```
PPPPPPPPPPPP        AAAAAAAAA     TTTTTTTTTTTTTTT     CCCCCCCCCCCC  HHH         HHH
PPPPPPPPPPPP        AAAAAAAAA     TTTTTTTTTTTTTTT     CCCCCCCCCCCC  HHH         HHH
PPPPPPPPPPPP        AAAAAAAAA     TTTTTTTTTTTTTTT     CCCCCCCCCCCC  HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPPPPPPPPPPP     AAA         AAA        TTT        CCC              HHHHHHHHHHHHHHH
PPPPPPPPPPPP     AAA         AAA        TTT        CCC              HHHHHHHHHHHHHHH
PPPPPPPPPPPP     AAA         AAA        TTT        CCC              HHHHHHHHHHHHHHH
PPP              AAAAAAAAAAAAAAA        TTT        CCC              HHH         HHH
PPP              AAAAAAAAAAAAAAA        TTT        CCC              HHH         HHH
PPP              AAAAAAAAAAAAAAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT           CCCCCCCCCCCC  HHH         HHH
PPP              AAA         AAA        TTT           CCCCCCCCCCCC  HHH         HHH
PPP              AAA         AAA        TTT           CCCCCCCCCCCC  HHH         HHH
```

FILEID**PATECO

PA
VO

```
PPPPPPPP      AAAAAA    TTTTTTTTTT  EEEEEEEEEE    CCCCCCCC    OOOOOO
PPPPPPPP      AAAAAA    TTTTTTTTTT  EEEEEEEEEE    CCCCCCCC    OOOOOO
PP      PP  AA      AA      TT      EE          CC          OO      OO
PP      PP  AA      AA      TT      EE          CC          OO      OO
PP      PP  AA      AA      TT      EE          CC          OO      OO
PP      PP  AA      AA      TT      EE          CC          OO      OO
PPPPPPPP    AA      AA      TT      EEEEEEEE    CC          OO      OO
PPPPPPPP    AA      AA      TT      EEEEEEEE    CC          OO      OO
PP          AAAAAAAAAA      TT      EE          CC          OO      OO
PP          AAAAAAAAAA      TT      EE          CC          OO      OO
PP          AA      AA      TT      EE          CC          OO      OO       ....
PP          AA      AA      TT      EE          CC          OO      OO       ....
PP          AA      AA      TT      EEEEEEEEEE    CCCCCCCC    OOOOOO         ....
PP          AA      AA      TT      EEEEEEEEEE    CCCCCCCC    OOOOOO         ....

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
```

```
L 0 MODULE PATECO (%IF %VARIANT EQL 1
  0                 %THEN
  0                         ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
  0                                          NONEXTERNAL = LONG_RELATIVE),
  0                 %FI
  0                 IDENT = 'V04-000'
  0                 ) =
  1 BEGIN
  1
  1 !
  1 !****************************************************************************
  1 !*                                                                          *
  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
  1 !*  ALL RIGHTS RESERVED.                                                    *
  1 !*                                                                          *
  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
  1 !*  TRANSFERRED.                                                            *
  1 !*                                                                          *
  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
  1 !*  CORPORATION.                                                            *
  1 !*                                                                          *
  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
  1 !*                                                                          *
  1 !*                                                                          *
  1 !****************************************************************************
  1
  1 !++
  1 ! FACILITY:     PATCH
  1 !
  1 ! ABSTRACT:     THIS MODULE CONTAINS ROUTINES TO HANDLE ECO COMMANDS.
  1 !
  1 ! ENVIRONMENT:  PART OF THE IMAGE FILE PATCH UTILITY FOR VAX.
  1 !
  1 ! AUTHOR:  K.D. MORSE   , CREATION DATE:        21-OCT-77
  1 !
  1 ! MODIFIED BY:
  1 !
  1 !       V02-006 PCG0001         Peter George            02-FEB-1981
  1 !               Add require statement for LIB$:PATDEF.REQ
  1 !
