000000G EF 00000000G  EF  D0 00222        MOVL    PAT$GL_IMGBLKS, PAT$GL_NEWXABALL+16            ; 1338
                     12 00000000G EF            05  E1 0022D        BBC     #5, PAT$GL_FLAGS, 25$                          ; 1340
                        00000000G EF 00000000G  EF  B0 00235        MOVW    PAT$GW_IMGVOL, PAT$GL_NEWXABALL+10             ; 1343
                        00000000G EF            02  90 00240        MOVB    #2, PAT$GL_NEWXABALL+9                         ; 1344
                                    00000000'   EF  D5 00247 25$:   TSTL    NUM_OF_UPDATES                                 ; 1347
                                                78  12 0024D        BNEQ    26$                                            ;
           50 00000000G  EF          01         04  EF 0024F        EXTZV   #4, #1, PAT$GL_OLDFAB+6, R0                    ; 1357
00000000G  EF            01          07         50  F0 00258        INSV    R0, #7, #1, PAT$GL_NEWXABALL+8                 ;
           50 00000000G  EF          01         05  EF 00261        EXTZV   #5, #1, PAT$GL_OLDFAB+6, R0                    ; 1358
00000000G  EF            01          05         50  F0 0026A        INSV    R0, #5, #1, PAT$GL_NEWXABALL+8                 ;
                                    00000000G   EF  9F 00273        PUSHAB  PAT$GL_NEWFAB                                  ; 1359
                        00000000G 00            01  FB 00279        CALLS   #1, SYS$CREATE                                 ;
                        00000000G EF            50  D0 00280        MOVL    R0, PAT$GL_ERRCODE                             ;
                                  51            50  E8 00287        BLBS    R0, 27$                                        ;
                        00000000G EF            20  88 0028A        BISB2   #32, PAT$GL_NEWXABALL+8                        ; 1362
                                    00000000G   EF  9F 00291        PUSHAB  PAT$GL_NEWFAB                                  ; 1363
                        00000000G 00            01  FB 00297        CALLS   #1, SYS$CREATE                                 ;
                        00000000G EF            50  D0 0029E        MOVL    R0, PAT$GL_ERRCODE                             ;
                                 36 00000000G   EF  E9 002A5        BLBC    PAT$GL_ERRCODE, 28$                            ; 1364
                                    00000000G   EF  DD 002AC        PUSHL   PAT$GL_NEWRAB+12                               ; 1366
                                    00000000G   EF  DD 002B2        PUSHL   PAT$GL_ERRCODE                                 ;
                                    006D82A3    8F  DD 002B8        PUSHL   #7176867                                       ;
                        00000000G 00            03  FB 002BE        CALLS   #3, LIB$SIGNAL                                 ;
                                                14  11 002C5        BRB     27$                                            ; 1347
                                    00000000G   EF  9F 002C7 26$:   PUSHAB  PAT$GL_NEWFAB                                  ; 1370
                        00000000G 00            01  FB 002CD        CALLS   #1, SYS$OPEN                                   ;
                        00000000G EF            50  D0 002D4        MOVL    R0, PAT$GL_ERRCODE                             ;
                                 0E 00000000G   EF  E8 002DB 27$:   BLBS    PAT$GL_ERRCODE, 29$                            ; 1371
                                    00000000G   EF  DD 002E2 28$:   PUSHL   PAT$GL_NEWRAB+12                               ; 1373
                                    00000000G   EF  DD 002E8        PUSHL   PAT$GL_NEWFAB+8                                ;
                                                22  11 002EE        BRB     30$                                            ;
                                    00000000G   EF  9F 002F0 29$:   PUSHAB  PAT$GL_NEWRAB                                  ; 1376
                        00000000G 00            01  FB 002F6        CALLS   #1  SYS$CONNECT                                ;
                        00000000G EF            50  D0 002FD        MOVL    R0, PAT$GL_ERRCODE                             ;
                                 27 00000000G   EF  E8 00304        BLBS    PAT$GL_ERRCODE, 31$                            ; 1377
                                 7E 00000000G   EF  7D 0030B        MOVQ    PAT$GL_NEWRAB+8, -(SP)                         ; 1379
```

```
                          00000000G   EF   9F 00312 30$:    PUSHAB  PAT$GL_NEWFAB                          ;
                00000000G EF          01   FB 00318         CALLS   #1, GETFILDSC                         ;
                                      50   DD 0031F         PUSHL   R0                                    ;
                                      01   DD 00321         PUSHL   #1                                    ;
                          00000000G   8F   DD 00323         PUSHL   #PAT$_OPENOUT                         ;
                00000000G 00          05   FB 00329         CALLS   #5, LIB$SIGNAL                        ;
                                      60   11 00330         BRB     32$                                   ;
                00000000' EF          01   D0 00332 31$:    MOVL    #1, NUM_OF_UPDATES                    ; 1382
                00000000G EF          08   88 00339         BISB2   #8, PAT$GL_FLAGS                      ; 1383
                00000000G EF          08   88 00340         BISB2   #8, PAT$GL_NEWFAB+7                   ; 1384
                00000000G EF          01   D0 00347         MOVL    #1, PAT$GL_NEWFAB+24                  ; 1385
                00000000G EF          02   90 0034E         MOVB    #2, PAT$GL_NEWFAB+31                  ; 1386
                          00000000G   EF   9F 00355         PUSHAB  PAT$GL_NEWFAB                         ; 1387
                00000000G 00          01   FB 0035B         CALLS   #1, SYS$MODIFY                        ;
                00000000G EF          50   D0 00362         MOVL    R0, PAT$GL_ERRCODE                    ;
                          26          50   E8 00369         BLBS    R0, 32$                               ; 1388
                          00000000G   EF   DD 0036C         PUSHL   PAT$GL_NEWRAB+12                      ; 1392
                                      50   DD 00372         PUSHL   R0                                    ;
                          00000000G   EF   9F 00374         PUSHAB  PAT$GB_NEWNAME                        ; 1390
                          7E 00000000G EF  9A 0037A         MOVZBL  PAT$GL_NEWNBK+3, -(SP)                ; 1391
                                      50   DD 00381         PUSHL   R0                                    ; 1390
                                      03   DD 00383         PUSHL   #3                                    ;
                          006D819C    8F   DD 00385         PUSHL   #7176604                              ;
                00000000G 00          07   FB 0038B         CALLS   #7, LIB$SIGNAL                        ;
                          00000000G   EF   9F 00392 32$:    PUSHAB  PAT$GB_NEWNAME                        ; 1399
                          7E 00000000G EF  9A 00398         MOVZBL  PAT$GL_NEWNBK+3, -(SP)                ;
                                      02   DD 0039F         PUSHL   #2                                    ;
                          006D82C3    8F   DD 003A1         PUSHL   #7176899                              ;
                00000000G 00          04   FB 003A7         CALLS   #4, LIB$SIGNAL                        ;
             01 00000000G EF          03   E0 003AE         BBS     #3, PAT$GL_FLAGS, 33$                 ; 1404
                                      04   003B6         RET                                           ;
             01 00000000G EF          02   E0 003B7 33$:    BBS     #2, PAT$GL_FLAGS, 34$                 ; 1405
                                      04   003BF         RET                                           ;
                          58 00000000G EF  D0 003C0 34$:    MOVL    PAT$GL_ISELHD, ISE_PTR                ; 1411
                          52       10  BE  3C 003C7         MOVZWL  @NEW_IHD_PTR, NEW_ISD_PTR             ; 1412
                          52       10  AE  C0 003CB         ADDL2   NEW_IHD_PTR, NEW_ISD_PTR              ;
                                      59   D4 003CF         CLRL    MAX_VBN_WRITTEN                       ; 1413
                                      58   D5 003D1 35$:    TSTL    ISE_PTR                               ; 1414
                                      03   12 003D3         BNEQ    36$                                   ;
                                   019E   31 003D5         BRW     54$                                   ;
                          57       14  A8  9E 003D8 36$:    MOVAB   20(R8), ISD_PTR                       ; 1417
                                      53   D4 003DC         CLRL    COUNTER                               ; 1418
             03       08  A7           02   E1 003DE         BBC     #2, 8(ISD_PTR), 38$                   ; 1419
                                   0163   31 003E3 37$:    BRW     52$                                   ;
                      FD  8F       0B  A7  91 003E6 38$:    CMPB    11(ISD_PTR), #253                     ; 1420
                                      F6   13 003EB         BEQL    37$                                   ;
                          05       08  A7  E9 003ED         BLBC    8(ISD_PTR), 39$                       ; 1421
                                   0C  A7  D5 003F1         TSTL    12(ISD_PTR)                           ; 1422
                                      ED   13 003F4         BEQL    37$                                   ;
                      08  AE       0C  A7  D0 003F6 39$:    MOVL    12(ISD_PTR), CUR_VBN                  ; 1428
                00000000G EF       0C  A2  D0 003FB         MOVL    12(NEW_ISD_PTR), PAT$GL_NEWRAB+56     ; 1429
                          55       02  A7  3C 00403 40$:    MOVZWL  2(ISD_PTR), R5                        ; 1435
                          55           53   D1 00407         CMPL    COUNTER, R5                           ;
                                      03   12 0040A         BNEQ    41$                                   ;
                                   012A   31 0040C         BRW     51$                                   ;
                                   10  A8  D5 0040F 41$:    TSTL    16(ISE_PTR)                           ; 1438
                                      03   13 00412         BEQL    42$                                   ;
```

