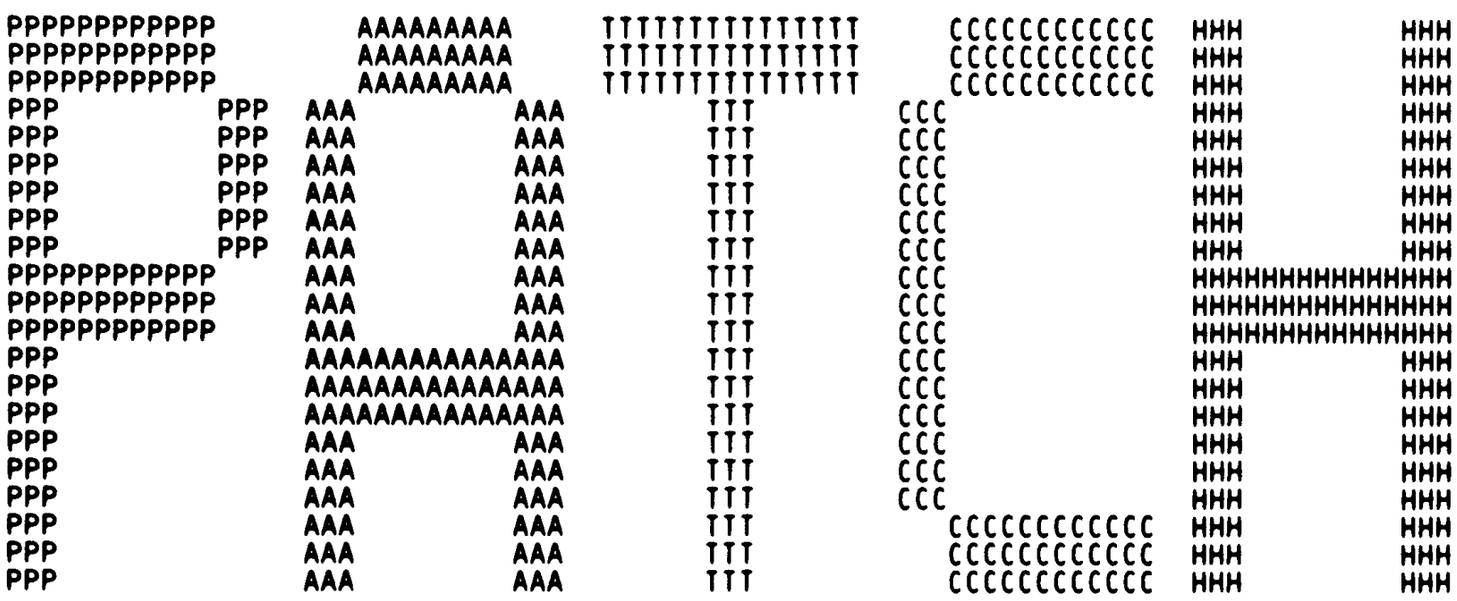
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPPPPPPPPPPP AAAAAAAAAAAAAAAA TTT CCC HHHHHHHHHHHHHHHH
PPP AAA AAA TTT CCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCCC HHH HHH

```
PPPPPPPP      AAAAAA    TTTTTTTTTT    IIIIII    HH      HH  DDDDDDDD
PPPPPPPP      AAAAAA    TTTTTTTTTT    IIIIII    HH      HH  DDDDDDDD
PP      PP   AA    AA       TT         II      HH      HH  DD      DD
PP      PP   AA    AA       TT         II      HH      HH  DD      DD
PP      PP   AA    AA       TT         II      HH      HH  DD      DD
PP      PP   AA    AA       TT         II      HH      HH  DD      DD
PPPPPPPP     AA    AA       TT         II      HHHHHHHHHH  DD      DD
PPPPPPPP     AA    AA       TT         II      HHHHHHHHHH  DD      DD
PP           AAAAAAAAAA     TT         II      HH      HH  DD      DD
PP           AAAAAAAAAA     TT         II      HH      HH  DD      DD
PP           AA    AA       TT         II      HH      HH  DD      DD      ....
PP           AA    AA       TT         II      HH      HH  DD      DD      ....
PP           AA    AA       TT       IIIIII    HH      HH  DDDDDDDD        ....
PP           AA    AA       TT       IIIIII    HH      HH  DDDDDDDD        ....

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 PATIHD (%IF %VARIANT EQL 1                                  ! ROUTINES TO HANDLE IMAGE HEADER AND SECTIO
;     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: ROUTINES TO HANDLE IMAGE HEADER DATA AND CREATE AND MAP IMAGE SECTIONS.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT: PART OF IMAGE FILE PATCH UTILITY
;    40          0040  1 !
;    41          0041  1 ! AUTHOR: K.D. MORSE     , CREATION DATE: 11-OCT-77
;    42          0042  1 !
;    43          0043  1 ! MODIFIED BY:
;    44          0044  1 !
;    45          0045  1 !       V03-005 MCN0157         Maria del C. Nasr       22-Feb-1984
;    46          0046  1 !               Use image activation routines to get image header and
;    47          0047  1 !               image section descriptors, instead of reading blocks
;    48          0048  1 !               directly.
;    49          0049  1 !
;    50          0050  1 !       V03-004 MCN0148         Maria del C. Nasr       6-Feb-1984
;    51          0051  1 !               Supply channel number address to LIB$_CREMAPSEC when
;    52          0052  1 !               doing a PATCH/ABSOLUTE/NONEW_VERSION.  In that way,
;    53          0053  1 !               the file is only opened once.
;    54          0054  1 !
;    55          0055  1 !       V03-003 MTR0025         Mike Rhodes             8-Aug-1983
;    56          0056  1 !               Add new routine BUILD_IHD to support the /ABSOLUTE feature.
;    57          0057  1 !               The /ABSOLUTE feature allows a user to patch ANY file via
```

```
;    58          0058  1 !               absolute (virtual) location values.
;    59          0059  1 !
;    60          0060  1 !       V03-002 MTR0016         Mike Rhodes             29-Oct-1982
;    61          0061  1 !               Modify PAT$GET_IHD to save the image type identifier word
;    62          0062  1 !               in the global scalar PAT$GW_IMGTYP (later restored by
;    63          0063  1 !               PAT$WRTIMG).
;    64          0064  1 !
;    65          0065  1 !       V03-001 MTR0007         Mike Rhodes             14-Jun-1982
;    66          0066  1 !               Use shared system messages.  Affected modules include:
;    67          0067  1 !               DYNMEM.B32, PATBAS.B32, PATCMD.B32, PATIHD.B32, PATINT.B32,
;    68          0068  1 !               PATIO.B32, PATMAI.B32, PATMSG.MSG, PATWRT.B32, and PATSPA.B32.
;    69          0069  1 !
;    70          0070  1 !               The shared messages are defined by DYNMEM.B32's invocation of
;    71          0071  1 !               SHRMSG.REQ and we simply link against these symbols.  They are
;    72          0072  1 !               declared as external literals below.
;    73          0073  1 !
;    74          0074  1 !       V02-008 MTR0001         Mike Rhodes             14-Oct-1981
;    75          0075  1 !               Modify routine PAT$CREMAP to allow the create and map
;    76          0076  1 !               section system service to do expand region mapping
;    77          0077  1 !               within P0 space.  This eliminates PATCH's trying to
;    78          0078  1 !               remember the last mapped address and getting errors
;    79          0079  1 !               due to stepping on RMS buffers in P0.
;    80          0080  1 !
;    81          0081  1 !       V02-007 KDM0044         Kathleen D. Morse       03-MAR-1981
;    82          0082  1 !               Reformat some comment lines.
;    83          0083  1 !
;    84          0084  1 !       V02-006 PCG0001         Peter George            02-FEB-1981
;    85          0085  1 !               Add require statement for LIB$:PATDEF.REQ
;    86          0086  1 !
;    87          0087  1 !
;    88          0088  1 !--
```

```
;    90          0089  1 !
;    91          0090  1 ! TABLE OF CONTENTS:
;    92          0091  1 !
;    93          0092  1
;    94          0093  1 FORWARD ROUTINE
;    95          0094  1         PAT$GET_IHD: NOVALUE,                   ! READS AND PROCESSES IMAGE HEADER
;    96          0095  1         PAT$CREMAP: NOVALUE,                    ! CREATES AND MAPS IMAGE SECTIONS
;    97          0096  1         BUILD_IHD: NOVALUE;                     ! BUILDS A PHONY IMAGE HEADER AND ISE LIST.
;    98          0097  1
;    99          0098  1 !
;   100          0099  1 ! INCLUDE FILES:
;   101          0100  1 !
;   102          0101  1
;   103          0102  1 LIBRARY 'SYS$LIBRARY:LIB.L32';                  ! SYSTEM STRUCTURE DEFINITIONS
;   104          0103  1 REQUIRE 'SRC$:PATPCT.REQ';                      ! DEFINES PSECTS
;   105          0143  1 REQUIRE 'SRC$:PREFIX.REQ';                      ! DEFINES STRUCTURE MACROS
;   106          0331  1 REQUIRE 'SRC$:PATPRE.REQ';                      ! DEFINES PATCH STRUCTURES
;   107          0494  1 REQUIRE 'LIB$:PATDEF.REQ';                      ! Defines literals
;   108          0548  1 REQUIRE 'LIB$:PATMSG.REQ';                      ! DEFINE ERROR CODES
;   109          0722  1 REQUIRE 'SRC$:PATGEN.REQ';                      ! DEFINE QUADWORD SIZE
;   110          0944  1 REQUIRE 'SRC$:SYSLIT.REQ';                      ! DEFINE TTY_OUT_WIDTH
;   111          0994  1 REQUIRE 'IMGMSGDEF';                            ! IMAGE ACTIVATOR DEF
;   112          1080  1
;   113          1081  1 !
;   114          1082  1 ! MACROS:
;   115          1083  1 !
;   116          1084  1
;   117          1085  1 !
;   118          1086  1 ! EQUATED SYMBOLS:
;   119          1087  1 !
;   120          1088  1
;   121          1089  1 LITERAL
;   122          1090  1         START_OFF = 0,                          ! OFFSET TO STARTING ADDRESS
;   123          1091  1         END_OFF = 1,                            ! OFFSET TO ENDING ADDRESS
;   124          1092  1         IDENT_SIZE = 2;                         ! SIZE OF IDENT FIELD
;   125          1093  1
;   126          1094  1 !
;   127          1095  1 ! EXTERNAL REFERENCES:
;   128          1096  1 !
;   129          1097  1
;   130          1098  1 EXTERNAL
;   131          1099  1         PAT$GL_IHPPTR: REF BLOCK[,BYTE],        ! POINTER TO IMAGE HEADER
;   132          1100  1         PAT$CP_OUT_STR,                         ! POINTER TO OUTPUT BUFFER
;   133          1101  1         PAT$GL_BUF_SIZ,                         ! SIZE OF MESSAGE IN OUTPUT BUFFER
;   134          1102  1         PAT$GL_CHANUM,                          ! CHANNEL NUMBER
;   135          1103  1         PAT$GL_EXPANDVA,                        ! FIRST EXPAND VA ADDRESS
;   136          1104  1         PAT$GL_PATAREA,                         ! POINTER TO PATCH AREA DESCRIPTOR
;   137          1105  1         PAT$GL_ERRCODE,                         ! GLOBAL ERROR CODE
;   138          1106  1         PAT$GL_FLAGS : BITVECTOR [32],          ! CLI RESULTANT PARSE FLAGS
;   139          1107  1         PAT$GL_ISVADDR: VECTOR[,LONG],          ! LAST MAPPED ADDRESS PAIR
;   140          1108  1         PAT$GB_INPBUF,                          ! INPUT BUFFER FOR IMAGE FILE
;   141          1109  1         PAT$GL_IMGHDR: REF BLOCK[,BYTE],        ! IMAGE HEADER DATA ADDRESS
;   142          1110  1         PAT$GL_ISELHD,                          ! IMAGE SECTION TABLE LIST HEAD
;   143          1111  1         PAT$GL_ISETAIL : REF BLOCK[,BYTE],      ! LAST ENTRY IN IMAGE SECTION TABLE
;   144          1112  1         PAT$GL_NEWVPNMX,                        ! MAX VPN IN IMAGE FILE
;   145          1113  1         PAT$GL_NEWVBNMX,                        ! MAX VBN IN IMAGE FILE
;   146          1114  1         PAT$GL_OLDVBNMX,                        ! OLD IMAGE FILE MAX ISD VBN
```

```
;  147     1115  1          PAT$GB_OLDNAME,                                  ! OLD IMAGE FILE NAME
;  148     1116  1          PAT$GL_OLDFAB: BLOCK[,BYTE],                     ! FAB FOR INPUT IMAGE FILE
;  149     1117  1          PAT$GL_OLDNBK: BLOCK[,BYTE],                     ! NAME BLOCK FOR INPUT IMAGE FILE
;  150     1118  1          PAT$GL_OLDRAB: BLOCK[,BYTE],                     ! RAB FOR INPUT IMAGE FILE
;  151     1119  1          PAT$GL_JNLRAB: BLOCK[,BYTE],                     ! RAB FOR JOURNAL FILE
;  152     1120  1          PAT$GW_IMGTYP: WORD;                             ! IMAGE TYPE IDENTIFIER
;  153     1121  1
;  154     1122  1  EXTERNAL ROUTINE
;  155     1123  1          PAT$ADD_PAL,                                     ! ADD PATCH AREA TO LIST
;  156     1124  1          PAT$BUILD_ISE,                                   ! BUILD AN IMAGE SECTION ENTRY AND ASSOC. IS
;  157     1125  1          PAT$FAO_PUT,                                     ! FORMATS OUTPUT MESSAGE
;  158     1126  1          PAT$WRITEFILE,                                   ! WRITES OUTPUT MESSAGE TO A FILE
;  159     1127  1          PAT$ALLOBLK,                                     ! ALLOCATES BLOCK OF FREE MEMORY
;  160     1128  1          GETFILDSC,                                       ! Returns the address of a file name descrip
;  161     1129  1          IMG$DECODE_IHD    : ADDRESSING_MODE (GENERAL),   ! Gets image header
;  162     1130  1          IMG$GET_NEXT_ISD : ADDRESSING_MODE (GENERAL),    ! Gets image section desc
;  163     1131  1          LIB$_CREMAPSEC;                                  ! CREATES AND MAPS SECTIONS
;  164     1132  1
;  165     1133  1  EXTERNAL LITERAL
;  166     1134  1  !