  1 !       V02-005 KDM0027         KATHLEEN D. MORSE       04-DEC-1980
  1 !               Change action taken on CHECK [NOT] ECO commands to be:
  1 !               skip this ECO and scan until you find the next SET ECO command.
  1 !
  1 !       , : VERSION
  1 !--
```

```
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1         PAT$ECO_CMDS : NOVALUE;                                  ! Handles ECO commands
1
1 !
1 ! INCLUDE FILES:
1 !
1 LIBRARY 'SYS$LIBRARY:LIB.L32';                                   ! System structure definitions
1 REQUIRE 'SRC$:PATPCT.REQ';                                       ! Defines PSECTS
1 REQUIRE 'SRC$:PATGEN.REQ';                                       ! Defines context bits
1 REQUIRE 'SRC$:PREFIX.REQ';                                       ! Defines structure macros
1 REQUIRE 'SRC$:PATPRE.REQ';                                       ! Defines PATCH structures
1 REQUIRE 'LIB$:PATDEF.REQ';                               ! Defines literals
1 REQUIRE 'LIB$:PATMSG.REQ';                                       ! Defines error message codes
1 REQUIRE 'SRC$:BSTRUC.REQ';                                       ! Defines basic structures
1 REQUIRE 'SRC$:LISTEL.REQ';                                       ! Defines list structures
1 REQUIRE 'SRC$:VXSMAC.REQ';                                       ! Defines TRUE and FALSE
1
1 !
1 ! MACROS:
1 !
1
1 !
1 ! EQUATED SYMBOLS:
1 !
1
1 !
1 ! OWN STORAGE:
1 !
1
1 !
1 ! EXTERNAL REFERENCES:
1 !
1 EXTERNAL
1         PAT$GL_FLAGS,                                            ! CLI flags
1         PAT$GL_ECO_UPD : BITVECTOR,                              ! /UPDATE qualifier ECO mask
1         PAT$GB_EXEC_CMD : BYTE,                                  ! Indicator whether or not to execute patch
1         PAT$GB_ECOLVL : BYTE,                                    ! ECO level for current patch
1         PAT$GL_IHPPTR : REF BLOCK[,BYTE],                        ! Pointer to patch section of image header
1         PAT$GL_IMGHDR : REF BLOCK[,BYTE],                        ! Pointer to image header
1         PAT$GL_CONTEXT : BITVECTOR,                              ! Context bits
1         PAT$GL_HEAD_LST,                                         ! Listhead for command parameters
1         PAT$GL_OLDNBK : BLOCK[,BYTE],                            ! Old image file name block
1         PAT$GB_OLDNAME;                                          ! Name of old image file
1
```

```
1 GLOBAL ROUTINE PAT$ECO_CMDS : NOVALUE =                              ! HANDLES ALL ECO COMMANDS
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! THIS ROUTINE HANDLES THE COMMANDS--CHECK ECO, CHECK NOT ECO, AND SET ECO.
1 ! THE ECO LEVEL NUMBERS HAVE ALREADY BEEN SCANNED AND INSERTED IN LIST ENTRIES.
1 ! ONE ENTRY CONSISTS OF THREE LONG WORDS:
1 !
1 !       1.) THE FORWARD LINK
1 !       2.) THE ECO LEVEL (OR THE START OF THE ECO RANGE)
1 !       3.) ZERO (OR THE END OF THE ECO RANGE)
1 !
1 ! NOTE THAT THE SECOND AND THIRD LONGWORDS DESCRIBE A SINGLE ECO LEVEL OR
1 ! A RANGE OF LEVELS, WITH THE SECOND LONGWORD BEING THE MINIMUM AND THE THIRD
1 ! BEING THE MAXIMUM.  IF THERE IS ONLY A SINGLE ECO LEVEL THEN THE THIRD
1 ! LONGWORD IS A ZERO.
1 !