```
                                          0082  31 00414        BRW     46$                                         ;
         50             55                  53  C3 00417 42$:   SUBL3   COUNTER, R5, R0                             ; 1444
         50             50                  09  78 0041B        ASHL    #9, R0, R0                                  ;
            00001400    8F                  50  D1 0041F        CMPL    R0, #5120                                   ;
                                            11  18 00426        BGEQ    43$                                         ;
                  04    AE                  50  D0 00428        MOVL    R0, BYTES_TO_READ                           ; 1450
            00000000G   EF          04      AE  B0 0042C        MOVW    BYTES_TO_READ, PAT$GL_NEWRAB+34             ; 1451
                        53                  55  D0 00434        MOVL    R5, COUNTER                                 ; 1452
                                            12  11 00437        BRB     44$                                         ; 1444
                  04    AE        1400      8F  3C 00439 43$:   MOVZWL  #5120, BYTES_TO_READ                        ; 1460
            00000000G   EF        1400      8F  B0 0043F        MOVW    #5120, PAT$GL_NEWRAB+34                     ; 1461
                        53                  0A  C0 00448        ADDL2   #10, COUNTER                                ; 1462
                                    04      AE  DD 0044B 44$:   PUSHL   BYTES_TO_READ                               ; 1471
                                    18      AE  DD 0044E        PUSHL   OUT_BUF_PTR                                 ; 1470
                                    10      AE  DD 00451        PUSHL   CUR_VBN                                     ; 1469
            00000000V   EF                  03  FB 00454        CALLS   #3, GET_IMAGE_BLOCK                         ;
            00000000G   EF                  50  D0 0045B        MOVL    R0, PAT$GL_ERRCODE                          ;
                        26 00000000G        EF  E8 00462        BLBS    PAT$GL_ERRCODE, 45$                         ; 1472
                                            7E  D4 00469        CLRL    -(SP)                                       ; 1474
                           00000000G        EF  DD 0046B        PUSHL   PAT$GL_ERRCODE                              ; 1475
                           00000000G        EF  9F 00471        PUSHAB  PAT$GL_OLDFAB                               ; 1474
            00000000G   EF                  01  FB 00477        CALLS   #1, GETFILDSC                               ;
                                            50  DD 0047E        PUSHL   R0                                          ;
                                            01  DD 00480        PUSHL   #1                                          ;
                           00000000G        8F  DD 00482        PUSHL   #PAT$_READERR                               ;
            00000000G   00                  05  FB 00488        CALLS   #5, LIB$SIGNAL                              ;
            00000000G   EF          14      AE  D0 0048F 45$:   MOVL    OUT_BUF_PTR, PAT$GL_NEWRAB+40               ; 1480
                                            49  11 00497        BRB     48$                                         ; 1438
         51             53                  09  78 00499 46$:   ASHL    #9, COUNTER, R1                             ; 1501
         50             55                  53  C3 0049D        SUBL3   COUNTER, R5, R0                             ; 1489
         50             50                  09  78 004A1        ASHL    #9, R0, R0                                  ;
            00001400    8F                  50  D1 004A5        CMPL    R0, #5120                                   ;
                                            1F  18 004AC        BGEQ    47$                                         ;
                        50          02      A2  3C 004AE        MOVZWL  2(NEW_ISD_PTR), R0                          ; 1499
                        50                  53  C2 004B2        SUBL2   COUNTER, R0                                 ;
00000000G  EF           50        0200      8F  A5 004B5        MULW3   #512, R0, PAT$GL_NEWRAB+34                  ; 1498
            00000000G   EF          0C    B841  9E 004BF        MOVAB   @12(ISE_PTR)[R1], PAT$GL_NEWRAB+40          ; 1501
                        53                  55  D0 004C8        MOVL    R5, COUNTER                                 ; 1502
                                            15  11 004CB        BRB     48$                                         ; 1489
            00000000G   EF        1400      8F  B0 004CD 47$:   MOVW    #5120, PAT$GL_NEWRAB+34                     ; 1511
            00000000G   EF          0C    B841  9E 004D6        MOVAB   @12(ISE_PTR)[R1], PAT$GL_NEWRAB+40          ; 1513
                        53                  0A  C0 004DF        ADDL2   #10, COUNTER                                ; 1514
                           00000000G        EF  9F 004E2 48$:   PUSHAB  PAT$GL_NEWRAB                               ; 1521
            00000000G   00                  01  FB 004E8        CALLS   #1, SYS$WRITE                               ;
            00000000G   EF                  50  D0 004EF        MOVL    R0, PAT$GL_ERRCODE                          ;
                        25 00000000G        EF  E8 004F6        BLBS    PAT$GL_ERRCODE, 49$                         ; 1522
                        7E 00000000G        EF  7D 004FD        MOVQ    PAT$GL_NEWRAB+8, -(SP)                      ; 1525
                           00000000G        EF  9F 00504        PUSHAB  PAT$GL_NEWFAB                               ; 1524
            00000000G   EF                  01  FB 0050A        CALLS   #1, GETFILDSC                               ;
                                            50  DD 00511        PUSHL   R0                                          ;
                                            01  DD 00513        PUSHL   #1                                          ;
                           00000000G        8F  DD 00515        PUSHL   #PAT$_WRITEERR                              ;
            00000000G   00                  05  FB 0051B        CALLS   #5, LIB$SIGNAL                              ;
                        55                  53  D1 00522 49$:   CMPL    COUNTER, R5                                 ; 1532
                                            0F  13 00525        BEQL    50$                                         ;
                  08    AE          0C    B743  9E 00527        MOVAB   @12(ISD_PTR)[COUNTER], CUR_VBN              ; 1535
            00000000G   EF          0C    B243  9E 0052D        MOVAB   @12(NEW_ISD_PTR)[COUNTER], PAT$GL_NEWRAB+56 ; 1536
```

```
                          FECA   31 00536 50$:     BRW     40$                                             ; 1435
                    50      02   A2  3C 00539 51$:     MOVZWL  2(NEW_ISD_PTR), R0                          ; 1542
                    50      0C   A2  C0 0053D          ADDL2   12(NEW_ISD_PTR), R0                         ;
                    50           59  D1 00541          CMPL    MAX_VBN_WRITTEN, R0                         ;
                                 03  1E 00544          BGEQU   52$                                         ;
                    59           50  D0 00546          MOVL    R0, MAX_VBN_WRITTEN                         ; 1544
                    58           68  D0 00549 52$:     MOVL    (ISE_PTR), ISE_PTR                          ; 1546
                    50           62  3C 0054C          MOVZWL  (NEW_ISD_PTR), R0                           ; 1547
                    52           50  C0 0054F          ADDL2   R0, NEW_ISD_PTR                             ;
            FFFF    8F           62  B1 00552          CMPW    (NEW_ISD_PTR), #65535                       ; 1551
                                 1A  12 00557          BNEQ    53$                                         ;
          50        52      10   AE  C3 00559          SUBL3   NEW_IHD_PTR, NEW_ISD_PTR, R0                ; 1554
                    50    01FF   C0  9E 0055E          MOVAB   511(R0), R0                                 ;
                    50 00000200  8F  C6 00563          DIVL2   #512, R0                                    ;
          50        50           09  78 0056A          ASHL    #9, R0, R0                                  ;
          52        50      10   AE  C1 0056E          ADDL3   NEW_IHD_PTR, R0, NEW_ISD_PTR                ;
                          FE5B   31 00573 53$:     BRW     35$                                             ; 1414
    00000000G   EF           02  88 00576 54$:     BISB2   #2, PAT$GL_NEWRAB+4                             ; 1561
                             52  D4 0057D          CLRL    R2                                              ; 1567
                             5B  D5 0057F          TSTL    OLD_IHSYM_PTR                                   ;
                             03  12 00581          BNEQ    55$                                             ;
                           0222  31 00583          BRW     76$                                             ;
                             52  D6 00586 55$:     INCL    R2                                              ;
                    55       08  AB  3C 00588          MOVZWL  8(OLD_IHSYM_PTR), R5                        ; 1570
                                 05  13 0058C          BEQL    56$                                         ;
                    02           6B  D1 0058E          CMPL    (OLD_IHSYM_PTR), #2                         ;
                                 03  14 00591          BGTR    57$                                         ;
                           00EA  31 00593 56$:     BRW     64$                                             ;
                             53  D4 00596 57$:     CLRL    COUNTER                                         ; 1573
                    66           59  D0 00598          MOVL    MAX_VBN_WRITTEN, (NEW_IHSYM_PTR)            ; 1574
    00000000G   EF               59  D0 0059B          MOVL    MAX_VBN_WRITTEN, PAT$GL_NEWRAB+56           ; 1575
            08  AE               6B  D0 005A2          MOVL    (OLD_IHSYM_PTR), CUR_VBN                    ; 1576
                    55           53  D1 005A6 58$:     CMPL    COUNTER, R5                                 ; 1577
                                 E8  13 005A9          BEQL    56$                                         ;
          50        55           53  C3 005AB          SUBL3   COUNTER, R5, R0                             ; 1583
          50        50           09  78 005AF          ASHL    #9, R0, R0                                  ;
    00001400    8F               50  D1 005B3          CMPL    R0, #5120                                   ;
                                 11  18 005BA          BGEQ    59$                                         ;
                04  AE           50  D0 005BC          MOVL    R0, BYTES_TO_READ                           ; 1589
    00000000G   EF      04   AE  B0  005C0          MOVW    BYTES_TO_READ, PAT$GL_NEWRAB+34             ; 1590
                    53           55  D0 005C8          MOVL    R5, COUNTER                                 ; 1591
                                 12  11 005CB          BRB     60$                                         ; 1583
                04  AE     1400  8F  3C 005CD 59$:     MOVZWL  #5120, BYTES_TO_READ                        ; 1599
    00000000G   EF      1400  8F  B0 005D3          MOVW    #5120, PAT$GL_NEWRAB+34                     ; 1600
                    53           0A  C0 005DC          ADDL2   #10, COUNTER                                ; 1601
                          04   AE  DD 005DF 60$:     PUSHL   BYTES_TO_READ                               ; 1610
                          18   AE  DD 005E2          PUSHL   OUT_BUF_PTR                                 ; 1609
                          10   AE  DD 005E5          PUSHL   CUR_VBN                                     ; 1608
    00000000V   EF               03  FB 005E8          CALLS   #3, GET_IMAGE_BLOCK                         ;
    00000000G   EF               50  D0 005EF          MOVL    R0, PAT$GL_ERRCODE                          ;
                26 00000000G     EF  E8 005F6          BLBS    PAT$GL_ERRCODE, 61$                         ; 1611
                             7E  D4  005FD          CLRL    -(SP)                                           ; 1613
                   00000000G     EF  DD 005FF          PUSHL   PAT$GL_ERRCODE                              ;
                   00000000G     EF  9F 00605          PUSHAB  PAT$GL_OLDFAB                               ;
    00000000G   EF               01  FB 0060B          CALLS   #1, GETFILDSC                               ;
                                 50  DD 00612          PUSHL   R0                                          ;
                                 01  DD 00614          PUSHL   #1                                          ;
```