;  167     1135  1  ! Define shared message references.  (resolved @ link time)
;  168     1136  1  !
;  169     1137  1          PAT$_CLOSEIN,                                    ! Error closing input file.
;  170     1138  1          PAT$_CLOSEOUT,                                   ! Error closing output file.
;  171     1139  1          PAT$_OPENIN,                                     ! Error opening input file.
;  172     1140  1          PAT$_OPENOUT,                                    ! Error opening output file.
;  173     1141  1          PAT$_READERR,                                    ! Error reading from file.
;  174     1142  1          PAT$_SYSERROR,                                   ! System Service error.
;  175     1143  1          PAT$_WRITEERR;                                   ! Error writing to file.
;  176     1144  1
;  177     1145  1  !
;  178     1146  1  ! OWN STORAGE:
;  179     1147  1  !
;  180     1148  1
;  181     1149  1  OWN
;  182     1150  1          BLKBUF: BLOCK [512,BYTE],                        ! Buffer where image blocks are read into
;  183     1151  1          CURISD: REF BLOCK[,BYTE],                        ! Current image section descriptor (ISD)
;  184     1152  1          INPBUF: REF BLOCK[,BYTE] INITIAL(PAT$GB_INPBUF); ! Input buffer (used by IHD and ISD)
;  185     1153  1
;  186     1154  1  BIND
;  187     1155  1          MAJOR_IDENT= UPLIT (%ASCII'02'),                 ! VALUE OF MAJOR IDENT ABLE TO PATCH
;  188     1156  1          MINOR_IDENT= UPLIT (%ASCII'01');                 ! MINIMUM VALUE OF MINOR IDENT ABLE TO PATCH
```

```
;  190          1157  1 GLOBAL ROUTINE PAT$GET_IHD :NOVALUE =                         ! READS AND PROCESSES THE IMAGE HEADER
;  191          1158  1
;  192          1159  1 !++
;  193          1160  1 ! FUNCTIONAL DESCRIPTION:
;  194          1161  1 !
;  195          1162  1 !       THIS ROUTINE READS THE IMAGE HEADER AND STORES THE DATA.
;  196          1163  1 !       IT THEN PROCESSES EACH IMAGE SECTION, BUILDING THE IMAGE SECTION TABLE.
;  197          1164  1 !       THE IMAGE SECTION ENTRIES CONSIST OF:
;  198          1165  1 !
;  199          1166  1 !               1. LINK TO NEXT IMAGE SECTION ENTRY (ISE)
;  200          1167  1 !               2. STARTING MAPPED VIRTUAL ADDRESS
;  201          1168  1 !               3. ENDING MAPPED VIRTUAL ADDRESS
;  202          1169  1 !               4. STARTING IMAGE VIRTUAL ADDRESS WITHIN IMAGE SECTION
;  203          1170  1 !               5. ENDING IMAGE VIRTUAL ADDRESS WITHIN IMAGE SECTION
;  204          1171  1 !               6. IMAGE SECTION DESCRIPTOR AS READ FROM IMAGE HEADER
;  205          1172  1 !
;  206          1173  1 !       THE MAPPED ADDRESSES ARE INITIALIZED TO ZERO.  WHEN THE PAGES
;  207          1174  1 !       ARE ACCESSED BY A PATCH COMMAND, THE IMAGE SECTION IS MAPPED
;  208          1175  1 !       AND THE MAPPED ADDRESSES SET.
;  209          1176  1 !
;  210          1177  1 !       THE BEGINNING ADDRESS INTO WHICH THE IMAGE SECTIONS ARE TO
;  211          1178  1 !       BE MAPPED IS ALSO INITIALIZED.
;  212          1179  1 !
;  213          1180  1 ! FORMAL PARAMETERS:
;  214          1181  1 !
;  215          1182  1 !       NONE
;  216          1183  1 !
;  217          1184  1 ! IMPLICIT INPUTS:
;  218          1185  1 !
;  219          1186  1 !       THE OLD IMAGE FILE HAS BEEN SET UP FOR INPUT.
;  220          1187  1 !
;  221          1188  1 ! IMPLICIT OUTPUTS:
;  222          1189  1 !
;  223          1190  1 !       THE IMAGE SECTION TABLE IS SET UP AND THE IMAGE HEADER DATA STORED.
;  224          1191  1 !
;  225          1192  1 ! ROUTINE VALUE:
;  226          1193  1 !
;  227          1194  1 ! COMPLETION CODES:
;  228          1195  1 !
;  229          1196  1 !       NONE
;  230          1197  1 !
;  231          1198  1 ! SIDE EFFECTS:
;  232          1199  1 !
;  233          1200  1 !       NONE
;  234          1201  1 !
;  235          1202  1 !--
;  236          1203  1
;  237          1204  2 BEGIN
;  238          1205  2 BIND
;  239          1206  2         TIME_STR = UPLIT (%ASCIC 'DATE/TIME OF PATCH:          !%D'),
;  240          1207  2         ECO_MSG_STR = UPLIT (%ASCIC 'ECO LEVELS SET:           '),
;  241          1208  2         ECO_LVL_STR = UPLIT (%ASCIC '!UL, ');
;  242          1209  2
;  243          1210  2
;  244          1211  2 LITERAL
;  245          1212  2         NO_MORE_ISE = 65535;                                  ! CODE FOR NO MORE ISE'S ON THIS PAGE
;  246          1213  2
```

```
;  247     1214  2 MAP
;  248     1215  2         PAT$GL_IMGHDR : REF BLOCK[,BYTE];                       ! REFERENCE IMAGE HEADER AS A BLOCK
;  249     1216  2
;  250     1217  2 LOCAL
;  251     1218  2         ALIAS_ADR,
;  252     1219  2         ALLOCSIZE,                                              ! SIZE OF IMAGE HEADER TO ALLOCATE
;  253     1220  2         DESPTR,                                                 ! DESTINATION STRING POINTER
;  254     1221  2         ECO_PTR : REF BITVECTOR,                                ! POINTER TO ECO LEVEL BITS
;  255     1222  2         ECO_BIT,                                                ! ECO BIT LEVEL COUNTER
;  256     1223  2         HDRVER_ADR,
;  257     1224  2         ISD_ADR,                                                ! ADDR OF DECODED ISD
;  258     1225  2         ISEADR: REF BLOCK[,BYTE],                               ! CONTAINS ADDRESS OF IMAGE SECTION ENTRY
;  259     1226  2         NXTPTR,                                                 ! STRING POINTER TO NEXT PART OF RECORD
;  260     1227  2         OUT_BUFFER : VECTOR[TTY_OUT_WIDTH,BYTE],                ! OUTPUT MESSAGE BUFFER
;  261     1228  2         SRCPTR,                                                 ! SOURCE STRING POINTER
;  262     1229  2         VBN_ADR;                                                ! Image header VBN
;  263     1230  2
;  264     1231  2 !++
;  265     1232  2 ! INITIALIZE THE IMAGE SECTION TABLE LIST HEAD.
;  266     1233  2 !--
;  267     1234  2 PAT$GL_ISELHD=0;                                                ! SET END OF TABLE
;  268     1235  2 PAT$GL_ISETAIL=0;                                               ! SET TAIL OF TABLE
;  269     1236  2
;  270     1237  2 !++
;  271     1238  2 ! INITIALIZE MAXIMUM NEW FILE VIRTUAL PAGE AND BLOCK NUMBERS.
;  272     1239  2 !--
;  273     1240  2 PAT$GL_NEWVBNMX = 0;
;  274     1241  2 PAT$GL_NEWVPNMX = 0;
;  275     1242  2
;  276     1243  2 ! Read the first block of the image header and store the data.  If we are
;  277     1244  2 ! patching the file in absolute mode, create a phony image header and ISD's
;  278     1245  2 ! for mapping.  Otherwise, get the real image header.
;  279     1246  2 !
;  280     1247  2
;  281     1248  2 IF .PAT$GL_FLAGS [PAT$S_ABSOLUTE]
;  282     1249  2 THEN
;  283     1250  2     BUILD_IHD (.INPBUF)
;  284     1251  2 ELSE
;  285     1252  3     IF NOT (PAT$GL_ERRCODE = IMG$DECODE_IHD ( .PAT$GL_CHANUM,
;  286     1253  3                                               BLKBUF,
;  287     1254  3                                               PAT$GB_INPBUF,
;  288     1255  3                                               VBN_ADR,
;  289     1256  3                                               NXTPTR,
;  290     1257  3                                               HDRVER_ADR,
;  291     1258  3                                               ALIAS_ADR ))
;  292     1259  2     THEN
;  293     1260  2         SIGNAL(PAT$_READERR,1,GETFILDSC(PAT$GL_OLDFAB),.PAT$GL_ERRCODE,0);
;  294     1261  2
;  295     1262  2 !++
;  296     1263  2 ! CHECK THAT THE PATCH UTILITY UNDERSTANDS THIS TYPE OF IMAGE HEADER.
;  297     1264  2 !--
;  298     1265  2 IF .INPBUF[IHD$W_MAJORID] NEQU .MAJOR_IDENT
;  299     1266  2 THEN
;  300     1267  2         SIGNAL(PAT$_BADIDENT,2,IDENT_SIZE,INPBUF[IHD$W_MAJORID]);
;  301     1268  2 IF .INPBUF[IHD$W_MINORID] LSSU .MINOR_IDENT
;  302     1269  2 THEN
;  303     1270  2         SIGNAL(PAT$_BADIDENT,2,IDENT_SIZE,INPBUF[IHD$W_MINORID]);
```

```
;  304          1271  2
;  305          1272  2 ! Store the image header for future use.  The buffer will be pointed to
;  306          1273  2 ! by PAT$GL_IMGHDR.
;  307          1274  2 !
;  308          1275  2 ! If there is no patch section in the header, then enlarge it to include one.
;  309          1276  2 !