1 ! THIS ROUTINE SETS UP THE POINTERS TO ACCESS THE ECO BITS IN THE IMAGE HEADER
1 ! AND THE ECO LEVELS FROM THE COMMAND.  THE CONTEXT BITS, SET_ECO AND
1 ! SET_NOT_ECO, SPECIFY WHETHER THE COMMAND WAS "SET ECO" OR "CHECK NOT ECO",
1 ! RESPECTIVELY.  IF NEITHER BIT IS SET, THEN THE COMMAND MUST HAVE BEEN "CHECK ECO".
1 !
1 ! IF THE APPROPRIATE ECO BITS ARE ALREADY SET, OR NOT SET, THEN AN ERROR
1 ! MESSAGE IS PRODUCED.  THE "SET ECO" COMMAND CHECKS THAT THE ECO LEVEL WAS
1 ! NOT ALREADY SET.  IF THE BIT IS NOT SET, THEN ANOTHER CHECK IS MADE TO SEE IF
1 ! A PREVIOUS "SET ECO" WAS SPECIFIED BEFORE AN "UPDATE" COMMAND.  THIS IS DONE
1 ! BY CHECKING PAT$GB_ECOLVL.  THIS INDICATOR IS SET BY THE "SET ECO" COMMAND
1 ! AND CLEARED BY THE "UPDATE" COMMAND.  IF IT CONTAINS A NON-ZERO VALUE WHEN
1 ! THIS ROUTINE EXECUTES, THEN TWO "SET ECO" COMMANDS WERE SPECIFIED FOR ONE
1 ! PATCH.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       THE CONTEXT BITS ARE SET AND THE COMMAND PARAMETERS ARE PARSED.
1 !       THE IMAGE HEADER HAS BEEN READ AND EXPANDED TO INCLUDE A PATCH SECTION,
1 !               IF NECESSARY.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       THE APPROPRIATE ECO BITS ARE SET OR CHECKED IN THE IMAGE HEADER.
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       AN ERROR MESSAGE AND EXIT OCCUR IF THE APPROPRIATE ECO BITS ARE NOT SET.
```

```
1 !
1 !--
1
2 BEGIN
2 LOCAL
2         LOOP_CNT,                                               ! LOOP COUNTER (MIN ECO LEVEL)
2         LOOP_MAX,                                               ! LOOP MAX (MAX ECO LEVEL)
2         LIST_PTR,                                               ! POINTER TO LIST ELEMENTS
2         ECO_PTR : REF [illegible]TVECTOR;                       ! POINTER TO HEADER ECO WORDS
2
2 !++
2 ! INITIALIZE POINTERS TO THE ECO BITS IN THE IMAGE HEADER AND TO THE FIRST
2 ! COMMAND PARAMETER, AN ECO LEVEL OR RANGE.
2 !--
2 ECO_PTR = CH$PTR(PAT$GL_IHPPTR[IHP$L_ECO1], 0);                 ! POINT TO FIRST ECO LONGWORD
2 LIST_PTR = CH$PTR(.PAT$GL_HEAD_LST, 0);                         ! POINT TO FIRST ECO LEVEL PARAMETER
2
2 !++
2 ! LOOP TO HANDLE ALL ECO LEVELS AND RANGES FOR THIS COMMAND.
2 !--
2 WHILE .LIST_PTR NEQA 0
2 DO
3         BEGIN
3         !++
3         ! SET UP A LOOP TO HANDLE A SINGLE ECO LEVEL OR RANGE OF ECO LEVELS.
3         !--
3         LOOP_CNT = .LIST_ELEM_EXP1(.LIST_PTR) - 1;              ! INITIALIZE LOOP COUNT
3         IF .[IST_ELEM_EXP2(.LIST_PTR) EQ[ 0                     !CHECK IF MAXIMUM RANGE
3         THEN                                                    ! EXISTS,
3                 LOOP_MAX = .LOOP_CNT                            ! IF NOT, SET MAX TO MIN AS SINGLE ECO LEVEL
3         ELSE
3                 LOOP_MAX = .LIST_ELEM_EXP2(.LIST_PTR) - 1;      ! IF SO, SET LOOP MAXIMUM (MAX ECO RANGE)
3
3         !++
3         ! CHECK FOR ERRORS IN RANGE OF ECO LEVELS.