```
                                00000000G   8F   DD 00616        PUSHL   #PAT$_READERR                              :
                     00000000G  00          05   FB 0061C        CALLS   #5, LIB$SIGNAL                             :
                     00000000G  EF     14   AE   D0 00623 61$:   MOVL    OUT_BUF_PTR, PAT$GL_NEWRAB+40              : 1618
                                00000000G   9F   9F 0062B        PUSHAB  PAT$GL_NEWRAB                              : 1623
                     00000000G  00          01   FB 00631        CALLS   #1, SYS$WRITE                              :
                     00000000G  EF          50   D0 00638        MOVL    R0, PAT$GL_ERRCODE                         :
                             25 00000000G   EF   E8 0063F        BLBS    PAT$GL_ERRCODE, 62$                        : 1624
                             7E 00000000G   EF   7D 00646        MOVQ    PAT$GL_NEWRAB+8, -(SP)                     : 1626
                                00000000G   EF   9F 0064D        PUSHAB  PAT$GL_NEWFAB                              :
                     00000000G  EF          01   FB 00653        CALLS   #1, GETFILDSC                              :
                                            50   DD 0065A        PUSHL   R0                                         :
                                            01   DD 0065C        PUSHL   #1                                         :
                                00000000G   8F   DD 0065E        PUSHL   #PAT$_WRITEERR                             :
                     00000000G  00          05   FB 00664        CALLS   #5, LIB$SIGNAL                             :
                                55          53   D1 0066B 62$:   CMPL    COUNTER, R5                                : 1632
                                            0D   13 0066E        BEQL    63$                                        :
       08  AE                   6B          53   C1 00670        ADDL3   COUNTER, (OLD_IHSYM_PTR), CUR_VBN          : 1635
00000000G  EF                   66          53   C1 00675        ADDL3   COUNTER, (NEW_IHSYM_PTR), PAT$GL_NEWRAB+56 : 1636
                                          FF26   31 0067D 63$:   BRW     58$                                        : 1577
                             50        08   A6   3C 00680 64$:   MOVZWL  8(NEW_IHSYM_PTR), R0                       : 1640
                             50             66   C0 00684        ADDL2   (NEW_IHSYM_PTR), R0                        :
                             50             59   D1 00687        CMPL    MAX_VBN_WRITTEN, R0                        :
                                            03   1E 0068A        BGEQU   65$                                        :
                             59             50   D0 0068C        MOVL    R0, MAX_VBN_WRITTEN                        : 1642
                             50 00000000G   EF   D0 0068F 65$:   MOVL    PAT$GL_IMGHDR, R0                          : 1649
                  03       20   A0          05   E0 00696        BBS     #5, 32(R0), 67$                            :
                                          010A   31 0069B 66$:   BRW     76$                                        :
                             55        10   AB   D0 0069E 67$:   MOVL    16(OLD_IHSYM_PTR), R5                      : 1651
                                            F7   13 006A2        BEQL    66$                                        :
                             50      0200   C5   9E 006A4        MOVAB   512(R5), R0                                : 1654
                  54         50 00000200    8F   C7 006A9        DIVL3   #512, R0, BLK_DIFF                         :
                                            53   D4 006B1        CLRL    COUNTER                                    : 1655
                     0C  A6                 59   D0 006B3        MOVL    MAX_VBN_WRITTEN, 12(NEW_IHSYM_PTR)         : 1656
              00000000G  EF                 59   D0 006B7        MOVL    MAX_VBN_WRITTEN, PAT$GL_NEWRAB+56          : 1657
                     08  AE        0C       AB   D0 006BE        MOVL    12(OLD_IHSYM_PTR), CUR_VBN                 : 1658
                             54             53   D1 006C3 68$:   CMPL    COUNTER, BLK_DIFF                          : 1660
                                            03   19 006C6        BLSS    69$                                        :
                                          00D0   31 006C8        BRW     75$                                        :
                  51         53             09   78 006CB 69$:   ASHL    #9, COUNTER, R1                            : 1663
                             50      1400   C1   9E 006CF        MOVAB   5120(R1), R0                               : 1662
                             50             55   D1 006D4        CMPL    R5, R0                                     :
                                            0E   14 006D7        BGTR    70$                                        :
                  50         54             53   C3 006D9        SUBL3   COUNTER, BLK_DIFF, R0                      : 1666
       04  AE                50             09   78 006DD        ASHL    #9, R0, BYTES_TO_READ                      :
                             53             50   C0 006E2        ADDL2   R0, COUNTER                                : 1667
                                            09   11 006E5        BRB     71$                                        : 1662
                     04  AE          1400   8F   3C 006E7 70$:   MOVZWL  #5120, BYTES_TO_READ                       : 1671
                             53             0A   C0 006ED        ADDL2   #10, COUNTER                               : 1672
                     00000000G  EF     04   AE   B0 006F0 71$:   MOVW    BYTES_TO_READ, PAT$GL_NEWRAB+34            : 1675
                                       04   AE   DD 006F8        PUSHL   BYTES_TO_READ                              : 1683
                                       18   AE   DD 006FB        PUSHL   OUT_BUF_PTR                                : 1682
                                       10   AE   DD 006FE        PUSHL   CUR_VBN                                    : 1681
                     00000000V  EF          03   FB 00701        CALLS   #3, GET_IMAGE_BLOCK                        :
                     00000000G  EF          50   D0 00708        MOVL    R0, PAT$GL_ERRCODE                         :
                             26 00000000G   EF   E8 0070F        BLBS    PAT$GL_ERRCODE, 72$                        : 1684
                                            7E   D4 00716        CLRL    -(SP)                                      : 1686
                                00000000G   EF   DD 00718        PUSHL   PAT$GL_ERRCODE                             :
```

```
                    00000000G   EF   9F 0071E                PUSHAB  PAT$GL_OLDFAB                                   ;
00000000G   EF                  01   FB 00724                CALLS   #1, GETFILDSC                                   ;
                                50   DD 0072B                PUSHL   R0                                              ;
                                01   DD 0072D                PUSHL   #1                                              ;
                    00000000G   8F   DD 0072F                PUSHL   #PAT$_READERR                                   ;
00000000G   00                  05   FB 00735                CALLS   #5, LIB$SIGNAL                                  ;
00000000G   EF          14      AE   D0 0073C 72$:           MOVL    OUT_BUF_PTR, PAT$GL_NEWRAB+40                   ; 1691
                    00000000G   EF   9F 00744                PUSHAB  PAT$GL_NEWRAB                                   ; 1692
00000000G   00                  01   FB 0074A                CALLS   #1, SYS$WRITE                                   ;
00000000G   EF                  50   D0 00751                MOVL    R0, PAT$GL_ERRCODE                              ;
              25 00000000G      EF   E8 00758                BLBS    PAT$GL_ERRCODE, 73$                             ; 1693
              7E 00000000G      EF   7D 0075F                MOVQ    PAT$GL_NEWRAB+8, -(SP)                          ; 1695
                    00000000G   EF   9F 00766                PUSHAB  PAT$GL_NEWFAB                                   ;
00000000G   EF                  01   FB 0076C                CALLS   #1, GETFILDSC                                   ;
                                50   DD 00773                PUSHL   R0                                              ;
                                01   DD 00775                PUSHL   #1                                              ;
                    00000000G   8F   DD 00777                PUSHL   #PAT$_WRITEERR                                  ;
00000000G   00                  05   FB 0077D                CALLS   #5, LIB$SIGNAL                                  ;
              54                53   D1 00784 73$:           CMPL    COUNTER, BLK_DIFF                               ; 1701
                                0F   13 00787                BEQL    74$                                             ;
        08    AE        0C BB43   9E 00789                MOVAB   @12(OLD_IHSYM_PTR)[COUNTER], CUR_VBN            ; 1704
00000000G   EF          0C B643   9E 0078F                MOVAB   @12(NEW_IHSYM_PTR)[COUNTER], -                  ; 1705
                                                                  PAT$GL_NEWRAB+56                                  ;
                      FF28  31   00798 74$:                BRW     68$                                             ; 1660
50              54      0C   A6   C1 0079B 75$:           ADDL3   12(NEW_IHSYM_PTR), BLK_DIFF, R0                 ; 1710
                50              59   D1 007A0                CMPL    MAX_VBN_WRITTEN, R0                             ;
                                03   1E 007A3                BGEQU   76$                                             ;
                59              50   D0 007A5                MOVL    R0, MAX_VBN_WRITTEN                             ; 1712
        DC    AD 00000000G      EF   9E 007A8 76$:           MOVAB   PAT$GB_OLDNAME, TMPFAB+44                       ; 1724
        E4    AD 00000000G      EF   90 007B0                MOVB    PAT$GL_OLDNBK+3, TMPFAB+52                      ; 1725
                        5C      AE   9F 007B8                PUSHAB  TMPFAB                                          ; 1726
00000000G   00                  01   FB 007BB                CALLS   #1, SYS$OPEN                                    ;
00000000G   EF                  50   D0 007C2                MOVL    R0, PAT$GL_ERRCODE                              ;
              21 00000000G      EF   E8 007C9                BLBS    PAT$GL_ERRCODE, 77$                             ; 1727
                        68      AE   DD 007D0                PUSHL   TMPFAB+12                                       ; 1730
                        68      AE   DD 007D3                PUSHL   TMPFAB+8                                        ;
                        64      AE   9F 007D6                PUSHAB  TMPFAB                                          ; 1729
00000000G   EF                  01   FB 007D9                CALLS   #1, GETFILDSC                                   ;
                                50   DD 007E0                PUSHL   R0                                              ;
                                01   DD 007E2                PUSHL   #1                                              ;
                    00000000G   8F   DD 007E4                PUSHL   #PAT$_OPENIN                                    ;
00000000G   00                  05   FB 007EA                CALLS   #5, LIB$SIGNAL                                  ;
                        18      AE   9F 007F1 77$:           PUSHAB  TMPRAB                                          ; 1732
00000000G   00                  01   FB 007F4                CALLS   #1, SYS$CONNECT                                 ;
00000000G   EF                  50   D0 007FB                MOVL    R0, PAT$GL_ERRCODE                              ;
              21 00000000G      EF   E8 00802                BLBS    PAT$GL_ERRCODE, 78$                             ; 1733
                        24      AE   DD 00809                PUSHL   TMPRAB+12                                       ; 1736
                        24      AE   DD 0080C                PUSHL   TMPRAB+8                                        ;
                        64      AE   9F 0080F                PUSHAB  TMPFAB                                          ; 1735
00000000G   EF                  01   FB 00812                CALLS   #1, GETFILDSC                                   ;
                                50   DD 00819                PUSHL   R0                                              ;
                                01   DD 0081B                PUSHL   #1                                              ;
                    00000000G   8F   DD 0081D                PUSHL   #PAT$_OPENIN                                    ;
00000000G   00                  05   FB 00823                CALLS   #5, LIB$SIGNAL                                  ;
        63    AE                08   88 0082A 78$:           BISB2   #8, TMPFAB+7                                    ; 1740
        74    AE                01   D0 0082E                MOVL    #1, TMPFAB+24                                   ; 1741
        7B    AE                02   90 00832                MOVB    #2, TMPFAB+31                                   ; 1742
```