;  310          1277  2 ALLOCSIZE=.INPBUF[IHD$W_SIZE];                                  ! ASSUME HEADER IS CORRECT SIZE
;  311          1278  2 IF .INPBUF[IHD$W_PATCHOFF] EQL 0
;  312          1279  2 THEN
;  313          1280  2         ALLOCSIZE=.ALLOCSIZE + IHP$K_LFNGTH;
;  314          1281  2
;  315          1282  2 PAT$ALLOBLK(.ALLOCSIZE,PAT$GL_IMGHDR);                         ! ALLOCATE STORAGE FOR IMAGE HEADER
;  316          1283  2 SRCPTR=CH$PTR(.INPBUF,0);                                     ! SET POINTER TO INPUT DATA
;  317          1284  2 DESPTR=CH$PTR(.PAT$GL_IMGHDR,0);                            ! SET POINTER TO HEADER STORAGE
;  318          1285  2 CH$MOVE(.INPBUF[IHD$W_SIZE],.SRCPTR,.DESPTR);               ! STORE HEADER DATA
;  319          1286  2 PAT$GW_IMGTYP = .ALIAS_ADR;                                 ! SAVE THE IMAGE TYPE IDENTIFIER
;  320          1287  2
;  321          1288  2 IF .PAT$GL_IMGHDR[IHD$W_PATCHOFF] EQL 0                     ! IF THERE WAS NO PATCH SECTION
;  322          1289  2 THEN                                                        ! THEN
;  323          1290  3         BEGIN
;  324          1291  3         PAT$GL_IMGHDR[IHD$W_PATCHOFF]=.PAT$GL_IMGHDR[IHD$W_SIZE]; ! SET OFFSET TO PATCH SECTION
;  325          1292  3         PAT$GL_IMGHDR[IHD$W_SIZE]=.PAT$GL_IMGHDR[IHD$W_SIZE] + IHP$K_LENGTH; ! INCREMENT SIZE OF HEADER
;  326          1293  3         PAT$GL_IHPPTR=CH$PTR(.PAT$GL_IMGHDR, .PAT$GL_IMGHDR[IHD$W_PATCHOFF]);
;  327          1294  3         PAT$GL_IHPPTR[IHP$L_ECO1]=0;                        ! ZERO THE ECO LEVEL BITS
;  328          1295  3         PAT$GL_IHPPTR[IHP$L_ECO2]=0;                        ! ZERO THE ECO LEVEL BITS
;  329          1296  3         PAT$GL_IHPPTR[IHP$L_ECO3]=0;                        ! ZERO THE ECO LEVEL BITS
;  330          1297  3         PAT$GL_IHPPTR[IHP$L_ECO4]=0;                        ! ZERO THE ECO LEVEL BITS
;  331          1298  3         PAT$GL_IHPPTR[IHP$L_RW_PATADR]=0;                   ! SET NO READ-WRITE PATCH AREA
;  332          1299  3         PAT$GL_IHPPTR[IHP$L_RW_PATSIZ]=0;                   ! SET NO READ-WRITE PATCH AREA
;  333          1300  3         PAT$GL_IHPPTR[IHP$L_RO_PATADR]=0;                   ! SET NO READ-ONLY PATCH AREA
;  334          1301  3         PAT$GL_IHPPTR[IHP$L_RO_PATSIZ]=0;                   ! SET NO READ-ONLY PATCH AREA
;  335          1302  3         PAT$GL_IHPPTR[IHP$L_PATCOMTXT]=0;                   ! SET NO PATCH COMMAND TEXT
;  336          1303  3         END
;  337          1304  2 ELSE
;  338          1305  2         PAT$GL_IHPPTR=CH$PTR(.PAT$GL_IMGHDR, .PAT$GL_IMGHDR[IHD$W_PATCHOFF]);
;  339          1306  2
;  340          1307  2 PAT$GL_PATAREA = CH$PTR(PAT$GL_IHPPTR[IHP$L_RW_PATSIZ] ,0);     ! SET POINTER TO PATCH AREA DESCRIPTOR
;  341          1308  2 PAT$GL_ERRCODE=$GETTIM(TIMADR=PAT$GL_IHPPTR[IHP$Q_PATDATE]);    ! SET LATEST PATCH DATE
;  342          1309  2 IF NOT .PAT$GL_ERRCODE
;  343          1310  2 THEN
;  344          1311  2         SIGNAL(PAT$_SYSERROR,0,.PAT$GL_ERRCODE);                ! REPORT ERROR
;  345          1312  2
;  346          1313  2 !++
;  347          1314  2 ! WRITE OUT PATCH TIME TO JOURNAL FILE.
;  348          1315  2 !--
;  349          1316  2 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER,0);
;  350          1317  2 PAT$GL_BUF_SIZ = 0;
;  351          1318  2 PAT$FAO_PUT(TIME_STR, PAT$GL_IHPPTR[IHP$Q_PATDATE]);
;  352          1319  2 PAT$WRITEFILE(.PAT$GL_BUF_SIZ, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);
;  353          1320  2
;  354          1321  2 !++
;  355          1322  2 ! NOW OUTPUT THE ECO LEVELS SET TO THE JOURNAL FILE.
;  356          1323  2 !--
;  357          1324  2 ECO_PTR = CH$PTR(PAT$GL_IHPPTR[IHP$L_ECO1], 0);
;  358          1325  2 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER,0);
;  359          1326  2 PAT$GL_BUF_SIZ = 0;
;  360          1327  2 INCR ECO_BIT FROM PAT$K_MIN_ECO-1 TO PAT$K_MAX_ECO-1 BY 1
```

```
;  361          1328  2 DO
;  362          1329  3         BEGIN
;  363          1330  3         IF .ECO_PTR[.ECO_BIT]
;  364          1331  3         THEN
;  365          1332  4                 BEGIN
;  366          1333  4                 IF .PAT$GL_BUF_SIZ GEQ TTY_OUT_WIDTH-6
;  367          1334  4                 THEN
;  368          1335  5                         BEGIN
;  369          1336  5                         PAT$WRITEFILE(.PAT$GL_BUF_SIZ-2, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);
;  370          1337  5                         PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER,0);
;  371          1338  5                         PAT$GL_BUF_SIZ = 0;
;  372          1339  4                         END;
;  373          1340  4                 IF .PAT$GL_BUF_SIZ EQL 0
;  374          1341  4                 THEN
;  375          1342  4                         PAT$FAO_PUT(ECO_MSG_STR);
;  376          1343  4                 PAT$FAO_PUT(ECO_LVL_STR, .ECO_BIT+1);
;  377          1344  3                 END;
;  378          1345  2         END;
;  379          1346  2 IF .PAT$GL_BUF_SIZ NEQ 0
;  380          1347  2 THEN
;  381          1348  2         PAT$WRITEFILE(.PAT$GL_BUF_SIZ-2, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);
;  382          1349  2 PAT$WRITEFILE(0, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);          ! OUTPUT BLANK LINE
;  383          1350  2
;  384          1351  2 !++
;  385          1352  2 ! NOW PLACE THE DEFAULT PATCH AREA AS THE FIRST ENTRY ON THE PATCH AREA LIST.
;  386          1353  2 !--
;  387          1354  2 PAT$ADD_PAL(.PAT$GL_IHPPTR[IHP$L_RW_PATADR],
;  388          1355  2              .PAT$GL_IHPPTR[IHP$L_RW_PATADR]+.PAT$GL_IHPPTR[IHP$L_RW_PATSIZ],
;  389          1356  2             PAL$K_ADD_PAREA);
;  390          1357  2
;  391          1358  2 !++
;  392          1359  2 ! FIND THE BOTTOM OF THE PATCH CODE AND INITIALIZE THE CURRENT ISD ADDRESS IN
;  393          1360  2 ! THE IMAGE HEADER RECORD TO GET READY FOR IMAGE SECTION PROCESSING.
;  394          1361  2 !--
;  395        P 1362  2 PAT$GL_ERRCODE=$EXPREG(   PAGCNT=1
;  396          1363  2                        , RETADR=PAT$GL_ISVADDR);              ! FIND BOTTOM OF PATCH CODE
;  397          1364  2 IF NOT .PAT$GL_ERRCODE
;  398          1365  2 THEN
;  399          1366  2         SIGNAL(PAT$_SYSERROR,0,.PAT$GL_ERRCODE);                    ! REPORT ERROR
;  400          1367  2 PAT$GL_EXPANDVA = .PAT$GL_ISVADDR[START_OFF];                ! REMEMBER FIRST EXPAND ADDRESS
;  401          1368  2 PAT$GL_ISVADDR[END_OFF]=.PAT$GL_ISVADDR[START_OFF] - 1;     ! SET NEXT ADDRESS FOR CREMAP TO MAP OVER
;  402          1369  2
;  403          1370  2 !++
;  404          1371  2 ! Now process every image section descriptor from the image header when
;  405          1372  2 ! not in absolute mode.
;  406          1373  2 !--
;  407          1374  2 IF NOT .PAT$GL_FLAGS [PAT$S_ABSOLUTE]
;  408          1375  2 THEN
;  409          1376  3     BEGIN
;  410          1377  3
;  411          1378  3     LOCAL
;  412          1379  3         ALLOC_SIZE;                                     ! SIZE OF ISE TO ALLOCATE
;  413          1380  3
;  414          1381  3     WHILE 1 DO
;  415          1382  4         BEGIN
;  416          1383  4
;  417          1384  4         PAT$GL_ERRCODE = IMG$GET_NEXT_ISD ( .PAT$GL_CHANUM,
```

```
;  418     1385  4                                    BLKBUF,
;  419     1386  4                                    .PAT$GL_IMGHDR,
;  420     1387  4                                    VBN_ADR,
;  421     1388  4                                    NXTPTR,
;  422     1389  4                                    PAT$GB_INPBUF,
;  423     1390  4                                    HDRVER_ADR );
;  424     1391  4
;  425     1392  4          IF NOT .PAT$GL_ERRCODE
;  426     1393  4          THEN
;  427     1394  4              IF .PAT$GL_ERRCODE EQL IMG$_ENDOFHDR
;  428     1395  4              THEN
;  429     1396  4                  EXITLOOP
;  430     1397  4              ELSE
;  431     1398  4                  SIGNAL(PAT$_READERR,1,GETFILDSC(PAT$GL_OLDFAB),.PAT$GL_ERRCODE,0);
;  432     1399  4
;  433     1400  4          CURISD=.INPBUF;                                 ! SET CURRENT ISD ADDRESS
;  434     1401  4          ALLOC_SIZE = .CURISD[ISD$W_SIZE] + ISE$C_SIZE;  ! SET SIZE OF ISE
;  435     1402  4
;  436     1403  4          IF .CURISD[ISD$V_DZRO]
;  437     1404  4          THEN
;  438     1405  4                  ALLOC_SIZE = .ALLOC_SIZE + A_QUADWORD;  ! ADD IN EXTRA SPACE FOR VBN AND IDENT
;  439     1406  4          !++
;  440     1407  4          ! ****** THIS CHECK IS A FUTURE FEATURE WHICH WILL INSURE SPACE FOR AN IDENT
;  441     1408  4          ! ****** LONGWORD IN ANY PROCESS PRIVATE IMAGE SECTION DESCRIPTOR.
;  442     1409  4          ! ****** THIS WILL ENABLE PATCH TO CREATE AN IDENT FOR A PROCESS
;  443     1410  4          ! ****** PRIVATE IMAGE SECTION, WHICH WILL PROBABLY BE NECESSARY WHEN
;  444     1411  4          ! ****** GLOBAL SECTIONS ARE PATCHED AND BECOME PRIVATE SECTIONS.