3         !--
3         IF .LOOP_MAX LSS .LOOP_CNT                              ! CHECK FOR REVERSED RANGE
3         THEN
3                 SIGNAL(PAT$_EXARANGE);                          ! REPORT ERROR
4         IF .LOOP_MAX GTR (PAT$K_MAX_ECO - 1)                            ! CHECK FOR ILLEGAL ECO LEVEL
3         THEN                                                    ! IF > MAX, REPORT ERROR
3                 SIGNAL(PAT$_BADECO, 3, .LOOP_MAX+1, .PAT$GL_OLDNBK[NAM$B_RSL],
3                         PAT$GB_OLDNAME);
4         IF .LOOP_CNT LSS (PAT$K_MIN_ECO - 1)                            ! CHECK FOR ILLEGAL ECO LEVEL
3         THEN                                                    ! IF < MIN, REPORT ERROR
3                 SIGNAL(PAT$_BADECO, 3, .LOOP_CNT+1, .PAT$GL_OLDNBK[NAM$B_RSL],
3                         PAT$GB_OLDNAME);
3
3         WHILE .LOOP_CNT LEQ .LOOP_MAX                           ! LOOP FOR ONE RANGE
3         DO
4                 BEGIN
4
4                 !++
4                 ! NOW HANDLE THE "SET ECO" COMMANDS, BY TESTING FOR THE
4                 ! SET_ECO CONTEXT BIT.  THEN TEST IF ANOTHER ECO LEVEL IS SET.
4                 !--
4                 IF .PAT$GL_CONTEXT[SET_ECO]
```

```
THEN
        BEGIN
        IF .ECO_PTR[.LOOP_CNT]                  ! CHECK ECO BIT NOT ALREADY SET
        THEN
                BEGIN
                SIGNAL(PAT$_ECOSET, 3, .LOOP_CNT+1, .PAT$GL_OLDNBK[NAM$B_RSL],
                        PAT$GB_OLDNAME);        ! REPORT ERROR
                PAT$GB_ECOLVL = 0;
                PAT$GB_EXEC_CMD = FALSE;
                RETURN;
                END;
        IF .PAT$GB_ECOLVL NEQ 0
        THEN
                SIGNAL(PAT$_MULTECO, 2, .PAT$GB_ECOLVL, .LOOP_CNT+1);
        PAT$GB_ECOLVL = .LOOP_CNT+1;            ! REMEMBER ECO LEVEL FOR "UPDATE" COMMAND
        PAT$GB_EXEC_CMD = TRUE;                 ! FORCE COMMANDS TO BE EXECUTED
        IF (.PAT$GL_FLAGS AND PAT$M_UPDATE) NEQ 0
        THEN
                BEGIN
                IF NOT .PAT$GL_ECO_UPD<.LOOP_CNT, 1> ! IF NO /UPDATE
                THEN                            ! THEN RESET THE INDICATOR
                        BEGIN                   ! FOR NO EXECUTION
                        PAT$GB_ECOLVL = 0;
                        PAT$GB_EXEC_CMD = FALSE;
                        SIGNAL(PAT$_UPDATE, 1, .LOOP_CNT+1);
                        END;
                END;
        END
ELSE
        !++
        ! COMMAND WAS NOT "SET ECO", THEREFOR IT MUST BE EITHER
        ! "CHECK NOT ECO" OR "CHECK ECO".  THE SET_NOT_ECO CONTEXT
        ! BIT WILL TELL WHICH COMMAND IT WAS.
        !--
        IF .PAT$GL_CONTEXT[SET_NOT_ECO]         ! "CHECK NOT ECO" COMMAND?