```
                        5C   AE  9F 00836          PUSHAB  TMPFAB                                  ; 1744
00000000G   00               01  FB 00839          CALLS   #1, SYS$MODIFY
00000000G   EF               50  D0 00840          MOVL    R0, PAT$GL_ERRCODE
                     21      50  E8 00847          BLBS    R0, 79$
                        24   AE  DD 0084A          PUSHL   TMPRAB+12                               ; 1747
                        24   AE  DD 0084D          PUSHL   TMPRAB+8
                        64   AE  9F 00850          PUSHAB  TMPFAB                                  ; 1746
00000000G   EF               01  FB 00853          CALLS   #1, GETFILDSC
                             50  DD 0085A          PUSHL   R0
                             01  DD 0085C          PUSHL   #1
                 00000000G   8F  DD 0085E          PUSHL   #PAT$_OPENIN
00000000G   00               05  FB 00864          CALLS   #5, LIB$SIGNAL
00000000G   .F          0200 8F  B0 0086B 79$:      MOVW    #512, PAT$GL_NEWRAB+32                  ; 1751
00000000G   EF            14 AE  D0 00874          MOVL    OUT_BUF_PTR, PAT$GL_NEWRAB+36           ; 1752
                          04 A6  D5 0087C          TSTL    4(NEW_IHSYM_PTR)                        ; 1753
                             04  13 0087F          BEQL    80$
          04   A6              59  D0 00881          MOVL    MAX_VBN_WRITTEN, 4(NEW_IHSYM_PTR)       ; 1755
                             59  D5 00885 80$:      TSTL    MAX_VBN_WRITTEN                         ; 1756
                             0A  13 00887          BEQL    81$
00000000G   EF               FF  A9 9E 00889          MOVAB   -1(R9), PAT$GL_NEWRAB+56                ; 1758
                             0D  11 00891          BRB     82$
                 006D814A    8F  DD 00893 81$:      PUSHL   #7176522                                ; 1760
00000000G   00               01  FB 00899          CALLS   #1, LIB$SIGNAL
                 00000000G   EF  9F 008A0 82$:      PUSHAB  PAT$GL_NEWRAB                           ; 1762
00000000G   00               01  FB 008A6          CALLS   #1, SYS$READ
00000000G   EF               50  D0 008AD          MOVL    R0, PAT$GL_ERRCODE
                     25 00000000G EF  E8 008B4          BLBS    PAT$GL_ERRCODE, 83$                     ; 1763
                     7E 00000000G EF  7D 008BB          MOVQ    PAT$GL_NEWRAB+8, -(SP)                  ; 1765
                 00000000G   EF  9F 008C2          PUSHAB  PAT$GL_NEWFAB
00000000G   EF               01  FB 008C8          CALLS   #1, GETFILDSC
                             50  DD 008CF          PUSHL   R0
                             03  DD 008D1          PUSHL   #3
                 00000000G   8F  DD 008D3          PUSHL   #PAT$_READERR
00000000G   00               05  FB 008D9          CALLS   #5, LIB$SIGNAL
          28   AE        04 AB  D0 008E0 83$:      MOVL    4(OLD_IHSYM_PTR), TMPRAB+16             ; 1769
                          2C AE  B4 008E5          CLRW    TMPRAB+20                               ; 1770
          36   AE              02  90 008E8          MOVB    #2, TMPRAB+30                           ; 1771
00000000G   EF            14 AE  D0 008EC          MOVL    OUT_BUF_PTR, PAT$GL_NEWRAB+40           ; 1775
          3C   AE         14 AE  D0 008F4          MOVL    OUT_BUF_PTR, TMPRAB+36                  ; 1776
                          03      52  E8 008F9          BLBS    R2, 85$                                 ; 1783
                             0110 31  008FC 84$:      BRW     91$
                          0A AB  B5 008FF 85$:      TSTW    10(OLD_IHSYM_PTR)
                             F8  13 00902          BEQL    84$
                          02 04 AB  D1 00904          CMPL    4(OLD_IHSYM_PTR), #2                    ; 1784
                             F2  15 00908          BLEQ    84$
                          53 0A AB  3C 0090A          MOVZWL  10(OLD_IHSYM_PTR), COUNTER              ; 1787
                             03  14 0090E 86$:      BGTR    87$                                     ; 1788
                             0089 31 00910          BRW     90$
                          18 AE  9F 00913 87$:      PUSHAB  TMPRAB                                  ; 1791
00000000G   00               01  FB 00916          CALLS   #1, SYS$GET
00000000G   EF               50  D0 0091D          MOVL    R0, PAT$GL_ERRCODE
                     21 00000000G EF  E8 00924          BLBS    PAT$GL_ERRCODE, 88$                     ; 1792
                        24   AE  DD 0092B          PUSHL   TMPRAB+12                               ; 1795
                        24   AE  DD 0092E          PUSHL   TMPRAB+8
                        64   AE  9F 00931          PUSHAB  TMPFAB                                  ; 1794
00000000G   EF               01  FB 00934          CALLS   #1, GETFILDSC
                             50  DD 0093B          PUSHL   R0
```

```
                                        01  DD 0093D          PUSHL    #1                                           ;
                            00000000G   8F  DD 0093F          PUSHL    #PAT$_READERR                                ;
               00000000G  00            05  FB 00945          CALLS    #5, LIB$SIGNAL                               ;
                                  36    AE  94 0094C  88$:    CLRB     TMPRAB+30                                    ; 1796
               00000000G  EF      3A    AE  B0 0094F          MOVW     TMPRAB+34, PAT$GL_NEWRAB+34                  ; 1797
                            00000000G   EF  9F 00957          PUSHAB   PAT$GL_NEWRAB                                ; 1798
               00000000G  00            01  FB 0095D          CALLS    #1, SYS$PUT                                  ;
               00000000G  EF            50  D0 00964          MOVL     R0, PAT$GL_ERRCODE                           ;
                          25  00000000G EF  E8 0096B          BLBS     PAT$GL_ERRCODE, 89$                          ; 1799
                          7E  00000000G EF  7D 00972          MOVQ     PAT$GL_NEWRAB+8, -(SP)                       ; 1801
                            00000000G   EF  9F 00979          PUSHAB   PAT$GL_NEWFAB                                ;
               00000000G  EF            01  FB 0097F          CALLS    #1, GETFILDSC                                ;
                                        50  DD 00986          PUSHL    R0                                           ;
                                        01  DD 00988          PUSHL    #1                                           ;
                            00000000G   8F  DD 0098A          PUSHL    #PAT$_WRITEERR                               ;
               00000000G  00            05  FB 00990          CALLS    #5, LIB$SIGNAL                               ;
                                        53  D7 00997  89$:    DECL     COUNTER                                      ; 1802
                                      FF72  31 00999          BRW      86$                                          ; 1788
                          50  00000000G EF  3C 0099C  90$:    MOVZWL   PAT$GL_NEWRAB+34, R0                         ; 1811
                          51  00000000G EF  3C 009A3          MOVZWL   PAT$GL_NEWRAB+20, R1                         ;
                          50            51  C0 009AA          ADDL2    R1, R0                                       ;
        00000000G  EF      01FC   8F    50  A3 009AD          SUBW3    R0, #508, PAT$GL_NEWRAB+34                   ;
                          50  00000000G EF  3C 009B7          MOVZWL   PAT$GL_NEWRAB+34, R0                         ; 1812
                          4F            15  009BE             BLEQ     91$                                          ;
                   0200   8F            50  B1 009C0          CMPW     R0, #512                                     ;
                          48            1E  009C5             BGEQU    91$                                          ;
      50       FF    8F   6E            00  2C 009C7          MOVC5    #0, (SP), #255, R0, @OUT_BUF_PTR             ; 1815
                                  14    BE     009CD                                                                ;
                            00000000G   EF  9F 009CF          PUSHAB   PAT$GL_NEWRAB                                ; 1816
               00000000G  00            01  FB 009D5          CALLS    #1, SYS$PUT                                  ;
               00000000G  EF            50  D0 009DC          MOVL     R0, PAT$GL_ERRCODE                           ;
                          25  00000000G EF  E8 009E3          BLBS     PAT$GL_ERRCODE, 91$                          ; 1817
                          7E  00000000G EF  7D 009EA          MOVQ     PAT$GL_NEWRAB+8, -(SP)                       ; 1819
                            00000000G   EF  9F 009F1          PUSHAB   PAT$GL_NEWFAB                                ;
               00000000G  EF            01  FB 009F7          CALLS    #1, GETFILDSC                                ;
                                        50  DD 009FE          PUSHL    R0                                           ;
                                        01  DD 00A00          PUSHL    #1                                           ;
                            00000000G   8F  DD 00A02          PUSHL    #PAT$_WRITEERR                               ;
               00000000G  00            05  FB 00A08          CALLS    #5, LIB$SIGNAL                               ;
                     50   6E            20  C1 00A0F  91$:    ADDL3    #32, NEW_IHPAT_PTR, R0                       ; 1826
                     60  00000000G  EF  01  C1 00A13          ADDL3    #1, PAT$GL_NEWRAB+16, (R0)                   ;
                          50  00000000G EF  D0 00A1B          MOVL     PAT$GL_IHPPTR, R0                            ; 1832
                                  20    A0  D5 00A22          TSTL     32(R0)                                       ;
                                        28  13 00A25          BEQL     93$                                          ;
                  36      AE            02  90 00A27          MOVB     #2, TMPRAB+30                                ; 1835
                  28      AE      20    A0  D0 00A2B          MOVL     32(R0), TMPRAB+16                            ; 1836
                                  2C    AE  B4 00A30          CLRW     TMPRAB+20                                    ; 1837
                                  18    AE  9F 00A33  92$:    PUSHAB   TMPRAB                                       ; 1844
               00000000G  00            01  FB 00A36          CALLS    #1, SYS$GET                                  ;
               00000000G  EF            50  D0 00A3D          MOVL     R0, PAT$GL_ERRCODE                           ;
               0001827A   8F  00000000G EF  D1 00A44          CMPL     PAT$GL_ERRCODE, #98938                       ; 1845
                                        76  13 00A4F  93$:    BEQL     95$                                          ;
                          21  00000000G EF  E8 00A51          BLBS     PAT$GL_ERRCODE, 94$                          ; 1848
                                  24    AE  DD 00A58          PUSHL    TMPRAB+12                                    ; 1851
                                  24    AE  DD 00A5B          PUSHL    TMPRAB+8                                     ;
                                  64    AE  9F 00A5E          PUSHAB   TMPFAB                                       ; 1850
               00000000G  EF            01  FB 00A61          CALLS    #1, GETFILDSC                                ;
```