;  445     1412  4          !--
;  446     1413  4          IF PAT$K_LENPRIV NEQ ISD$K_LENPRIV              ! CHECK IF LENGTH INCLUDES IDENT
;  447     1414  4          THEN
;  448     1415  4                  IF .CURISD[ISD$W_SIZE] EQL ISD$K_LENPRIV  ! CHECK IF THIS IS PROCESS PRIVATE ISD
;  449     1416  4                  THEN
;  450     1417  4                          ALLOC_SIZE = .ALLOC_SIZE + A_LONGWORD;  ! ADD IN IDENT LENGTH
;  451     1418  4          PAT$ALLOBLK(.ALLOC_SIZE,ISEADR);                ! GET IMAGE SECTION ENTRY
;  452     1419  4          DESPTR=CH$PTR(.ISEADR,ISE$C_SIZE);              ! SET DESTINATION POINTER
;  453     1420  4          CH$MOVE(.CURISD[ISD$W_SIZE],.INPBUF,.DESPTR);   ! MOVE IN ISD
;  454     1421  4
;  455     1422  4          IF .PAT$GL_ISETAIL EQLA 0                       ! IF FIRST ENTRY
;  456     1423  4          THEN
;  457     1424  5                  BEGIN
;  458     1425  5                  PAT$GL_ISELHD=CH$PTR(.ISEADR, 0);       ! SET TABLE LIST HEAD
;  459     1426  5                  PAT$GL_ISETAIL=CH$PTR(.ISEADR, 0);      ! SET TABLE TAIL
;  460     1427  5                  END
;  461     1428  4          ELSE
;  462     1429  5                  BEGIN
;  463     1430  5                  PAT$GL_ISETAIL[ISE$L_NXTISE]=CH$PTR(.ISEADR, 0); ! SET LINK TO THIS ISE
;  464     1431  5                  PAT$GL_ISETAIL=CH$PTR(.ISEADR, 0);      ! SET NEW TAIL OF TABLE
;  465     1432  4                  END;
;  466     1433  4          ISEADR[ISE$L_NXTISE]=0;                         ! SET FORWARD LINK
;  467     1434  4          ISEADR[ISE$L_IMGVST]=.CURISD[ISD$V_VPG]^9;      ! SET STARTING IMAGE VIRTUAL ADDRESS
;  468     1435  4          ISEADR[ISE$L_IMGVEND]=(.CURISD[ISD$V_VPG]+.CURISD[ISD$W_PAGCNT])^9 - 1; ! SET ENDING IMAGE VIRTUAL A
;  469     1436  4          ISEADR[ISE$L_MAPVST]=0;                         ! SET NO START MAPPED ADDRESS
;  470     1437  4          ISEADR[ISE$L_MAPVEND]=0;                        ! SET NO ENDING MAPPED ADDRESS
;  471     1438  4          IF .CURISD[ISD$B_TYPE] NEQ ISD$K_USRSTACK
;  472     1439  4          THEN
;  473     1440  4                  IF .PAT$GL_NEWVPNMX LSSU .CURISD[ISD$V_VPG]  ! SEE IF LARGER VPN
;  474     1441  4                  THEN
```

```
;  475          1442  4                          PAT$GL_NEWVPNMX = .CURISD[ISD$V_VPG] +.CURISD[ISD$W_PAGCNT] - 1;
;  476          1443  4          IF NOT .CURISD[ISD$V_DZRO]                          ! IF NOT DEMAND ZERO
;  477          1444  4          THEN                                               ! THEN CHECK IF LARGER VBN
;  478          1445  4                  IF .PAT$GL_NEWVBNMX LSS .CURISD[ISD$L_VBN]
;  479          1446  4                  THEN
;  480          1447  4                          PAT$GL_NEWVBNMX = .CURISD[ISD$L_VBN] + .CURISD[ISD$W_PAGCNT] - 1;
;  481          1448  3          END;
;  482          1449  2      END;
;  483          1450  2 PAT$GL_OLDVBNMX = .PAT$GL_NEWVBNMX;                        ! REMEMBER OLD FILE MAX ISD VBN USED
;  484          1451  1 END;                                                       ! END OF PAT$GET_IHD


                                                        .TITLE  PATIHD
                                                        .IDENT  \V04-000\

                                                        .PSECT  _PAT$PLIT,NOWRT,NOEXE,0

                                                00 00 32 30 00000 P.AAA:  .ASCII  \02\<0><0>                                ;
                                                00 00 31 30 00004 P.AAB:  .ASCII  \01\<0><0>                                ;
50 20 46 4F 20 45 4D 49 54 2F 45 54 41 44 18    00008 P.AAC:  .ASCII  <24>\DATE/TIME OF PATCH:\<9><9>\!%\                    ;
                  25 21 09 09 3A 48 43 54 41    00017                                                                       ;
                                                00 00 00 44 00020         .ASCII  \D\<0><0><0>                              ;
54 45 53 20 53 4C 45 56 45 4C 20 4F 43 45 11    00024 P.AAD:  .ASCII  <17>\ECO LEVELS SET:\<9><9><0><0>                      ;
                                                00 00 09 09 3A 00033                                                        ;
                        00 00 20 2C 4C 55 21 05 00038 P.AAE:  .ASCII  <5>\!UL, \<0><0>                                      ;

                                                        .PSECT  _PAT$OWN,NOEXE,2

                                                00000 BLKBUF: .BLKB   512
                                                00200 CURISD: .BLKB   4
                                      00000000G 00204 INPBUF: .ADDRESS PAT$GB_INPBUF                                        ;

                                                ISE$C_SIZE==        20
                                                TXT$C_SIZE==        4
                                                PAL$C_SIZE==        16
                                                ASD$C_SIZE==        9
                                                FWR$C_SIZE==        24
                                                MAJOR_IDENT=        P.AAA
                                                MINOR_IDENT=        P.AAB
                                                TIME_STR=           P.AAC
                                                ECO_MSG_STR=        P.AAD
                                                ECO_LVL_STR=        P.AAE
                                                        .EXTRN  PAT$GL_IHPPTR, PAT$CP_OUT_STR
                                                        .EXTRN  PAT$GL_BUF_SIZ, PAT$GL_CHANUM
                                                        .EXTRN  PAT$GL_EXPANDVA
                                                        .EXTRN  PAT$GL_PATAREA, PAT$GL_ERRCODE
                                                        .EXTRN  PAT$GL_FLAGS, PAT$GL_ISVADDR                                ;
                                                        .EXTRN  PAT$GB_INPBUF, PAT$GL_IMGHDR
                                                        .EXTRN  PAT$GL_ISELHD, PAT$GL_ISETAIL
                                                        .EXTRN  PAT$GL_NEWVPNMX
                                                        .EXTRN  PAT$GL_NEWVBNMX
                                                        .EXTRN  PAT$GL_OLDVBNMX
                                                        .EXTRN  PAT$GB_OLDNAME, PAT$GL_OLDFAB
                                                        .EXTRN  PAT$GL_OLDNBK, PAT$GL_OLDRAB
                                                        .EXTRN  PAT$GL_JNLRAB, PAT$GW_IMGTYP
                                                        .EXTRN  PAT$ADD_PAL, PAT$BUILD_ISE
                                                        .EXTRN  PAT$FAO_PUT, PAT$WRITEFILE
```

```
                                        .EXTRN  PAT$ALLOBLK, GETFILDSC
                                        .EXTRN  IMG$DECODE_IHD, IMG$GET_NEXT_ISD
                                        .EXTRN  LIB$_CREMAPSEC, PAT$_CLOSEIN
                                        .EXTRN  PAT$_CLOSEOUT, PAT$_OPENIN
                                        .EXTRN  PAT$_OPENOUT, PAT$_READERR
                                        .EXTRN  PAT$_SYSERROR, PAT$_WRITEERR
                                        .EXTRN  SYS$GETTIM, SYS$EXPREG

                                        .PSECT  _PAT$CODE,NOWRT,2

                          0FFC 00000    .ENTRY  PAT$GET_IHD, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 1157
                                                R10,R11
                5B 00000000G   EF 9E 00002      MOVAB   PAT$GL_ERRCODE, R11
                5A 00000000'   EF 9E 00009      MOVAB   INPBUF, R10
                5E       FF68  CE 9E 00010      MOVAB   -152(SP), SP
                   00000000G   EF D4 00015      CLRL    PAT$GL_ISELHD                       ; 1234
                   00000000G   EF D4 0001B      CLRL    PAT$GL_ISETAIL                      ; 1235
                   00000000G   EF D4 00021      CLRL    PAT$GL_NEWVBNMX                     ; 1240
                   00000000G   EF D4 00027      CLRL    PAT$GL_NEWVPNMX                     ; 1241
     0B 00000000G  EF           06 E1 0002D      BBC     #6, PAT$GL_FLAGS, 1$                ; 1248
        00000000V  EF           6A DD 00035      PUSHL   INPBUF                              ; 1250
                                01 FB 00037      CALLS   #1, BUILD_IHD
                                4A 11 0003E      BRB     2$
                                5E DD 00040 1$:  PUSHL   SP                                 ; 1252
                          08    AE 9F 00042      PUSHAB  HDRVER_ADR
                          10    AE 9F 00045      PUSHAB  NXTPTR
                          18    AE 9F 00048      PUSHAB  VBN_ADR
                   00000000G    EF 9F 0004B      PUSHAB  PAT$GB_INPBUF
                       FDFC     CA 9F 00051      PUSHAB  BLKBUF
                   00000000G    EF DD 00055      PUSHL   PAT$GL_CHANUM
                00000000G   00  07 FB 0005B      CALLS   #7, IMG$DECODE_IHD
                            68  50 D0 00062      MOVL    R0, PAT$GL_ERRCODE
                            22  50 E8 00065      BLBS    R0, 2$
                                7E D4 00068      CLRL    -(SP)                               ; 1260
                                6B DD 0006A      PUSHL   PAT$GL_ERRCODE
                   00000000G    EF 9F 0006C      PUSHAB  PAT$GL_OLDFAB
                00000000G   EF  01 FB 00072      CALLS   #1, GETFILDSC
                                50 DD 00079      PUSHL   R0
                                01 DD 0007B      PUSHL   #1
                   00000000G    8F DD 0007D      PUSHL   #PAT$_READERR
                00000000G   00  05 FB 00083      CALLS   #5, LIB$SIGNAL
                            50  6A D0 0008A 2$:  MOVL    INPBUF, R0                          ; 1265
00000000'  EF       0C  A0      10  00 ED 0008D      CMPZV   #0, #16, 12(R0), MAJOR_IDENT
                                14  13 00097      BEQL    3$
                          0C    A0 9F 00099      PUSHAB  12(R0)                              ; 1267
                                02 DD 0009C      PUSHL   #2
                                02 DD 0009E      PUSHL   #2
                   006D8164     8F DD 000A0      PUSHL   #7176548
                00000000G   00  04 FB 000A6      CALLS   #4, LIB$SIGNAL
                            50  6A D0 000AD 3$:  MOVL    INPBUF, R0                          ; 1268
00000000'  EF       0E  A0      10  00 ED 000B0      CMPZV   #0, #16, 14(R0), MINOR_IDENT
                                14  1E 000BA      BGEQU   4$
                          0E    A0 9F 000BC      PUSHAB  14(R0)                              ; 1270
                                02 DD 000BF      PUSHL   #2
                                02 DD 000C1      PUSHL   #2
                   006D8164     8F DD 000C3      PUSHL   #7176548
                00000000G   00  04 FB 000C9      CALLS   #4, LIB$SIGNAL
```

```
                                50              6A  D0 000D0 4$:    MOVL    INPBUF, R0                              ; 1277
                                51              60  3C 000D3        MOVZWL  (R0), ALLOCSIZE                         ;
                                        08      A0  B5 000D6        TSTW    8(R0)                                   ; 1278
                                                03  12 000D9        BNEQ    5$                                      ;
                                51              2C  C0 000DB        ADDL2   #44, ALLOCSIZE                          ; 1280
                                    00000000G   EF  9F 000DE 5$:    PUSHAB  PAT$GL_IMGHDR                           ; 1282
                                                51  DD 000E4        PUSHL   ALLOCSIZE                               ;
                            00000000G           EF  02  FB 000E6    CALLS   #2, PAT$ALLOBLK                         ;
                                50              6A  D0 000ED        MOVL    INPBUF, SRCPTR                          ; 1283
                                57  00000000G   EF  D0 000F0        MOVL    PAT$GL_IMGHDR, R7                       ; 1284
                                56              57  D0 000F7        MOVL    R7, DESPTR                              ;
                        66                      60      00  BA 28 000FA    MOVC3   @INPBUF, (SRCPTR), (DESPTR)     ; 1285
                            00000000G           EF      6E  B0 000FF    MOVW    ALIAS_ADR, PAT$GW_IMGTYP                ; 1286
                                                    08  A7  B5 00106    TSTW    8(R7)                                   ; 1288
                                                        2A  12 00109    BNEQ    6$                                      ;
                                                08  A7      67  B0 0010B    MOVW    (R7), 8(R7)                         ; 1291
                                                67          2C  A0 0010F    ADDW2   #44, (R7)                           ; 1292
                                                50      08  A7  3C 00112    MOVZWL  8(R7), R0                           ; 1293
                    00000000G   EF                  57          50  C1 00116    ADDL3   R0, R7, PAT$GL_IHPPTR           ;
                                                50  00000000G   EF  D0 0011E    MOVL    PAT$GL_IHPPTR, R0               ; 1294
                                                60              7C  00125        CLRQ    (R0)                            ;
                                                    08          A0  7C 00127    CLRQ    8(R0)                           ; 1296
                                                    10          A0  7C 0012A    CLRQ    16(R0)                          ; 1299
                                                    18          A0  7C 0012D    CLRQ    24(R0)                          ; 1301
                                                    20          A0  D4 00130    CLRL    32(R0)                          ; 1302
                                                                0C  11 00133    BRB     7$                              ; 1288
                                                50      08  A7  3C 00135 6$:    MOVZWL  8(R7), R0                       ; 1305
                    00000000G   EF                  57          50  C1 00139    ADDL3   R0, R7, PAT$GL_IHPPTR           ;
                    00000000G   EF  00000000G       EF          10  C1 00141 7$:    ADDL3   #16, PAT$GL_IHPPTR, PAT$GL_PATAREA  ; 1307
                                7E  00000000G       EF          24  C1 0014D        ADDL3   #36, PAT$GL_IHPPTR, -(SP)   ; 1308
                                    00000000G       00          01  FB 00155        CALLS   #1, SYS$GETTIM              ;
                                                    6B          50  D0 0015C        MOVL    R0, PAT$GL_ERRCODE          ;
                                                    11          6B  E8 0015F        BLBS    PAT$GL_ERRCODE, 8$          ; 1309
                                                                6B  DD 00162        PUSHL   PAT$GL_ERRCODE              ; 1311
                                                                7E  D4 00164        CLRL    -(SP)                       ;
                                                        00000000G   8F  DD 00166    PUSHL   #PAT$_SYSERROR              ;
                                    00000000G           00          03  FB 0016C    CALLS   #3, LIB$SIGNAL              ;
                                    00000000G           EF      14  AE  9E 00173 8$:    MOVAB   OUT_BUFFER, PAT$CP_OUT_STR  ; 1316
                                                        00000000G   EF  D4 0017B        CLRL    PAT$GL_BUF_SIZ              ; 1317
                                7E  00000000G           EF          24  C1 00181        ADDL3   #36, PAT$GL_IHPPTR, -(SP)   ; 1318