```
                                      50   DD 00A68          PUSHL    R0                                          ;
                                      01   DD 00A6A          PUSHL    #1                                          ;
                           00000000G  8F   DD 00A6C          PUSHL    #PAT$_READERR                               ;
              00000000G 00            05   FB 00A72          CALLS    #5, LIB$SIGNAL                              ;
                               36     AE   94 00A79 94$:     CLRB     TMPRAB+30                                   ; 1852
              00000000G EF     3A     AE   B0 00A7C          MOVW     TMPRAB+34, PAT$GL_NEWRAB+34                 ; 1853
                           00000000G  EF   9F 00A84          PUSHAB   PAT$GL_NEWRAB                               ; 1854
              00000000G 00            01   FB 00A8A          CALLS    #1, SYS$PUT                                 ;
              00000000G EF            50   D0 00A91          MOVL     R0, PAT$GL_ERRCODE                          ;
                        94 00000000G  EF   E8 00A98          BLBS     PAT$GL_ERRCODE, 92$                         ; 1855
                        7E 00000000G  EF   7D 00A9F          MOVQ     PAT$GL_NEWRAB+8, -(SP)                      ; 1857
                           00000000G  EF   9F 00AA6          PUSHAB   PAT$GL_NEWFAB                               ;
              00000000G EF            01   FB 00AAC          CALLS    #1, GETFILDSC                               ;
                                      50   DD 00AB3          PUSHL    R0                                          ;
                                      01   DD 00AB5          PUSHL    #1                                          ;
                           00000000G  8F   DD 00AB7          PUSHL    #PAT$_WRITEERR                              ;
              00000000G 00            05   FB 00ABD          CALLS    #5, LIB$SIGNAL                              ;
                                    FF6C   31 00AC4          BRW      92$                                         ; 1837
                        5A 00000000G  EF   D0 00AC7 95$:     MOVL     PAT$GL_TXTLHD, COM_TXT_PTR                  ; 1867
                                      68   13 00ACE 96$:     BEQL     100$                                        ; 1868
                        52     04     AA   9E 00AD0          MOVAB    4(R10), COM_PTR                             ; 1871
                                      62   95 00AD4 97$:     TSTB     (COM_PTR)                                   ; 1872
                                      5B   13 00AD6          BEQL     99$                                         ;
              00000000G EF            62   9B 00AD8          MOVZBW   (COM_PTR), PAT$GL_NEWRAB+34                 ; 1875
              00000000G EF     01     A2   9E 00ADF          MOVAB    1(R2), PAT$GL_NEWRAB+40                     ; 1876
                           00000000G  EF   9F 00AE7          PUSHAB   PAT$GL_NEWRAB                               ; 1877
              00000000G 00            01   FB 00AED          CALLS    #1, SYS$PUT                                 ;
              00000000G EF            50   D0 00AF4          MOVL     R0, PAT$GL_ERRCODE                          ;
                        25 00000000G  EF   E8 00AFB          BLBS     PAT$GL_ERRCODE, 98$                         ; 1878
                        7E 00000000G  EF   7D 00B02          MOVQ     PAT$GL_NEWRAB+8, -(SP)                      ; 1880
                           00000000G  EF   9F 00B09          PUSHAB   PAT$GL_NEWFAB                               ;
              00000000G EF            01   FB 00B0F          CALLS    #1, GETFILDSC                               ;
                                      50   DD 00B16          PUSHL    R0                                          ;
                                      01   DD 00B18          PUSHL    #1                                          ;
                           00000000G  8F   DD 00B1A          PUSHL    #PAT$_WRITEERR                              ;
              00000000G 00            05   FB 00B20          CALLS    #5, LIB$SIGNAL                              ;
                        50            62   9A 00B27 98$:     MOVZBL   (COM_PTR), R0                               ; 1881
                        50            52   C0 00B2A          ADDL2    COM_PTR, R0                                 ;
                        52     01     A0   9E 00B2D          MOVAB    1(R0), COM_PTR                              ;
                                      A1   11 00B31          BRB      97$                                         ; 1872
                        5A            6A   D0 00B33 99$:     MOVL     (COM_TXT_PTR), COM_TXT_PTR                  ; 1883
                                      96   11 00B36          BRB      96$                                         ; 1868
                        51 00000000G  EF   9A 00B38 100$:    MOVZBL   PAT$GB_ECOLVL, R1                           ; 1894
                                      14   13 00B3F          BEQL     102$                                        ;
                        50            6E   D0 00B41          MOVL     NEW_IHPAT_PTR, ECO_LEVEL_PTR                ; 1897
                                      51   D7 00B44          DECL     R1                                          ; 1898
           00           60            51   E2 00B46          BBSS     R1, (ECO_LEVEL_PTR), 101$                   ;
                        50 00000000G  EF   D0 00B4A 101$:    MOVL     PAT$GL_IHPPTR, ECO_LEVEL_PTR                ; 1899
           00           60            51   E2 00B51          BBSS     R1, (ECO_LEVEL_PTR), 102$                   ; 1900
              00000000G EF            01   D0 00B55 102$:    MOVL     #1, PAT$GL_NEWRAB+56                        ; 1902
00000000G  EF    0C     AE     0200   8F   A5 00B5C          MULW3    #512, NUM_HDR_BLKS, PAT$GL_NEWRAB+34        ; 1903
              00000000G EF     10     AE   D0 00B67          MOVL     NEW_IHD_PTR, PAT$GL_NEWRAB+40               ; 1904
              00000000G EF            02   8A 00B6F          BICB2    #2, PAT$GL_NEWRAB+4                         ; 1910
                           00000000G  EF   9F 00B76          PUSHAB   PAT$GL_NEWRAB                               ; 1911
              00000000G 00            01   FB 00B7C          CALLS    #1, SYS$WRITE                               ;
              00000000G EF            50   D0 00B83          MOVL     R0, PAT$GL_ERRCODE                          ;
                        25 00000000G  EF   E8 00B8A          BLBS     PAT$GL_ERRCODE, 103$                        ; 1912
```

```
                7E 00000000G   EF   7D 00B91          MOVQ    PAT$GL_NEWRAB+8, -(SP)                  ; 1914
                   00000000G   EF   9F 00B98          PUSHAB  PAT$GL_NEWFAB                           ;
00000000G       EF             01   FB 00B9E          CALLS   #1, GETFILDSC                           ;
                               50   DD 00BA5          PUSHL   R0                                      ;
                               01   DD 00BA7          PUSHL   #1                                      ;
                   00000000G   8F   DD 00BA9          PUSHL   #PAT$_WRITEERR                          ;
00000000G       00             05   FB 00BAF          CALLS   #5, LIB$SIGNAL                          ;
                   00000000G   EF   94 00BB6 103$:    CLRB    PAT$GB_ECOLVL                           ; 1915
00000000G       EF             08   88 00BBC          BISB2   #8, PAT$GL_NEWFAB+7                     ; 1920
00000000G       EF             01   D0 00BC3          MOVL    #1, PAT$GL_NEWFAB+24                    ; 1921
00000000G       EF             01   90 00BCA          MOVB    #1, PAT$GL_NEWFAB+31                    ; 1922
                   00000000G   EF   9F 00BD1          PUSHAB  PAT$GL_NEWFAB                           ; 1923
00000000G       00             01   FB 00BD7          CALLS   #1, SYS$MODIFY                          ;
00000000G       EF             50   D0 00BDE          MOVL    R0, PAT$GL_ERRCODE                      ;
                26             50   E8 00BE5          BLBS    R0, 104$                                ; 1924
                   00000000G   EF   DD 00BE8          PUSHL   PAT$GL_NEWRAB+12                        ; 1927
                               50   DD 00BEE          PUSHL   R0                                      ;
                   00000000G   EF   9F 00BF0          PUSHAB  PAT$GB_NEWNAME                          ; 1926
                7E 00000000G   EF   9A 00BF6          MOVZBL  PAT$GL_NEWNBK+3, -(SP)                  ;
                               50   DD 00BFD          PUSHL   R0                                      ;
                               03   DD 00BFF          PUSHL   #3                                      ;
                   006D819C    8F   DD 00C01          PUSHL   #7176604                                ;
00000000G       00             07   FB 00C07          CALLS   #7, LIB$SIGNAL                          ;
                   00000000G   EF   9F 00C0E 104$:    PUSHAB  PAT$GL_NEWFAB                           ; 1933
00000000G       00             01   FB 00C14          CALLS   #1, SYS$CLOSE                           ;
00000000G       EF             50   D0 00C1B          MOVL    R0, PAT$GL_ERRCODE                      ;
                27 00000000G   EF   E8 00C22          BLBS    PAT$GL_ERRCODE, 105$                    ; 1934
                7E 00000000G   EF   7D 00C29          MOVQ    PAT$GL_NEWFAB+8, -(SP)                  ; 1936
                   00000000G   EF   9F 00C30          PUSHAB  PAT$GL_NEWFAB                           ;
00000000G       EF             01   FB 00C36          CALLS   #1, GETFILDSC                           ;
                               50   DD 00C3D          PUSHL   R0                                      ;
                               01   DD 00C3F          PUSHL   #1                                      ;
                   00000000G   8F   DD 00C41          PUSHL   #PAT$_CLOSEOUT                          ;
00000000G       00             05   FB 00C47          CALLS   #5, LIB$SIGNAL                          ;
                               07   11 00C4E          BRB     106$                                    ;
00000000G       EF             08   8A 00C50 105$:    BICB2   #8, PAT$GL_FLAGS                        ; 1938
                      5C       AE   9F 00C57 106$:    PUSHAB  TMPFAB                                  ; 1943
00000000G       00             01   FB 00C5A          CALLS   #1, SYS$CLOSE                           ;
00000000G       EF             50   D0 00C61          MOVL    R0, PAT$GL_ERRCODE                      ;
                21 00000000G   EF   E8 00C68          BLBS    PAT$GL_ERRCODE, 107$                    ; 1944
                         68    AE   DD 00C6F          PUSHL   TMPFAB+12                               ; 1946
                         68    AE   DD 00C72          PUSHL   TMPFAB+8                                ;
                         54    AE   9F 00C75          PUSHAB  TMPFAB                                  ;
00000000G       EF             01   FB 00C78          CALLS   #1, GETFILDSC                           ;
                               50   DD 00C7F          PUSHL   R0                                      ;
                               01   DD 00C81          PUSHL   #1                                      ;
                   0000000CG   8F   DD 00C83          PUSHL   #PAT$_CLOSEIN                           ;
00000000G       00             05   FB 00C89          CALLS   #5, LIB$SIGNAL                          ;
                               04   00C90 107$:       RET                                             ; 1949

; Routine Size:  3217 bytes,    Routine Base:  _PAT$CODE + 0000
```

```
;  1056          1950  1 %SBTTL  'GET_IMAGE_BLOCK -- read block from input image'
;  1057          1951  1 ROUTINE GET_IMAGE_BLOCK (VBN, BUFFER, BYTES_TO_READ) =
;  1058          1952  1 !++
;  1059          1953  1 !
;  1060          1954  1 ! Functional Description:
;  1061          1955  1 !       This routine is called to read a block from the input image file,
;  1062          1956  1 !       which is assumed to be open.
;  1063          1957  1 !
;  1064          1958  1 ! FORMAL PARAMETERS
;  1065          1959  1 !       VBN - virtual block number of desired block
;  1066          1960  1 !       BUFFER  - Address of buffer pointer to fill in with the address of our
;  1067          1961  1 !                 buffer.
;  1068          1962  1 !       BYTES_TO_READ = number of bytes to read
;  1069          1963  1 !
;  1070          1964  1 ! ROUTINE VALUE
;  1071          1965  1 !       $QIOW status
;  1072          1966  1 !
;  1073          1967  1 !--
;  1074          1968  2 BEGIN
;  1075          1969  2
;  1076          1970  2 LOCAL
;  1077          1971  2     STATUS,
;  1078          1972  2     IOSB : VECTOR [4,WORD];
;  1079          1973  2
;  1080          1974  2 ! Read in the desired block to the static buffer.
;  1081          1975  2 !
;  1082          1976  2
;  1083        P 1977  2 STATUS = $QIOW ( EFN = 7,
;  1084        P 1978  2                  CHAN = .PAT$GL_CHANUM,
;  1085        P 1979  2                  FUNC = IO$_READVBLK,
;  1086        P 1980  2                  IOSB = IOSB,
;  1087        P 1981  2                  P1 = .BUFFER,
;  1088        P 1982  2                  P2 = .BYTES_TO_READ,
;  1089          1983  2                  P3 = .VBN );
;  1090          1984  2
;  1091          1985  2 IF NOT .STATUS
;  1092          1986  2 THEN
;  1093          1987  2     RETURN (.STATUS);
;  1094          1988  2
;  1095          1989  2 ! Point the caller's pointer at our buffer.  Then return the $QIOW status
;  1096          1990  2 !
;  1097          1991  2 RETURN (.IOSB[0]);
;  1098          1992  2
;  1099          1993  1 END;


                                                .EXTRN  SYS$QIOW

                                0000 00000 GET_IMAGE_BLOCK:
                                                .WORD   Save nothing                    ; 1951
                             5E          08  C2 00002        SUBL2   #8, SP             ;
                                         7E  7C 00005        CLRQ    -(SP)              ; 1983
                                         7E  D4 00007        CLRL    -(SP)              ;
                                   04    AC  DD 00009        PUSHL   VBN                ;
                             7E    08    AC  7D 0000C        MOVQ    BUFFER, -(SP)      ;
                                         7E  7C 00010        CLRQ    -(SP)              ;
```

```
                                    20  AE  9F 00012          PUSHAB  IOSB                                                    ;
                                        31  DD 00015          PUSHL   #49                                                     ;
                                00000000G   EF  DD 00017          PUSHL   PAT$GL_CHANUM                                           ;
                                        07  DD 0001D          PUSHL   #7                                                      ;
                        00000000G   00      0C  FB 0001F          CALLS   #12, SYS$QIOW                                           ;
                                    03      50  E9 00026          BLBC    STATUS, 1$                                              ; 1985
                                    50      6E  3C 00029          MOVZWL  IOSB, R0                                                ; 1991
                                            04 0002C 1$:          RET                                                             ; 1993

; Routine Size:  45 bytes,    Routine Base:  _PAT$CODE + 0C91
```

```
;  1101          1994  1 %SBTTL  'write_binary -- Write a binary file after patching it /ABSOLUTE'
;  1102          1995  1 ROUTINE write_binary : NOVALUE =
;  1103          1996  1 !++
;  1104          1997  1 !
;  1105          1998  1 ! Functional Description:
;  1106          1999  1 !
;  1107          2000  1 !       This routine writes the patched file as a binary data file.  The
;  1108          2001  1 !       output file is created/opened (or attempts to...) using the same
;  1109          2002  1 !       attributes of the file being patched, failing that it will attempt
;  1110          2003  1 !       to create the file contiguous best try (failing this its an error).
;  1111          2004  1 !
;  1112          2005  1 !       Once the output file is established we traverse the ISE list to
;  1113          2006  1 !       determine which sections must be copied from the orignal file to
;  1114          2007  1 !       the output file and which are already mapped into virtual memory.
;  1115          2008  1 !
;  1116          2009  1 ! Side Effects:
;  1117          2010  1 !
;  1118          2011  1 !       Upon exit the the ISE list remains intact, however, the MAPVST/END
;  1119          2012  1 !       elements are set to zero and the sections are unmapped.  The output
;  1120          2013  1 !       file has been written and $CLOSEd.
;  1121          2014  1 !
;  1122          2015  1 !--
;  1123          2016  2 BEGIN
;  1124          2017  2
;  1125          2018  2 LITERAL
;  1126          2019  2         MAX_TRANSFER = %X'FE00';                        ! Maximum number of bytes rms can write
;  1127          2020  2                                                         ! per transfer.
;  1128          2021  2 LOCAL
;  1129          2022  2         BYTE_COUNT : WORD,                              ! Number of bytes per transfer.
;  1130          2023  2         ISD_PTR : REF BLOCK[,BYTE],                     ! Pointer to current ISD
;  1131          2024  2         ISE_PTR : REF BLOCK[,BYTE],                     ! Pointer to current ISE
;  1132          2025  2         REMAINDER,                                      ! Number of byte remaining to be transfered.
;  1133          2026  2         TRANSFER_COUNT,                                 ! Number of bytes transfered.
;  1134          2027  2         SECTION_SIZE,                                   ! Size of this section in bytes.
;  1135          2028  2         START_VA;                                       ! Starting virtual address of the transfer
;  1136          2029  2
;  1137          2030  2 IF .PAT$GL_FLAGS [PAT$S_VOLUME]                         ! The VOLUME qualifier specifies the RVN
;  1138          2031  2 THEN
;  1139          2032  3         BEGIN
;  1140          2033  3         PAT$GL_NEWXABALL[XAB$W_VOL] = .PAT$GW_IMGVOL;   ! The Relative Volume Number
;  1141          2034  3         PAT$GL_NEWXABALL[XAB$B_ALN] = XAB$C_LBN;        ! To enable XAB$W_VOL
;  1142          2035  2         END;
;  1143          2036  2
;  1144          2037  2 IF .PAT$GL_FLAGS [PAT$S_NEW_VERSION]
;  1145          2038  2 THEN
;  1146          2039  3     BEGIN
;  1147          2040  4     IF (.NUM_OF_UPDATES EQL 0)                          ! Check number of updates done
;  1148          2041  3     THEN
;  1149          2042  4         BEGIN
;  1150          2043  4         !++
;  1151          2044  4         ! Create the output file.  Try to make it a contiguous file if
;  1152          2045  4         ! the input file was contiguous, i.e., first try a create with
;  1153          2046  4         ! the same attributes.  If the file cannot be created with the same
;  1154          2047  4         ! attributes, then attempt a second try with contiguous-best-try.  If
;  1155          2048  4         ! this succeeds, then print an informational message.
;  1156          2049  4         !--
;  1157          2050  4         PAT$GL_NEWXABALL[XAB$L_ALQ] = .PAT$GL_OLDFAB[FAB$L_ALQ];        ! Allocation quantity.
```

```
;1158    2051  4          PAT$GL_NEWXABALL[XAB$V_CTG] = .PAT$GL_OLDFAB[FAB$V_CTG];        ! Contiguous.
;1159    2052  4          PAT$GL_NEWXABALL[XAB$V_CBT] = .PAT$GL_OLDFAB[FAB$V_CBT];        ! Contiguous Best Try.
;1160    2053  4          PAT$GL_NEWFAB [FAB$B_DNS] = .PAT$GL_OLDNBK [NAM$B_TYPE];        ! File type size.
;1161    2054  4          PAT$GL_NEWFAB [FAB$L_DNA] = .PAT$GL_OLDNBK [NAM$L_TYPE];        ! File type address.
;1162    2055  4          PAT$GL_NEWFAB [FAB$W_MRS] = .PAT$GL_OLDFAB [FAB$W_MRS];         ! Maximum record size.
;1163    2056  4          PAT$GL_NEWFAB [FAB$B_RAT] = .PAT$GL_OLDFAB [FAB$B_RAT];         ! Record attributes.
;1164    2057  4          PAT$GL_NEWFAB [FAB$B_RFM] = .PAT$GL_OLDFAB [FAB$B_RFM];         ! Record format.
;1165    2058  4          PAT$GL_NEWRAB [RAB$V_BIO] = TRUE;                               ! Perform Block IO.
;1166    2059  4          PAT$GL_NEWRAB [RAB$V_TPT] = TRUE;                               ! Truncate the file after each write
;1167    2060  4
;1168    2061  5          IF NOT (PAT$GL_ERRCODE=$CREATE(FAB=PAT$GL_NEWFAB))
;1169    2062  4          THEN
;1170    2063  5                  BEGIN                                                   ! Attempt a contiguous best try
;1171    2064  5                  PAT$GL_NEWXABALL[XAB$V_CBT] = TRUE;
;1172    2065  5                  PAT$GL_ERRCODE = $CREATE(FAB=PAT$GL_NEWFAB);
;1173    2066  5                  IF .PAT$GL_ERRCODE
;1174    2067  5                  THEN
;1175    2068  5                          SIGNAL(PAT$_NONCONTIG+MSG$K_INFO,.PAT$GL_ERRCODE,.PAT$GL_NEWRAB[RAB$L_STV]);
;1176    2069  4                  END;
;1177    2070  4          END
;1178    2071  3      ELSE
;1179    2072  3          PAT$GL_ERRCODE=$OPEN(FAB=PAT$GL_NEWFAB);                        ! Open output file
;1180    2073  3
;1181    2074  3      IF NOT .PAT$GL_ERRCODE                                              ! Success on open?
;1182    2075  3      THEN SIGNAL(PAT$_OPENOUT,1,GETFILDSC(PAT$GL_NEWFAB),.PAT$GL_NEWFAB[FAB$L_STS],.PAT$GL_NEWRAB[RAB$L_STV])
;1183    2076  3      ELSE
;1184    2077  4          BEGIN
;1185    2078  4          PAT$GL_ERRCODE=$CONNECT(RAB=PAT$GL_NEWRAB);                     ! Connect input file
;1186    2079  4          IF NOT .PAT$GL_ERRCODE                                          ! Success on connect?
;1187    2080  4          THEN
;1188    2081  4              SIGNAL(PAT$_OPENOUT,1,GETFILDSC(PAT$GL_NEWFAB)
;1189    2082  4                     .PAT$GL_NEWRAB[RAB$L_STS],.PAT$GL_NEWRAB[RAB$L_STV]) ! REPORT FAILURE
;1190    2083  4          ELSE
;1191    2084  5              BEGIN
;1192    2085  5              NUM_OF_UPDATES = 1;                                        ! Set indicator for already created
;1193    2086  5              PAT$GL_FLAGS [PAT$S_OUTPUT] = 1;                           ! Set open file flag
;1194    2087  4              END;
;1195    2088  3          END;
;1196    2089  2      END;
;1197    2090  2
;1198    2091  2 !++
;1199    2092  2 ! Report file being written.
;1200    2093  2 !--
;1201    2094  2 IF .PAT$GL_FLAGS [PAT$S_NEW_VERSION]
;1202    2095  2 THEN     SIGNAL(PAT$_WRTFIL+MSG$K_INFO, 2, .PAT$GL_NEWNBK[NAM$B_RSL], PAT$GB_NEWNAME)
;1203    2096  2 ELSE
;1204    2097  3          BEGIN
;1205    2098  3          LOCAL
;1206    2099  3                  OLD_FILE : $BBLOCK [DSC$C_S_BLN];
;1207    2100  3          OLD_FILE [DSC$W_LENGTH] = .PAT$GL_OLDNBK[NAM$B_RSL];
;1208    2101  3          OLD_FILE [DSC$A_POINTER] = PAT$GB_OLDNAME;
;1209    2102  3          SIGNAL(PAT$_OVERLAY, 1, OLD_FILE);
;1210    2103  2          END;
;1211    2104  2
;1212    2105  2 ISE_PTR=CH$PTR(.PAT$GL_ISELHD,0);                                       ! Point to first ise
;1213    2106  2 WHILE .ISE_PTR NEQA 0                                                  ! Loop until ise's are exhausted
;1214    2107  3 DO       BEGIN
```