                                                        00000000'   EF  9F 00189        PUSHAB  TIME_STR                    ;
                                    00000000G           EF          02  FB 0018F    CALLS   #2, PAT$FAO_PUT             ;
                                                        00000000G   EF  9F 00196        PUSHAB  PAT$GL_JNLRAB               ; 1319
                                                                18  AE  9F 0019C        PUSHAB  OUT_BUFFER                  ;
                                                        00000000G   EF  DD 0019F        PUSHL   PAT$GL_BUF_SIZ              ;
                                    00000000G           EF          03  FB 001A5    CALLS   #3, PAT$WRITEFILE           ;
                                                        53  00000000G   EF  D0 001AC    MOVL    PAT$GL_IHPPTR, ECO_PTR      ; 1324
                                    00000000G           EF      14  AE  9E 001B3    MOVAB   OUT_BUFFER, PAT$CP_OUT_STR  ; 1325
                                                        00000000G   EF  D4 001BB        CLRL    PAT$GL_BUF_SIZ              ; 1326
                                                        52          D4  001C1        CLRL    ECO_BIT                     ; 1330
                    58                          63              52  E1  001C3 9$:    BBC     ECO_BIT, (ECO_PTR), 12$     ;
                                    0000007E    8F  00000000G   EF  D1  001C7        CMPL    PAT$GL_BUF_SIZ, #126        ; 1333
                                                                26  19  001D2        BLSS    10$                         ;
                                                        00000000G   EF  9F 001D4        PUSHAB  PAT$GL_JNLRAB               ; 1336
                                                                18  AE  9F 001DA        PUSHAB  OUT_BUFFER                  ;
                                7E  00000000G           EF          02  C3 001DD        SUBL3   #2, PAT$GL_BUF_SIZ, -(SP)   ;
                                    00000000G           EF          03  FB 001E5        CALLS   #3, PAT$WRITEFILE           ;
```

```
                00000000G  EF         14    AE  9E 001EC              MOVAB   OUT_BUFFER, PAT$CP_OUT_STR                       ; 1337
                                  00000000G  EF  D4 001F4              CLRL    PAT$GL_BUF_SIZ                                  ; 1338
                                  00000000G  EF  D5 001FA 10$:         TSTL    PAT$GL_BUF_SIZ                                  ; 1340
                                             0D  12 00200              BNEQ    11$                                             ;
                                  00000000'  EF  9F 00202              PUSHAB  ECO_MSG_STR                                     ; 1342
                00000000G  EF                01  FB 00208              CALLS   #1, PAT$FAO_PUT                                 ;
                                         01    A2  9F 0020F 11$:       PUSHAB  1(ECO_BIT)                                      ; 1343
                                  00000000'  EF  9F 00212              PUSHAB  ECO_LVL_STR                                     ;
                00000000G  EF                02  FB 00218              CALLS   #2, PAT$FAO_PUT                                 ;
            9C              52 0000007F      8F  F3 0021F 12$:       AOBLEQ  #127, ECO_BIT, 9$                               ; 1327
                            50 00000000G     EF  D0 00227              MOVL    PAT$GL_BUF_SIZ, R0                              ; 1346
                                             13  13 0022E              BEQL    13$                                             ;
                                  00000000G  EF  9F 00230              PUSHAB  PAT$GL_JNLRAB                                   ; 1348
                                         18    AE  9F 00236              PUSHAB  OUT_BUFFER                                      ;
                                         FE    A0  9F 00239              PUSHAB  -2(R0)                                          ;
                00000000G  EF                03  FB 0023C              CALLS   #3, PAT$WRITEFILE                               ;
                                  00000000G  EF  9F 00243 1 $.           PUSHAB  PAT$GL_JNLRAB                                   ; 1349
                                         18    AE  9F 00249              PUSHAB  OUT_BUFFER                                      ;
                                               7E  D4 0024C              CLRL    -(SP)                                           ;
                00000000G  EF                03  FB 0024E              CALLS   #3, PAT$WRITEFILE                               ;
                                               7E  D4 00255              CLRL    -(SP)                                           ; 1354
                            50 00000000G     EF  D0 00257              MOVL    PAT$GL_IHPPTR, R0                               ; 1355
            7E        14    A0         10    A0  C1 0025E              ADDL3   16(R0), 20(R0), -(SP)                           ;
                                         14    A0  DD 00264              PUSHL   20(R0)                                          ; 1354
                00000000G  EF                03  FB 00267              CALLS   #3, PAT$ADD_PAL                                 ;
                                               7E  7C 0026E              CLRQ    -(SP)                                           ; 1363
                                  00000000G  EF  9F 00270              PUSHAB  PAT$GL_ISVADDR                                  ;
                                               01  DD 00276              PUSHL   #1                                              ;
                00000000G  00                04  FB 00278              CALLS   #4, SYS$EXPREG                                  ;
                           6B                50  D0 0027F              MOVL    R0, PAT$GL_ERRCODE                              ;
                           11                6B  E8 00282              BLBS    PAT$GL_ERRCODE, 14$                             ; 1364
                                             6B  DD 00285              PUSHL   PAT$GL_ERRCODE                                  ; 1366
                                             7E  D4 00287              CLRL    -(SP)                                           ;
                                  00000000G  8F  DD 00289              PUSHL   #PAT$_SYSERROR                                  ;
                00000000G  00                03  FB 0028F              CALLS   #3, LIB$SIGNAL                                  ;
                00000000G  EF  00000000G     EF  D0 00296 14$:       MOVL    PAT$GL_ISVADDR, PAT$GL_EXPANDVA                 ; 1367
    00000000G   EF  00000000G  EF                01  C3 002A1              SUBL3   #1, PAT$GL_ISVADDR, PAT$GL_ISVADDR+4            ; 1368
                03  00000000G  EF                06  E1 002AD              BBC     #6, PAT$GL_FLAGS, 16$                           ; 1374
                                           0112  31  002B5 15$:         BRW     24$                                             ;
                                      04   AE  9F 002B8 16$:         PUSHAB  HDRVER_ADR                                      ; 1384
                                  00000000G  EF  9F 002BB              PUSHAB  PAT$GB_INPBUF                                   ;
                                         10    AE  9F 002C1              PUSHAB  NXTPTR                                          ;
                                         18    AE  9F 002C4              PUSHAB  VBN_ADR                                         ;
                                  00000000G  EF  DD 002C7              PUSHL   PAT$GL_IMGHDR                                   ; 1386
                                       FDFC  CA  9F 002CD              PUSHAB  BLKBUF                                          ; 1384
                                  00000000G  EF  DD 002D1              PUSHL   PAT$GL_CHANUM                                   ;
                00000000G  00                07  FB 002D7              CALLS   #7, IMG$GET_NEXT_ISD                            ;
                           6B                50  D0 002DE              MOVL    R0, PAT$GL_ERRCODE                              ;
                           2B                50  E8 002E1              BLBS    R0, 17$                                         ; 1392
                084D8640   8F                50  D1 002E4              CMPL    R0, #139298368                                  ; 1394
                                             C8  13 002EB              BEQL    15$                                             ;
                                             7E  D4 002ED              CLRL    -(SP)                                           ; 1398
                                             50  DD 002EF              PUSHL   R0                                              ;
                                  00000000G  EF  9F 002F1              PUSHAB  PAT$GL_OLDFAB                                   ;
                00000000G  EF                01  FB 002F7              CALLS   #1, GETFILDSC                                   ;
                                             50  DD 002FE              PUSHL   R0                                              ;
                                             01  DD 00300              PUSHL   #1                                              ;
```

```
                                00000000G   8F   DD 00302          PUSHL   #PAT$_READERR                                    ;
                        00000000G   00      05   FB 00308          CALLS   #5, LIB$SIGNAL                                  ;
                              FC    AA      6A   D0 0030F 17$:     MOVL    INPBUF, CURISD                                  ; 1400
                           50       FC  AA  D0 00313              MOVL    CURISD, R0                                      ; 1401
                           59           60  3C 00317              MOVZWL  (R0), ALLOC_SIZE                                ;
                           59           14  C0 0031A              ADDL2   #20, ALLOC_SIZE                                 ;
                     03        08      A0  02  E1 0031D              BBC     #2, 8(R0), 18$                                  ; 1403
                           59           08  C0 00322              ADDL2   #8, ALLOC_SIZE                                  ; 1405
                           10           60  B1 00325 18$:         CMPW    (R0), #16                                       ; 1415
                                        03  12 00328              BNEQ    19$                                             ;
                           59           04  C0 0032A              ADDL2   #4, ALLOC_SIZE                                  ; 1417
                                10      AE  9F 0032D 19$:         PUSHAB  ISEADR                                          ; 1418
                                        59  DD 00330              PUSHL   ALLOC_SIZE                                      ;
                        00000000G   EF      02  FB 00332          CALLS   #2, PAT$ALLOBLK                                 ;
                           58       10      AE  D0 00339          MOVL    ISEADR, R8                                      ; 1419
                           56       14      A8  9E 0033D          MOVAB   20(R8), DESPTR                                  ;
                           57       FC      AA  D0 00341          MOVL    CURISD, R7                                      ; 1420
                     66        00      BA      67 28 00345          MOVC3   (R7), @INPBUF, (DESPTR)                         ;
                           50 00000000G   EF      D0 0034A          MOVL    PAT$GL_ISETAIL, R0                              ; 1422
                                        09  12 00351              BNEQ    20$                                             ;
                        00000000G   EF      58  D0 00353          MOVL    R8, PAT$GL_ISELHD                               ; 1425
                                        03  11 0035A              BRB     21$                                             ; 1422
                           60           58  D0 0035C 20$:         MOVL    R8, (R0)                                        ; 1430
                        00000000G   EF      58  D0 0035F 21$:     MOVL    R8, PAT$GL_ISETAIL                              ; 1426
                                        68  D4 00366              CLRL    (R8)                                            ; 1433
          50        04  A7              17  00 EF 00368          EXTZV   #0, #23, 4(R7), R0                              ; 1434
          04        A8              50  09 78 0036E              ASHL    #9, R0, 4(R8)
          51        04  A7              17  00 EF 00373          EXTZV   #0, #23, 4(R7), R1                              ; 1435
                           50       02      A7  3C 00379          MOVZWL  2(R7), R0                                       ;
                           51           50  C0 0037D              ADDL2   R0, R1                                          ;
                     50                 51  09 78 00380          ASHL    #9, R1, R0                                      ;
                              08    A8      FF  A0 9E 00384          MOVAB   -1(R0), 8(R8)                                   ;
                                    0C  A8  7C 00389              CLRQ    12(R8)                                          ; 1436
                              FD    8F      0B  A7 91 0038C          CMPB    11(R7), #253                                    ; 1438
                                        14  13 00391              BEQL    22$                                             ;
00000000G  EF       04  A7              17  00 ED 00393          CMPZV   #0, #23, 4(R7), PAT$GL_NEWVPNMX                 ; 1440
                                        08  1B 0039D              BLEQU   22$                                             ;
                        00000000G   EF      FF  A1 9E 0039F          MOVAB   -1(R1), PAT$GL_NEWVPNMX                         ; 1442
                     1B        08  A7          02  E0 003A7 22$:     BBS     #2, 8(R7), 23$                                  ; 1443
                              0C    A7 00000000G  EF  D1 003AC          CMPL    PAT$GL_NEWVBNMX, 12(R7)                         ; 1445
                                        11  18 003B4              BGEQ    23$                                             ;
                           50       02      A7  3C 003B6          MOVZWL  2(R7), R0                                       ; 1447
                     57                 50  0C  A7 C1 003BA          ADDL3   12(R7), R0, R7                                  ;
                        00000000G   EF      FF  A7 9E 003BF          MOVAB   -1(R7), PAT$GL_NEWVBNMX                         ;
                                      FEEE  31  003C7 23$:         BRW     16$                                             ; 1381
                        00000000G   EF 00000000G  EF  D0 003CA 24$:     MOVL    PAT$GL_NEWVBNMX, PAT$GL_OLDVBNMX                ; 1450
                                            04  003D5              RET                                                     ; 1451

; Routine Size:  982 bytes,    Routine Base:  _PAT$CODE + 0000

;  485          1452  1
```

```
;  487          1453  1 %SBTTL  'build_ihd -- Build a phony image header for the input file'
;  488          1454  1 ROUTINE build_ihd (header_addr) : NOVALUE =
;  489          1455  1 !++
;  490          1456  1 !