```
;  1215     2108  3          ISD_PTR = CH$PTR(.ISE_PTR, ISE$C_SIZE);                 ! Find old isd address
;  1216     2109  3
;  1217     2110  3          IF .PAT$GL_FLAGS [PAT$S_NEW_VERSION]
;  1218     2111  3          THEN                                                    ! We're creating a new patched copy of the f
;  1219     2112  4              BEGIN
;  1220     2113  4              IF .ISE_PTR[ISE$L_MAPVEND] EQLA 0                   ! If the image section is not mapped, then m
;  1221     2114  4              THEN        PAT$CREMAP (.ISE_PTR);
;  1222     2115  4
;  1223     2116  4              START_VA = .ISE_PTR[ISE$L_MAPVST];                  ! Set the starting virtual address of the se
;  1224     2117  4              SECTION_SIZE = .ISE_PTR [ISE$L_MAPVEND] - .ISE_PTR [ISE$L_MAPVST] + 1;
;  1225     2118  4
;  1226     2119  4              IF .SECTION_SIZE GTR MAX_TRANSFER                   ! If we can't write the section as a whole,
;  1227     2120  4              THEN    BYTE_COUNT = MAX_TRANSFER                   ! it up into pieces that RMS can deal with a
;  1228     2121  4              ELSE    BYTE_COUNT = .SECTION_SIZE;                 ! transfer it one chunk at a time.
;  1229     2122  4
;  1230     2123  4              !++
;  1231     2124  4              ! Now write out the section (in whole or parts...)
;  1232     2125  4              !--
;  1233     2126  4              TRANSFER_COUNT = 0;
;  1234     2127  4              REMAINDER = 0;
;  1235     2128  4              WHILE .START_VA LSSA .ISE_PTR [ISE$L_MAPVEND] DO    ! Transfer the entire section.
;  1236     2129  5                  BEGIN
;  1237     2130  5                  PAT$GL_NEWRAB[RAB$W_RSZ] = .BYTE_COUNT;         ! Set the number of byte to transfer.
;  1238     2131  5                  PAT$GL_NEWRAB[RAB$L_RBF] = .START_VA;           ! Set the starting address of the buffer.
;  1239     2132  6                  IF NOT (PAT$GL_ERRCODE = $WRITE(RAB=PAT$GL_NEWRAB)) ! Write it to the new file.
;  1240     2133  5                  THEN    SIGNAL(PAT$_WRITEERR, 1, GETFILDSC(PAT$GL_NEWFAB),
;  1241     2134  5                                 .PAT$GL_NEWRAB[RAB$L_STS],
;  1242     2135  5                                 .PAT$GL_NEWRAB[RAB$L_STV]);
;  1243     2136  5
;  1244     2137  5                  TRANSFER_COUNT = .TRANSFER_COUNT + .BYTE_COUNT; ! Update the transfer count
;  1245     2138  5                  START_VA = .START_VA + .BYTE_COUNT;             ! and the starting address of the buffer.
;  1246     2139  5
;  1247     2140  6                  IF ((REMAINDER = .SECTION_SIZE - .TRANSFER_COUNT) LEQ MAX_TRANSFER)
;  1248     2141  6                  AND (.REMAINDER GTR 0)                          ! If there's still something to transfer,
;  1249     2142  5                  THEN BYTE_COUNT = .REMAINDER;                   ! modifiy the byte (once its less than max!)
;  1250     2143  5
;  1251     2144  4                  END; ! of WHILE
;  1252     2145  4              END
;  1253     2146  3          ELSE                                                    ! We're patching the file in place, only upd
;  1254     2147  3              IF .ISE_PTR [ISE$L_MAPVEND] NEQ 0                   ! those portions affected.
;  1255     2148  4              THEN IF NOT (PAT$GL_ERRCODE = $UPDSEC (INADR = ISE_PTR [ISE$L_MAPVST]))
;  1256     2149  3                   THEN SIGNAL (PAT$_SYSERROR, 0, .PAT$GL_ERRCODE);
;  1257     2150  3          !+
;  1258     2151  3          ! Conserve virtual address space!  Once were done writing a section, release the
;  1259     2152  3          ! space back to the process for future use.
;  1260     2153  3          !-
;  1261     2154  3          IF .ISE_PTR [ISE$L_MAPVEND] NEQ 0
;  1262     2155  3          THEN
;  1263     2156  4              BEGIN
;  1264     2157  5              IF NOT (PAT$GL_ERRCODE = $DELTVA (INADR = ISE_PTR [ISE$L_MAPVST]))
;  1265     2158  4              THEN        SIGNAL (PAT$_SYSERROR, 0, .PAT$GL_ERRCODE);
;  1266     2159  4              ISE_PTR [ISE$L_MAPVST] = ISE_PTR [ISE$L_MAPVEND] = 0;
;  1267     2160  3              END;
;  1268     2161  3
;  1269     2162  3          ISE_PTR = .ISE_PTR[ISE$L_NXTISE];
;  1270     2163  3
;  1271     2164  2          END; ! of WHILE
```

```
2
2 !++
2 ! Now close the output image file.  This is done here so that the "update"
2 ! command can rewrite the file if update is specified more than once.
2 !--
2 IF .PAT$GL_FLAGS [PAT$S_NEW_VERSION]
2 THEN
3         BEGIN
3         PAT$GL_ERRCODE = $CLOSE(FAB=PAT$GL_NEWFAB);
3         IF NOT .PAT$GL_ERRCODE
3         THEN    SIGNAL(PAT$_CLOSEOUT, 1, GETFILDSC(PAT$GL_NEWFAB), .PAT$GL_NEWFAB[FAB$L_STS], .PAT$GL_NEWFAB
3         ELSE    PAT$GL_FLAGS [PAT$S_OUTPUT] = 0;
2         END;
2
1 END;    ! of ROUTINE write_binary


                                                            .EXTRN  SYS$UPDSEC, SYS$DELTVA

                                            0FFC 00000 WRITE_BINARY:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 1995
                       5B 00000000G  EF   9E 00002          MOVAB   PAT$GL_NEWFAB, R11                        ;
                       5A 00000000G  EF   9E 00009          MOVAB   PAT$GL_ERRCODE, R10                       ;
                       5E            08   C2 00010          SUBL2   #8, SP                                    ;
            12 00000000G  EF         05   E1 00013          BBC     #5, PAT$GL_FLAGS, 1$                      ; 2030
               00000000G  EF 00000000G  EF   B0 0001B          MOVW    PAT$GW_IMGVOL, PAT$GL_NEWXABALL+10        ; 2033
               00000000G  EF         02   90 00026          MOVB    #2, PAT$GL_NEWXABALL+9                    ; 2034
                          00000000G  EF   95 0002D 1$:      TSTB    PAT$GL_FLAGS                              ; 2037
                                     03   19 00033          BLSS    2$                                        ;
                                   00FD   31 00035          BRW     10$                                       ;
                          00000000'  EF   D5 00038 2$:      TSTL    NUM_OF_UPDATES                            ; 2040
                                     03   13 0003E          BEQL    3$                                        ;
                                   0094   31 00040          BRW     4$                                        ;
               00000000G  EF 00000000G  EF   D0 00043 3$:      MOVL    PAT$GL_OLDFAB+16, PAT$GL_NEWXABALL+16     ; 2050
      50 00000000G  EF         01         04   EF 0004E          EXTZV   #4, #1, PAT$GL_OLDFAB+6, R0               ; 2051
00000000G  EF         01         07         50   F0 00057          INSV    R0, #7, #1, PAT$GL_NEWXABALL+8            ;
      50 00000000G  EF         01         05   EF 00060          EXTZV   #5, #1, PAT$GL_OLDFAB+6, R0               ; 2052
00000000G  EF         01         05         50   F0 00069          INSV    R0, #5, #1, PAT$GL_NEWXABALL+8            ;
                       35   AB 00000000G  EF   90 00072          MOVB    PAT$GL_OLDNBK+60, PAT$GL_NEWFAB+53        ; 2053
                       30   AB 00000000G  EF   D0 0007A          MOVL    PAT$GL_OLDNBK+80, PAT$GL_NEWFAB+48        ; 2054
                       36   AB 00000000G  EF   B0 00082          MOVW    PAT$GL_OLDFAB+54, PAT$GL_NEWFAB+54        ; 2055
                       1E   AB 00000000G  EF   B0 0008A          MOVW    PAT$GL_OLDFAB+30, PAT$GL_NEWFAB+30        ; 2056
               00000000G  EF       0802   8F   A8 00092          BISW2   #2050, PAT$GL_NEWRAB+4                    ; 2058
                                     5B   DD 0009B          PUSHL   R11                                       ; 2061
               00000000G  00         01   FB 0009D          CALLS   #1, SYS$CREATE                            ;
                       6A            50   D0 000A4          MOVL    R0, PAT$GL_ERRCODE                        ;
                       39            50   E8 000A7          BLBS    R0, 5$                                    ;
               00000000G  EF         20   88 000AA          BISB2   #32, PAT$GL_NEWXABALL+8                   ; 2064
                                     5B   DD 000B1          PUSHL   R11                                       ; 2065
               00000000G  00         01   FB 000B3          CALLS   #1, SYS$CREATE                            ;
                       6A            50   D0 000BA          MOVL    R0, PAT$GL_ERRCODE                        ;
                       26            6A   E9 000BD          BLBC    PAT$GL_ERRCODE, 6$                        ; 2066
                          00000000G  EF   DD 000C0          PUSHL   PAT$GL_NEWRAB+12                          ; 2068
                                     6A   DD 000C6          PUSHL   PAT$GL_ERRCODE                            ;
                          006D82A3   8F   DD 000C8          PUSHL   #7176867                                  ;
               00000000G  00         03   FB 000CE          CALLS   #3, LIB$SIGNAL                            ;
```