;  491          1457  1 ! Functional Description:
;  492          1458  1 !
;  493          1459  1 !       This routine will build a phony image header for the input file
;  494          1460  1 !       being patched.  Thus allowing PATCH to map the input file as if
;  495          1461  1 !       it were an image.  The side effect of this will be that almost
;  496          1462  1 !       all PATCH commands will function as usual and the entry and
;  497          1463  1 !       display modes will operate normally.
;  498          1464  1 !
;  499          1465  1 !       We compute the number of ISDs required to map the input file and
;  500          1466  1 !       build the ISDs pointing to the various sections of the file.
;  501          1467  1 !       This is done by calling $GETJPI to determine how many PTEs are
;  502          1468  1 !       available for mapping the input file, adjusting by a fudge factor
;  503          1469  1 !       (the ISE/ISDs take up VM too), and finally arriving at our result
;  504          1470  1 !       by dividing the file size by the number of available PTEs.
;  505          1471  1 !
;  506          1472  1 ! Inputs:
;  507          1473  1 !
;  508          1474  1 !       header_addr     addr.rl The address of the buffer to write the
;  509          1475  1 !                               phony 'image' header.
;  510          1476  1 !
;  511          1477  1 ! Implicit Inputs:
;  512          1478  1 !
;  513          1479  1 !       pat$gl_iselhd   addr.ml The address of the image section entry
;  514          1480  1 !       pat$gl_isetail  addr.ml list head and tail (both 0 on entry!).
;  515          1481  1 !       pat$gl_oldfab   addr.rl The address of the input files FAB.
;  516          1482  1 !
;  517          1483  1 !       The image header offsets and associted 'default' values.
;  518          1484  1 !       The fill character used to pad out the remainder of the image
;  519          1485  1 !       header.
;  520          1486  1 !
;  521          1487  1 ! Outputs:
;  522          1488  1 !
;  523          1489  1 !       The header will be written into the output buffer.
;  524          1490  1 !       The ISDs have been created and written following the header.
;  525          1491  1 !
;  526          1492  1 ! Implicit Outputs:
;  527          1493  1 !
;  528          1494  1 !       pat$gl_iselhd   addr.ml The addresses of the first and last
;  529          1495  1 !       pat$gl_isetail  addr.ml entries in the image section list.
;  530          1496  1 !
;  531          1497  1 !       The fields in the phony image header will be adjusted as we
;  532          1498  1 !       create the necessary mapping information in the ISE list.
;  533          1499  1 !
;  534          1500  1 ! Routine Value:
;  535          1501  1 !
;  536          1502  1 !       None, yet.
;  537          1503  1 !
;  538          1504  1 ! Side Effects:
;  539          1505  1 !
;  540          1506  1 !       None, yet.
;  541          1507  1 !
;  542          1508  1 !--
;  543          1509  2 BEGIN
```

```
;  544          1510  2
;  545          1511  2 BIND
;  546          1512  2     block_count = pat$gl_oldfab [fab$l_alq];          ! Number of blocks in the input file.
;  547          1513  2
;  548          1514  2 LITERAL
;  549          1515  2     false = 0, true = 1,                              ! Boolean operands.
;  550          1516  2     fill = %X'FF';                                    ! Fill value for the remainder of the header block.
;  551          1517  2
;  552          1518  2 LOCAL
;  553          1519  2     current_ise : REF $bblock,                        ! Pointer to current ISE/ISD block.
;  554          1520  2     current_isd : REF $bblock,                        ! The address of the ISD we are currently working on
;  555          1521  2     freptecnt : INITIAL (0),                          ! Number of PTEs available for mapping VM in this pr
;  556          1522  2     page_count,                                       ! Number of pages mapped by each ISD.
;  557          1523  2     status,                                           ! Local status scalar.
;  558          1524  2     jpilist :   $bblock [12] INITIAL (                ! The item list for the $GETJPI system service.
;  559          1525  2                 WORD (4, jpi$_freptecnt),             ! Buffer size and item code (free pte count)
;  560          1526  2                 LONG (freptecnt, 0));                 ! The buffer and return addresses.
;  561          1527  2
;  562          1528  2 MAP
;  563          1529  2     header_addr : REF $bblock;
;  564          1530  2
;  565          1531  3 IF NOT (status = $GETJPI (ITMLST = jpilist))          ! Get the number of free PTEs for mapping the input
;  566          1532  2 THEN    SIGNAL (pat$_syserror, 0, .status);           ! Signal system service errors.
;  567          1533  2
;  568          1534  2 freptecnt = .freptecnt / 2;                           ! Reduce available page count by 50%.
;  569          1535  2 page_count = MINU (.block_count, .freptecnt);         ! Determine the number of pages mapped by each ISD.
;  570          1536  2
;  571          1537  2 CH$FILL (fill, a_page, .header_addr);                 ! Initialize the contents of the header.
;  572          1538  2 header_addr [ihd$w_size] =      a_page;              ! Size of 'image' header in bytes.
;  573          1539  2 header_addr [ihd$w_activoff] =  0;                    ! Offset to the 'image' activation data.
;  574          1540  2 header_addr [ihd$w_symdbgoff] = 0;                    ! Offset to the debug symbol table data.
;  575          1541  2 header_addr [ihd$w_imgidoff] =  0;                    ! Offset to the ident data.
;  576          1542  2 header_addr [ihd$w_patchoff] =  0;                    ! Offset to the patch data.
;  577          1543  2 header_addr [ihd$w_majorid] =   ihd$k_majorid;        ! Major identification.
;  578          1544  2 header_addr [ihd$w_minorid] =   ihd$k_minorid;        ! Minor identification.
;  579          1545  2 header_addr [ihd$b_hdrblkcnt] = 1;                    ! Number of blocks making up the 'image' header.
;  580          1546  2 header_addr [ihd$b_imgtype] =   ihd$k_exe;           ! Image type.
;  581          1547  2 header_addr [ihd$w_iochancnt] = 0;                    ! # of requested IO channels (0 is default).
;  582          1548  2 header_addr [ihd$w_imgiocnt] =  0;                    ! # of pages of image IO section requested (0 is def
;  583          1549  2 header_addr [ihd$l_lnkflags] =  0;                    ! Linker produced image flags (defaulted).
;  584          1550  2 header_addr [ihd$l_ident] =     0;                    ! Global section ident for linkable images.
;  585          1551  2 header_addr [ihd$l_sysver] =    0;                    ! System Version (0 default NOT linked with exec).
;  586          1552  2 header_addr [ihd$l_iafva] =     0;                    ! Relative virtual address of image activator fixup
;  587          1553  2
;  588          1554  2 !+
;  589          1555  2 ! We will always need at least one ISE/ISD.  It is created here to initialize the ISE list
;  590          1556  2 ! thus allowing routine pat$build_ise to perform the routine task of creating normal image
;  591          1557  2 ! section descriptors.  It will be called iteratively below to create enough ISDs to map
;  592          1558  2 ! the input file.
;  593          1559  2 !-
;  594          1560  2 pat$alloblk (ise$c_size+isd$k_lenpriv, current_ise);  ! Allocate a block of storage for the ISE/ISD.
;  595          1561  2 current_ise [ise$l_nxtise] =    0;                    ! Address of the next ISE list entry.
;  596          1562  2 current_ise [ise$l_imgvst] =    0;                    ! Starting virtual address of the section.
;  597          1563  2 current_ise [ise$l_imgvend] =   (.page_count*a_page)-1; ! Ending virtual address of the section.
;  598          1564  2 current_ise [ise$l_mapvst] =    0;                    ! Mapped starting address of this section.
;  599          1565  2 current_ise [ise$l_mapvend] =   0;                    ! Mapped ending address of this section.
;  600          1566  2 current_isd = .current_ise + ise$c_size;             ! Compute starting address of the ISD for this ISE.
```

```
;  601          1567  2 current_isd [isd$w_size] =      isd$k_lenpriv;          ! Size of the ISD in bytes for a GLOBAL ISD.
;  602          1568  2 current_isd [isd$w_pagcnt] =    .page_count;            ! # pages described by this ISD.
;  603          1569  2 current_isd [isd$l_vpnpfc] =    0;                      ! VPN and PFC fields.
;  604          1570  2 current_isd [isd$l_flags] =     0;                      ! Flags and ISD Type.
;  605          1571  2 current_isd [isd$v_crf] =       true;                   ! Copy on Reference.
;  606          1572  2 current_isd [isd$v_wrt] =       true;                   ! Writable section..
;  607          1573  2 current_isd [isd$v_matchctl] =  isd$k_matnev;           ! Match control.
;  608          1574  2 current_isd [isd$b_type] =      isd$k_normal;           ! Normal image section.
;  609          1575  2 current_isd [isd$l_vbn] =       1;                      ! Base virtual block number.
;  610          1576  2
;  611          1577  2 IF .pat$gl_iselhd EQL 0                                ! If this is the first entry in the list, make
;  612          1578  2 THEN    pat$gl_iselhd = pat$gl_isetail = .current_ise;  ! it appear at the head and tail of the list.
;  613          1579  2
;  614          1580  2 !+
;  615          1581  2 ! Now iteratively create enough ISE/ISDs to map the entire input file.
;  616          1582  2 !-
;  617          1583  2 INCR current_vpn FROM .page_count TO .block_count - 1 BY .page_count
;  618          1584  3 DO  BEGIN
;  619          1585  3     pat$build_ise (current_ise, .current_vpn, .current_vpn + 1, .page_count);
;  620          1586  3     current_ise [ise$l_imgvst] = .current_vpn * a_page;                  ! Set the starting and ending virtua
;  621          1587  3     current_ise [ise$l_imgvend] = .current_ise [ise$l_imgvst] + (.page_count * a_page) - 1;     ! for the se
;  622          1588  3     pat$gl_newvpnmx = .current_vpn;                                      ! Update the max VPN processed.
;  623          1589  3     pat$gl_newvbnmx = .current_vpn + 1;                                  ! Likewise with the VBN.
;  624          1590  2     END;    ! of INCR
;  625          1591  2
;  626          1592  2 pat$gl_oldvbnmx = .pat$gl_newvbnmx;                                    ! Set the highest VBN for the old fi
;  627          1593  2
;  628          1594  1 END;    ! of ROUTINE build_ihd


                                                .PSECT  _PAT$PLIT,NOWRT,NOEXE,0

                        0415  0004  00040 P.AAF:  .WORD   4, 1045                                ;
                00000000  00000000  00044         .LONG   0, 0                                  ;

                                                .EXTRN  SYS$GETJPI

                                                .PSECT  _PAT$CODE,NOWRT,2

                              07FC 00000 BUILD_IHD:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10      ; 1454
                        5A 00000000G   EF   9E 00002    MOVAB   PAT$GL_NEWVBNMX, R10          ;
                        59 00000000G   EF   9E 00009    MOVAB   PAT$GL_ISELHD, R9             ;
                        58 00000000G   EF   9E 00010    MOVAB   BLOCK_COUNT, R8               ;
                        5E             10   C2 00017    SUBL2   #16, SP                       ;
                                       7E   D4 0001A    CLRL    FREPTECNT                     ; 1509
                08   AE 00000000'      EF   0C 28 0001C MOVC3   #12, P.AAF, JPILIST           ; 1526
                             0C        AE   6E 9E 00025 MOVAB   FREPTECNT, JPILIST+4          ; 1509
                                       7E   7C 00029    CLRQ    -(SP)                        ; 1531
                                       7E   D4 0002B    CLRL    -(SP)                        ;
                                  14   AE   9F 0002D    PUSHAB  JPILIST                      ;
                                       7E   7C 00030    CLRQ    -(SP)                        ;
                                       7E   D4 00032    CLRL    -(SP)                        ;
                    00000000G          00   FB 00034    CALLS   #7, SYS$GETJPI               ;
                                       11   50 E8 0003B BLBS    STATUS, 1$                   ;
                                            50 DD 0003E PUSHL   STATUS                       ; 1532
```

```
                                                7E  D4 00040          CLRL    -(SP)                                          ;
                                     00000000G  8F  DD 00042          PUSHL   #PAT$_SYSERROR                                 ;
                          00000000G  00         03  FB 00048          CALLS   #3, LIB$SIGNAL                                 ;
                                     6E         02  C6 0004F 1$:      DIVL2   #2, FREPTECNT                                  ; 1534
                                     50         68  D0 00052          MOVL    BLOCK_COUNT, R0                                ; 1535
                                     6E         50  D1 00055          CMPL    R0, FREPTECNT                                  ;
                                                03  1B 00058          BLEQU   2$                                             ;
                                     50         6E  D0 0005A          MOVL    FREPTECNT, R0                                  ;
                                     56         50  D0 0005D 2$:      MOVL    R0, PAGE_COUNT                                 ;
                                     57    04   AC  D0 00060          MOVL    HEADER_ADDR, R7                                ; 1537
    0200  8F      FF  8F         6E         00  2C 00064          MOVC5   #0, (SP), #255, #512, (R7)                     ;
                                            67       0006C                                                                 ;
                                     67  0200   8F  3C 0006D          MOVZWL  #512, (R7)                                     ; 1538
                                         04     A7  D4 00072          CLRL    4(R7)                                          ; 1540
                                         08     A7  B4 00075          CLRW    8(R7)                                          ; 1542
                                   0C    A7 35303230  8F  D0 00078          MOVL    #892351024, 12(R7)                     ; 1543
                                   10    A7     0101  8F  B0 00080          MOVW    #257, 16(R7)                           ; 1545
                                         1C     A7  7C 00086          CLRQ    28(R7)                                         ; 1547
                                         24     A7  7C 00089          CLRQ    36(R7)                                         ; 1550
                                         2C     A7  D4 0008C          CLRL    44(R7)                                         ; 1552
                                         04     AE  9F 0008F          PUSHAB  CURRENT_ISE                                    ; 1560
                                                24  DD 00092          PUSHL   #36                                            ;
                          00000000G      EF         02  FB 00094          CALLS   #2, PAT$ALLOBLK                                ;
                                     51  04     AE  D0 0009B          MOVL    CURRENT_ISE, R1                                ; 1561
                                                61  7C 0009F          CLRQ    (R1)                                           ;
                    54                   56         09  78 000A1          ASHL    #9, PAGE_COUNT, R4                             ; 1563
                                   08    A1     FF  A4 9E 000A5          MOVAB   -1(R4), 8(R1)                                  ;
                                         0C     A1  7C 000AA          CLRQ    12(R1)                                         ; 1564
                                     50  14     A1  9E 000AD          MOVAB   20(R1), CURRENT_ISD                            ; 1566
                                     60         10  B0 000B1          MOVW    #16, (CURRENT_ISD)                             ; 1567
                                   02    A0         56  B0 000B4          MOVW    PAGE_COUNT, 2(CURRENT_ISD)                     ; 1568
                                         04     A0  7C 000B8          CLRQ    4(CURRENT_ISD)                                 ; 1569
                                   08    A0         0A  88 000BB          BISB2   #10, 8(CURRENT_ISD)                            ; 1572
    08  A0                 03                   04  03  F0 000BF          INSV    #3, #4, #3, 8(CURRENT_ISD)                     ; 1573
                                         0B     A0  94 000C5          CLRB    11(CURRENT_ISD)                                ; 1574
                                   0C    A0         01  D0 000C8          MOVL    #1, 12(CURRENT_ISD)                            ; 1575
                                                69  D5 000CC          TSTL    PAT$GL_ISELHD                                  ; 1577
                                                0A  12 000CE          BNEQ    3$                                             ;
                          00000000G      EF         51  D0 000D0          MOVL    R1, PAT$GL_ISETAIL                             ; 1578
                                         69         51  D0 000D7          MOVL    R1, PAT$GL_ISELHD                              ;
                    53                   68         01  C3 000DA 3$:      SUBL3   #1, BLOCK_COUNT, R3                            ; 1583
                                         52         56  D0 000DE          MOVL    PAGE_COUNT, CURRENT_VPN                        ;
                                                    32  11 000E1          BRB     5$                                             ;
                                                    56  DD 000E3 4$:      PUSHL   PAGE_COUNT                                     ; 1585
                                               01   A2  9F 000E5          PUSHAB  1(CURRENT_VPN)                                 ;
                                                    52  DD 000E8          PUSHL   CURRENT_VPN                                    ;
                                               10   AE  9F 000EA          PUSHAB  CURRENT_ISE                                    ;
                          00000000G      EF         04  FB 000ED          CALLS   #4, PAT$BUILD_ISE                              ;
                                         50    04   AE  D0 000F4          MOVL    CURRENT_ISE, R0                                ; 1586
                    04    A0                   52  09  78 000F8          ASHL    #9, CURRENT_VPN, 4(R0)                         ;
                    51                   54    04   A0  C1 000FD          ADDL3   4(R0), R4, R1                                  ; 1587
                                   08    A0         FF  A1 9E 00102          MOVAB   -1(R1), 8(R0)                                  ;
                          00000000G      EF         52  D0 00107          MOVL    CURRENT_VPN, PAT$GL_NEWVPNMX                   ; 1588
                                         6A    01   A2  9E 0010E          MOVAB   1(CURRENT_VPN), PAT$GL_NEWVBNMX                ; 1589
                                         52         56  C0 00112          ADDL2   PAGE_COUNT, CURRENT_VPN                        ; 1583
                                         53         52  D1 00115 5$:      CMPL    CURRENT_VPN, R3                                ;
                                                    C9  15 00118          BLEQ    4$                                             ;
```

```
                     00000000G  EF              6A  D0 0011A         MOVL    PAT$GL_NEWVBNMX, PAT$GL_OLDVBNMX          ; 1592
                                                     04 00121         RET                                               ; 1594

; Routine Size:  290 bytes,    Routine Base:  _PAT$CODE + 03D6

;  629          1595  1
```

```
;  631          1596  1 %SBTTL  'PAT$CREMAP -- MAP SECTIONS'
;  632          1597  1 GLOBAL ROUTINE PAT$CREMAP(ISEADR) :NOVALUE =                    ! CREATES AND "APS AN IMAGE SECTION
;  633          1598  1
;  634          1599  1 !++
;  635          1600  1 ! FUNCTIONAL DESCRIPTION:
;  636          1601  1 !
;  637          1602  1 !       THIS ROUTINE CREATES AND MAPS IMAGE SECTIONS.  ALL SECTIONS
;  638          1603  1 !       ARE MAPPED WRITABLE.  GLOBAL SECTIONS WHICH MUST ALWAYS BE
;  639          1604  1 !       OBTAINED EXTERNALLY, WILL NOT BE MAPPED (AS PATCH HAS NO WAY TO
;  640          1605  1 !       OBTAIN THEM) AND INSTEAD, AN ERROR IS REPORTED.  IF THERE
;  641          1606  1 !       IS A LOCAL COPY OF THE GLOBAL SECTION, THEN THAT WILL BE PATCHED
;  642          1607  1 !       NO MATTER WHAT THE MATCH CONTROL FIELD CONTAINS.
;  643          1608  1 !
;  644          1609  1 !       THE IMAGE SECTION DESCRIBING THE STACK MAY NOT BE PATCHED.
;  645          1610  1 !
;  646          1611  1 !       THE IMAGE SECTION ENTRY IS UPDATED TO CONTAIN THE MAPPED VIRTUAL ADDRESSES.
;  647          1612  1 !
;  648          1613  1 !       THE SECTION IS MAPPED INTO THE FIRST AVAILABLE ADDRESSES IN P0 SPACE.
;  649          1614  1 !
;  650          1615  1 ! FORMAL PARAMETERS:
;  651          1616  1 !
;  652          1617  1 !       CURISE - ADDRESS OF IMAGE SECTION ENTRY TO BE MAPPED
;  653          1618  1 !
;  654          1619  1 ! IMPLICIT INPUTS:
;  655          1620  1 !
;  656          1621  1 !       THE OLD IMAGE FILE HAS BEEN SET UP FOR INPUT.
;  657          1622  1 !
;  658          1623  1 ! IMPLICIT OUTPUTS:
;  659          1624  1 !
;  660          1625  1 !       THE IMAGE SECTION IS MAPPED AND THE IMAGE SECTION ENTRY UPDATED.
;  661          1626  1 !
;  662          1627  1 ! ROUTINE VALUE:
;  663          1628  1 !
;  664          1629  1 ! COMPLETION CODES:
;  665          1630  1 !
;  666          1631  1 !       NONE
;  667          1632  1 !
;  668          1633  1 ! SIDE EFFECTS:
;  669          1634  1 !
;  670          1635  1 !       ERROR MESSAGE IS REPORTED IF UNABLE TO MAP OR MAP FAILS.
;  671          1636  1 !       IMAGE SECTION IS MAPPED.
;  672          1637  1 !
;  673          1638  1 !--
;  674          1639  1
;  675          1640  2 BEGIN
;  676          1641  2
;  677          1642  2 LOCAL
;  678          1643  2         IMG_FIL_DESC: VECTOR[2,LONG],                          ! TEMPORARY IMAGE FILE NAME DESCRIPTOR
;  679          1644  2         CURISD: REF BLOCK[,BYTE];                              ! CONTAINS ADDRESS OF ISD PART OF ISE
;  680          1645  2 MAP
;  681          1646  2         ISEADR: REF BLOCK[,BYTE];                              ! CONTAINS ADDRESS OF IMAGE SECTION ENTRY
;  682          1647  2
;  683          1648  2 !++
;  684          1649  2 ! SET UP THE VIRTUAL PAGES TO MAP IMAGE SECTION TO.
;  685          1650  2 !--
;  686          1651  2 CURISD=.ISEADR + ISE$C_SIZE;                                   ! SET ADDRESS OF ISD PART OF ISE
;  687          1652  2 PAT$GL_ISVADDR[START_OFF]= 200;                                ! START MAPPING AT FISRT AVAILABLE ADDRESS I
```

```
;  688     1653  2 PAT$GL_ISVADDR[END_OFF]= 200;                                  ! MAP AS MUCH AS NEEDED (.CURISD[ISD$W_PAGCN
;  689     1654  2
;  690     1655  2 IMG_FIL_DESC[0]=.PAT$GL_OLDNBK[NAM$B_RSL];                     ! SET LENGTH OF NAME
;  691     1656  2 IMG_FIL_DESC[1]=PAT$GB_OLDNAME;                               ! SET ADDRESS OF NAME
;  692     1657  2
;  693     1658  2 !++
;  694     1659  2 ! CHECK FOR STACK IMAGE SECTION DESCRIPTOR.
;  695     1660  2 !--
;  696     1661  2 IF .CURISD[ISD$B_TYPE] EQL ISD$K_USRSTACK
;  697     1662  2 THEN
;  698     1663  2         SIGNAL(PAT$_NOACCESS);                                  ! REPORT ERROR
;  699     1664  2
;  700     1665  2 !++
;  701     1666  2 ! CHECK FOR DEMAND ZERO IMAGE SECTIONS.