```
                                    0C  11 000D5          BRB     5$                                                  ; 2040
                                    5B  DD 000D7 4$:     PUSHL   R11                                                 ; 2072
                00000000G   00      01  FB 000D9        CALLS   #1, SYS$OPEN                                        ;
                            6A      50  D0 000E0        MOVL    R0, PAT$GL_ERRCODE                                  ;
                            0B      6A  E8 000E3 5$:     BLBS    PAT$GL_ERRCODE, 7$                                  ; 2074
                                00000000G   EF  DD 000E6 6$:     PUSHL   PAT$GL_NEWRAB+12                            ; 2075
                                        08  AB  DD 000EC         PUSHL   PAT$GL_NEWFAB+8                             ;
                                    1A      11  000EF          BRB     8$                                          ;
                                00000000G   EF  9F 000F1 7$:     PUSHAB  PAT$GL_NEWRAB                               ; 2078
                00000000G   00      01  FB 000F7        CALLS   #1, SYS$CONNECT                                     ;
                            6A      50  D0 000FE        MOVL    R0, PAT$GL_ERRCODE                                  ;
                            23      6A  E8 00101        BLBS    PAT$GL_ERRCODE, 9$                                  ; 2079
                            7E  00000000G   EF  7D 00104        MOVQ    PAT$GL_NEWRAB+8, -(SP)                      ; 2082
                                    5B  DD 0010B 8$:     PUSHL   R11                                                 ; 2081
                00000000G   EF      01  FB 0010D        CALLS   #1, GETFILDSC                                       ;
                                    50  DD 00114        PUSHL   R0                                                  ;
                                    01  DD 00116        PUSHL   #1                                                  ;
                                00000000G   8F  DD 00118        PUSHL   #PAT$_OPENOUT                               ;
                00000000G   00      05  FB 0011E        CALLS   #5, LIB$SIGNAL                                      ;
                                    0E  11 00125        BRB     10$                                                 ;
                00000000'   EF      01  D0 00127 9$:     MOVL    #1, NUM_OF_UPDATES                                  ; 2085
                00000000G   EF      08  88 0012E        BISB2   #8, PAT$GL_FLAGS                                    ; 2086
                                00000000G   EF  95 00135 10$:    TSTB    PAT$GL_FLAGS                                ; 2094
                                    1E  18 0013B        BGEQ    11$                                                 ;
                                00000000G   EF  9F 0013D        PUSHAB  PAT$GB_NEWNAME                              ; 2095
                            7E  00000000G   EF  9A 00143        MOVZBL  PAT$GL_NEWNBK+3, -(SP)                      ;
                                    02  DD 0014A        PUSHL   #2                                                  ;
                                006D82C3    8F  DD 0014C        PUSHL   #7176899                                    ;
                00000000G   00      04  FB 00152        CALLS   #4, LIB$SIGNAL                                      ;
                                    20  11 00159        BRB     12$                                                 ;
                            6E  00000000G   EF  9B 0015B 11$:    MOVZBW  PAT$GL_OLDNBK+3, OLD_FILE                   ; 2100
                    04      AE  00000000G   EF  9E 00162        MOVAB   PAT$GB_OLDNAME, OLD_FILE+4                  ; 2101
                                    5E  DD 0016A        PUSHL   SP                                                  ; 2102
                                    01  DD 0016C        PUSHL   #1                                                  ;
                                00000000G   8F  DD 0016E        PUSHL   #PAT$_OVERLAY                               ;
                00000000G   00      03  FB 00174        CALLS   #3, LIB$SIGNAL                                      ;
                            52  00000000G   EF  D0 0017B 12$:    MOVL    PAT$GL_ISELHD, ISE_PTR                      ; 2105
                                    03  12 00182 13$:    BNEQ    14$                                                 ; 2106
                                010F    31 00184        BRW     26$                                                 ;
                            54      14  A2  9E 00187 14$:    MOVAB   20(R2), ISD_PTR                             ; 2108
                            55      10  A2  9E 0018B        MOVAB   16(ISE_PTR), R5                             ; 2113
                                00000000G   EF  95 0018F        TSTB    PAT$GL_FLAGS                                ; 2110
                                    03  19 00195        BLSS    15$                                                 ;
                                009D    31 00197        BRW     22$                                                 ;
                                    65  D5 0019A 15$:    TSTL    (R5)                                                ; 2113
                                    09  12 0019C        BNEQ    16$                                                 ;
                                    52  DD 0019E        PUSHL   ISE_PTR                                             ; 2114
                00000000G   EF      01  FB 001A0        CALLS   #1, PAT$CREMAP                                      ;
                            58      0C  A2  D0 001A7 16$:    MOVL    12(ISE_PTR), START_VA                       ; 2116
50                          65      0C  A2  C3 001AB        SUBL3   12(ISE_PTR), (R5), R0                       ; 2117
                            53      01  A0  9E 001B0        MOVAB   1(R0), SECTION_SIZE                         ;
                0000FE00    8F          53  D1 001B4        CMPL    SECTION_SIZE, #65024                        ; 2119
                                    07  15 001BB        BLEQ    17$                                                 ;
                            56      FE00    8F  B0 001BD        MOVW    #-512, BYTE_COUNT                           ; 2120
                                    03  11 001C2        BRB     18$                                                 ;
                            56              53  B0 001C4 17$:    MOVW    SECTION_SIZE, BYTE_COUNT                    ; 2121
                                    59  D4 001C7 18$:    CLRL    TRANSFER_COUNT                                      ; 2126
```

```
                                     57  D4 001C9         CLRL    REMAINDER                                     ; 2127
                      65             58  D1 001CB 19$:    CMPL    START_VA, (R5)                                ; 2128
                                     03  1F 001CE         BLSSU   20$                                           ;
                                   0091  31 001D0         BRW     23$                                           ;
              00000000G  EF          56  B0 001D3 20$:    MOVW    BYTE_COUNT, PAT$GL_NEWRAB+34                  ; 2130
              00000000G  EF          58  D0 001DA         MOVL    START_VA, PAT$GL_NEWRAB+40                    ; 2131
                         00000000G   EF  9F 001E1         PUSHAB  PAT$GL_NEWRAB                                 ; 2132
              00000000G  00          01  FB 001E7         CALLS   #1, SYS$WRITE                                 ;
                         6A          50  D0 001EE         MOVL    R0, PAT$GL_ERRCODE                            ;
                         21          50  E8 001F1         BLBS    R0, 21$                                       ;
                         7E 00000000G EF 7D 001F4         MOVQ    PAT$GL_NEWRAB+8, -(SP)                        ; 2134
                                     5B  DD 001FB         PUSHL   R11                                           ; 2133
              00000000G  EF          01  FB 001FD         CALLS   #1, GETFILDSC                                 ;
                                     50  DD 00204         PUSHL   R0                                            ;
                                     01  DD 00206         PUSHL   #1                                            ;
                         00000000G   8F  DD 00208         PUSHL   #PAT$_WRITEERR                                ;
              00000000G  00          05  FB 0020E         CALLS   #5, LIB$SIGNAL                                ;
                         50          56  3C 00215 21$:    MOVZWL  BYTE_COUNT, R0                                ; 2137
                         59          50  C0 00218         ADDL2   R0, TRANSFER_COUNT                            ;
                         50          56  3C 0021B         MOVZWL  BYTE_COUNT, R0                                ; 2138
                         58          50  C0 0021E         ADDL2   R0, START_VA                                  ;
           57            53          59  C3 00221         SUBL3   TRANSFER_COUNT, SECTION_SIZE, REMAINDER       ; 2140
              0000FE00   8F          57  D1 00225         CMPL    REMAINDER, #65024                             ;
                                     9D  14 0022C         BGTR    19$                                           ;
                                     57  D5 0022E         TSTL    REMAINDER                                     ; 2141
                                     99  15 00230         BLEQ    19$                                           ;
                         56          57  B0 00232         MOVW    REMAINDER, BYTE_COUNT                         ; 2142
                                     94  11 00235         BRB     19$                                           ; 2128
                                     65  D5 00237 22$:    TSTL    (R5)                                          ; 2147
                                     29  13 00239         BEQL    23$                                           ;
                                     7E  7C 0023B         CLRQ    -(SP)                                         ; 2148
                                     7E  7C 0023D         CLRQ    -(SP)                                         ;
                                     7E  7C 0023F         CLRQ    -(SP)                                         ;
                                     7E  D4 00241         CLRL    -(SP)                                         ;
                               0C    A2  9F 00243         PUSHAB  12(ISE_PTR)                                   ;
              00000000G  00          08  FB 00246         CALLS   #8, SYS$UPDSEC                                ;
                         6A          50  D0 0024D         MOVL    R0, PAT$GL_ERRCODE                            ;
                         11          50  E8 00250         BLBS    R0, 23$                                       ;
                                     6A  DD 00253         PUSHL   PAT$GL_ERRCODE                                ; 2149
                                     7E  D4 00255         CLRL    -(SP)                                         ;
                         00000000G   8F  DD 00257         PUSHL   #PAT$_SYSERROR                                ;
              00000000G  00          03  FB 0025D         CALLS   #3, LIB$SIGNAL                                ;
                                     65  D5 00264 23$:    TSTL    (R5)                                          ; 2154
                                     28  13 00266         BEQL    25$                                           ;
                                     7E  7C 00268         CLRQ    -(SP)                                         ; 2157
                               0C    A2  9F 0026A         PUSHAB  12(ISE_PTR)                                   ;
              00000000G  00          03  FB 0026D         CALLS   #3, SYS$DELTVA                                ;
                         6A          50  D0 00274         MOVL    R0, PAT$GL_ERRCODE                            ;
                         11          50  E8 00277         BLBS    R0, 24$                                       ;
                                     6A  DD 0027A         PUSHL   PAT$GL_ERRCODE                                ; 2158
                                     7E  D4 0027C         CLRL    -(SP)                                         ;
                         00000000G   8F  DD 0027E         PUSHL   #PAT$_SYSERROR                                ;
              00000000G  00          03  FB 00284         CALLS   #3, LIB$SIGNAL                                ;
                                     65  D4 0028B 24$:    CLRL    (R5)                                          ; 2159
                               0C    A2  D4 0028D         CLRL    12(ISE_PTR)                                   ;
                         52          62  D0 00290 25$:    MOVL    (ISE_PTR), ISE_PTR                            ; 2162
                                   FEEC  31 00293         BRW     13$                                           ; 2106
```

```
                                    00000000G  EF  95 00296 26$:     TSTB    PAT$GL_FLAGS                            ; 2170
                                               35  18 0029C         BGEQ    28$                                     ;
                                               5B  DD 0029E         PUSHL   R11                                     ; 2173
                          00000000G  00        01  FB 002A0         CALLS   #1, SYS$CLOSE                           ;
                                     6A        50  D0 002A7         MOVL    R0, PAT$GL_ERRCODE                      ;
                                     1F        6A  E8 002AA         BLBS    PAT$GL_ERRCODE, 27$                     ; 2174
                                     7E  08    AB  7D 002AD         MOVQ    PAT$GL_NEWFAB+8, -(SP)                  ; 2175
                                               5B  DD 002B1         PUSHL   R11                                     ;
                          00000000G  EF        01  FB 002B3         CALLS   #1, GETFILDSC                           ;
                                               50  DD 002BA         PUSHL   R0                                      ;
                                               01  DD 002BC         PUSHL   #1                                      ;
                                    00000000G  8F  DD 002BE         PUSHL   #PAT$_CLOSEOUT                          ;
                          00000000G  00        05  FB 002C4         CALLS   #5, LIB$SIGNAL                          ;
                                               04  002CB         RET                                             ;
                          00000000G  EF        08  8A 002CC 27$:    BICB2   #8, PAT$GL_FLAGS                        ; 2176
                                               04  002D3 28$:    RET                                             ; 2179

; Routine Size:  724 bytes,    Routine Base:  _PAT$CODE + 0CBE

; 1287          2180  1
; 1288          2181  1 END  ! of MODULE patwrt
; 1289          2182  0 ELUDOM


                                                                .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
;      Name                      Bytes                         Attributes
;
;  _PAT$OWN                          4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  _PAT$PLIT                       148  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(0)
;  _PAT$CODE                      3986  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       104         0      1000          00:01.8
;
;                                  COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/VARIANT:1/LIS=LIS$:PATWRT/OBJ=OBJ$:PATWRT MSRC$:PATWRT/UPDATE=(ENH$:PATWRT)
```

```
; Size:          3986 code + 152 data bytes
; Run Time:        01:12.4
; Elapsed Time:    04:11.1
; Lines/CPU Min:    1809
; Lexemes/CPU-Min: 27172
; Memory Used:  680 pages
; Compilation Complete
```



```
RMS
SCR
SCR
SCR
SCR
SCR
SCR
SCR
SCR
SCR
SS$
SS$
SS$
STR
STR
STR
SYS
SYS

SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS
SYS

SYS
SYS
SYS
SYS
SYS
SYS

UNS
```

0304 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PHONE

PATSYM
LIS

PHONE
MAP

BASICCMDS
LIS

PATSTO
LIS

PATVEC
LIS

PHONEREQ
REQ

PATWRT
LIS

FILECMDS
LIS