;  702     1667  2 !--
;  703     1668  2 IF .CURISD[ISD$V_DZRO]
;  704     1669  2 THEN
;  705     1670  3         BEGIN
;  706     1671  3 !       PAT$GL_ERRCODE=$CRMPSC(   INADR=PAT$GL_ISVADDR          ! CREATE AND MAP IMAGE SECTION
;  707     1672  3 !                               , RETADR=ISEADR[ISE$L_MAPVST]   ! RETURNED MAP START AND END ADDRESSES
;  708     1673  3 !                               , FLAGS=(SEC$M_DZRO OR SEC$M_WRT OR SEC$M_CRF OR SEC$M_EXPREG) ! READ/WRITE
;  709     1674  3 !                               , CHAN=.PAT$GL_OLDFAB[FAB$L_STV] ! CHANNEL NUMBER
;  710     1675  3 !                               , PAGCNT=.CURISD[ISD$W_PAGCNT]); ! NUMBER OF PAGES TO MAP
;  711     1676  3         PAT$GL_ERRCODE=LIB$_CREMAPSEC(  PAT$GL_ISVADDR
;  712     1677  3                                       , ISEADR[ISE$L_MAPVST]
;  713     1678  4                                       , (SEC$M_DZRO OR SEC$M_WRT OR SEC$M_CRF OR SEC$M_EXPREG)
;  714     1679  3                                       , 0
;  715     1680  3                                       , 0
;  716     1681  3                                       , IMG_FIL_DESC
;  717     1682  3                                       , .CURISD[ISD$W_PAGCNT]
;  718     1683  3                                       , 0);
;  719     1684  3         IF NOT .PAT$GL_ERRCODE
;  720     1685  3         THEN
;  721     1686  3                 SIGNAL(PAT$_SYSERROR,0,.PAT$GL_ERRCODE);        ! REPORT ERROR
;  722     1687  3         END
;  723     1688  3
;  724     1689  3 !++
;  725     1690  3 ! IF NOT DEMAND ZERO, THEN CHECK FOR A GLOBAL SECTION THAT HAS NO LOCAL COPY.
;  726     1691  3 !--
;  727     1692  2 ELSE
;  728     1693  3         BEGIN
;  729     1694  3         IF .CURISD[ISD$V_GBL]                                   ! IF IT IS GLOBAL
;  730     1695  3         THEN
;  731     1696  4                 IF (.CURISD[ISD$L_VBN] EQL 0)                   ! AND DOES NOT HAVE A LOCAL COPY
;  732     1697  3                 THEN
;  733     1698  3                         SIGNAL(PAT$_GBLONLY,1,CURISD[ISD$T_GBLNAM]); ! THEN REPORT ERROR AND UNWIND/EXIT
;  734     1699  3 !       PAT$GL_ERRCODE=$CRMPSC(   INADR=PAT$GL_ISVADDR          ! CREATE AND MAP IMAGE SECTION
;  735     1700  3 !                               , RETADR=ISEADR[ISE$L_MAPVST]   ! RETURNED MAP STAR AND END ADDRESSES
;  736     1701  3 !                               , VBN=.CURISD[ISD$L_VBN]        ! BLOCK ADDRESS OF FIRST BLOCK TO MAP
;  737     1702  3 !                               , FLAGS=(SEC$M_WRT OR SEC$M_CRF OR SEC$M_EXPREG)        ! READ/WRITE
;  738     1703  3 !                               , CHAN=.PAT$GL_OLDFAB[FAB$L_STV] ! CHANNEL NUMBER
;  739     1704  3 !                               , PAGCNT=.CURISD[ISD$W_PAGCNT]); ! NUMBER OF PAGES TO MAP
;  740     1705  3
;  741     1706  3         PAT$GL_ERRCODE=LIB$_CREMAPSEC(  PAT$GL_ISVADDR,
;  742     1707  3                                         ISEADR[ISE$L_MAPVST],
;  743     1708  3                                         IF .PAT$GL_FLAGS [PAT$S_ABSOLUTE] AND NOT .PAT$GL_FLAGS [PAT$S_NEW_VERSION]
;  744     1709  4                                          THEN (SEC$M_WRT OR SEC$M_EXPREG)
```

```
;  745          1710  3                                  ELSE (SEC$M_WRT OR SEC$M_CRF OR SEC$M_EXPREG),
;  746          1711  3                                0,
;  747          1712  3                                0,
;  748          1713  3                                IMG_FIL_DESC,
;  749          1714  3                                .CURISD[ISD$W_PAGCNT],
;  750          1715  3                                .CURISD[ISD$L_VBN],
;  751          1716  3                                0,
;  752          1717  3                                IF .PAT$GL_FLAGS [PAT$S_ABSOLUTE] AND NOT .PAT$GL_FLAGS [PAT$S_NEW_VERSION]
;  753          1718  3                                 THEN  PAT$GL_CHANUM);
;  754          1719  3
;  755          1720  3
;  756          1721  3          IF NOT .PAT$GL_ERRCODE
;  757          1722  3          THEN
;  758          1723  3                  SIGNAL(PAT$_SYSERROR,0,.PAT$GL_ERRCODE);        ! REPORT ERROR
;  759          1724  3          RETURN                                                  ! RETURN ALWAYS SUCCESSFULLY
;  760          1725  2          END;
;  761          1726  2
;  762          1727  2 !++
;  763          1728  2 ! UPDATE THE LAST MAPPED ADDRESSES IN P0.
;  764          1729  2 !--
;  765          1730  2          PAT$GL_ISVADDR [START_OFF] = .ISEADR [ISE$L_MAPVST];
;  766          1731  2          PAT$GL_ISVADDR [END_OFF] = .ISEADR [ISE$L_MAPVEND];
;  767          1732  2
;  768          1733  1 END;                                                            ! END OF PATMAP
; INFO#212                 L1:1717
; Null expression appears in value-required context


                                    03FC 00000          .ENTRY  PAT$CREMAP, Save R2,R3,R4,R5,R6,R7,R8,R9    ; 1597
                59 00000000G   8F   D0 00002          MOVL    #PAT$_SYSERROR, R9                          ;
                58 00000000G   EF   9E 00009          MOVAB   LIB$_CREMAPSEC, R8                          ;
                57 00000000G   EF   9E 00010          MOVAB   PAT$GL_FLAGS, R7                            ;
                56 00000000G   00   9E 00017          MOVAB   LIB$SIGNAL, R6                              ;
                55 00000000G   EF   9E 0001E          MOVAB   PAT$GL_ISVADDR, R5                          ;
                54 00000000G   EF   9E 00025          MOVAB   PAT$GL_ERRCODE, R4                          ;
                5E             04   C2 0002C          SUBL2   #4, SP                                      ;
                53       04    AC   D0 0002F          MOVL    ISEADP  R3                                  ; 1651
                52       14    A3   9E 00033          MOVAB   20(R3), CURISD                              ;
                65       C8    8F   9A 00037          MOVZBL  #200, PAT$GL_ISVADDR                        ; 1652
          04    A5       C8    8F   9A 0003B          MOVZBL  #200, PAT$GL_ISVADDR+4                      ; 1653
                7E 00000000G   EF   9A 00040          MOVZBL  PAT$GL_OLDNBR+3, IMG_FIL_DESC               ; 1655
          04    AE 00000000G   EF   9E 00047          MOVAB   PAT$GB_OLDNAME, IMG_FIL_DESC+4              ; 1656
          FD    8F       0B    A2   91 0004F          CMPB    11(CURISD), #253                            ; 1661
                                    09   12 00054          BNEQ    1$                                      ;
                   006D80FA    8F   DD 00056          PUSHL   #7176442                                    ; 1663
                66             01   FB 0005C          CALLS   #1, LIB$SIGNAL                              ;
    2A    08    A2             02   E1 0005F 1$:      BBC     #2, 8(CURISD), 2$                           ; 1668
                                    7E   D4 00064          CLRL    -(SP)                                   ; 1677
                7E       02    A2   3C 00066          MOVZWL  2(CURISD), -(SP)                            ; 1682
                         08    AE   9F 0006A          PUSHAB  IMG_FIL_DESC                                ; 1677
                                    7E   7C 0006D          CLRQ    -(SP)                                   ;
                   0002000E    8F   DD 0006F          PUSHL   #131086                                     ; 1678
                         0C    A3   9F 00075          PUSHAB  12(R3)                                      ; 1677
                                    55   DD 00078          PUSHL   R5                                      ; 1676
```

```
                         68            08  FB 0007A          CALLS   #8, LIB$_CREMAPSEC                      ; 1677
                         64            50  D0 0007D          MOVL    R0, PAT$GL_ERRCODE                      ;
                         73            64  E8 00080          BLBS    PAT$GL_ERRCODE, 8$                      ; 1684
                                       64  DD 00083          PUSHL   PAT$GL_ERRCODE                          ; 1686
                                       7E  D4 00085          CLRL    -(SP)                                   ;
                                       59  DD 00087          PUSHL   R9                                      ;
                         66            03  FB 00089          CALLS   #3, LIB$SIGNAL                          ;
                                       68  11 0008C          BRB     8$                                      ; 1668
                         13        08  A2  E9 0008E 2$:      BLBC    8(CURISD), 3$                           ; 1694
                                   0C  A2  D5 00092          TSTL    12(CURISD)                              ; 1696
                                       0E  12 00095          BNEQ    3$                                      ;
                                   14  A2  9F 00097          PUSHAB  20(CURISD)                              ; 1698
                                       01  DD 0009A          PUSHL   #1                                      ;
                             006D8068      8F  DD 0009C      PUSHL   #7176296                                ;
                         66                03  FB 000A2      CALLS   #3, LIB$SIGNAL                          ;
                   0F    67                06  E1 000A5 3$:  BBC     #6, PAT$GL_FLAGS, 4$                    ; 1717
                                           67  95 000A9      TSTB    PAT$GL_FLAGS                            ;
                                           0B  19 000AB      BLSS    4$                                      ;
                         50 00000000G      EF  9E 000AD      MOVAB   PAT$GL_CHANUM, R0                       ;
                                           50  DD 000B4      PUSHL   R0                                      ;
                                           02  11 000B6      BRB     5$                                      ;
                                           7E  D4 000B8 4$:  CLRL    -(SP)                                   ;
                                           7E  D4 000BA 5$:  CLRL    -(SP)                                   ; 1707
                                   0C  A2  DD 000BC          PUSHL   12(CURISD)                              ; 1715
                         7E        02  A2  3C 000BF          MOVZWL  2(CURISD), -(SP)                        ; 1714
                                   10  AE  9F 000C3          PUSHAB  IMG_FIL_DESC                            ; 1707
                                       7E  7C 000C6          CLRQ    -(SP)                                   ;
                   0C    67                06  E1 000C8      BBC     #6, PAT$GL_FLAGS, 6$                    ; 1708
                                           67  95 000CC      TSTB    PAT$GL_FLAGS                            ;
                                           08  19 000CE      BLSS    6$                                      ;
                             00020008      8F  DD 000D0      PUSHL   #131080                                 ; 1709
                                           06  11 000D6      BRB     7$                                      ;
                             0002000A      8F  DD 000D8 6$:  PUSHL   #131082                                 ; 1710
                                   0C  A3  9F 000DE 7$:      PUSHAB  12(R3)                                  ; 1707
                                       55  DD 000E1          PUSHL   R5                                      ; 1706
                         68            0A  FB 000E3          CALLS   #10, LIB$_CREMAPSEC                     ; 1707
                         64            50  D0 000E6          MOVL    R0, PAT$GL_ERRCODE                      ;
                         0E            64  E8 000E9          BLBS    PAT$GL_ERRCODE, 9$                      ; 1721
                                       64  DD 000EC          PUSHL   PAT$GL_ERRCODE                          ; 1723
                                       7E  D4 000EE          CLRL    -(SP)                                   ;
                                       59  DD 000F0          PUSHL   R9                                      ;
                         66            03  FB 000F2          CALLS   #3, LIB$SIGNAL                          ;
                                           04  000F5        RET                                             ; 1693
                         65        0C  A3  7D 000F6 8$:      MOVQ    12(R3), PAT$GL_ISVADDR                  ; 1730
                                           04  000FA 9$:    RET                                             ; 1733

; Routine Size:  251 bytes,    Routine Base:  _PAT$CODE + 04F8
```

```
;  770          1734  1 END
;  771          1735  0 ELUDOM                                        ! END OF MODULE


                                                                .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  _PAT$OWN                      520  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  _PAT$PLIT                      76  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(0)
;  _PAT$CODE                    1523  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        63         0      1000          00:01.9



; Information:  1
; Warnings:     0
; Errors:       0


;                                     COMMAND QUALIFIERS

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

; Size:          1523 code + 596 data bytes
; Run Time:         00:35.6
; Elapsed Time:     01:40.3
; Lines/CPU Min:     2925
; Lexemes/CPU-Min: 33254
; Memory Used:  298 pages
; Compilation Complete
```

PATINT
LIS

PATINS
LIS

PATERR
LIS

PATIHD
LIS

PATEXA
LIS

PATFRE
LIS