        THEN                                    ! IF SET THEN YES
                BEGIN
                IF .ECO_PTR[.LOOP_CNT]          ! CHECK ECO LEVEL BIT NOT SET
                THEN                            ! IF IT IS, REPORT ERROR
                        BEGIN
                        SIGNAL(PAT$_ECOSET, 3, .LOOP_CNT+1,
                                .PAT$GL_OLDNBK[NAM$B_RSL], PAT$GB_OLDNAME);
                        PAT$GB_ECOLVL = C;
                        PAT$GB_EXEC_CMD = FALSE;
                        RETURN;
                        END;
                END
        ELSE
                IF NOT .ECO_PTR[.LOOP_CNT]      ! COMMAND WAS "CHECK ECO"
                THEN                            ! REPORT ERROR IF NOT SET
                        BEGIN
                        SIGNAL(PAT$_ECONOTSET, 3, .LOOP_CNT+1,
                                .PAT$GL_OLDNBK[NAM$B_RSL], PAT$GB_OLDNAME);
                        PAT$GB_ECOLVL = 0;
                        PAT$GB_EXEC_CMD = FALSE;
                        RETURN;
                        END;
```

```
;  275      1288 4              LOOP_CNT = .LOOP_CNT + 1;
;  276      1289 3              END;
;  277      1290 3          LIST_PTR = CH$PTR( .LIST_ELEM_FLINK(.LIST_PTR), 0);
;  278      1291 2          END;
;  279      1292 2 RETURN
;  280      1293 1 END;                                                    ! END OF PAT$ECO_CMDS


                                                         .TITLE  PATECO
                                                         .IDENT  \V04-000\

                                                ISE$C_SIZE==       20
                                                TXT$C_SIZE==       4
                                                PAL$C_SIZE==       16
                                                ASD$C_SIZE==       9
                                                FWR$C_SIZE==       24
                                                         .EXTRN  PAT$GL_FLAGS, PAT$GL_ECO_UPD
                                                         .EXTRN  PAT$GB_EXEC_CMD
                                                         .EXTRN  PAT$GB_ECOLVL, PAT$GL_IHPPTR
                                                         .EXTRN  PAT$GL_IMGHDR, PAT$GL_CONTEXT
                                                         .EXTRN  PAT$GL_HEAD_LST
                                                         .EXTRN  PAT$GL_OLDNBK, PAT$GB_OLDNAME
                                                         .WEAK   ACCESS_CHECK

                                                         .PSECT  _PAT$CODE,NOWRT,2

                                        07FC 00000       .ENTRY  PAT$ECO_CMDS, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1117
                                                                 R10                                           :
                  5A 00000000G  EF   9E 00002            MOVAB   PAT$GB_EXEC_CMD, R10                          :
                  59 00000000G  EF   9E 00009            MOVAB   PAT$GB_ECOLVL, R9                             :
                  58 00000000G  EF   9E 00010            MOVAB   PAT$GL_OLDNBK+3, R8                           :
                  57 00000000G  EF   9E 00017            MOVAB   PAT$GB_OLDNAME, R7                            :
                  56 00000000G  00   9E 0001E            MOVAB   LIB$SIGNAL, R6                                :
                  55 00000000G  EF   D0 00025            MOVL    PAT$GL_IHPPTR, ECO_PTR                        ; 1188
                  53 00000000G  EF   D0 0002C            MOVL    PAT$GL_HEAD_LST, LIST_PTR                     ; 1189
                                01   12 00033 1$:        BNEQ    2$                                            ; 1194
                                     04 00035            RET                                                   :
      52       04  A3               01   C3 00036 2$:        SUBL3   #1, 4(LIST_PTR), LOOP_CNT                     ; 1200
                          08  A3    D5 0003B            TSTL    8(LIST_PTR)                                   ; 1201
                                05   12 0003E            BNEQ    3$                                            :
                  54            52   D0 00040            MOVL    LOOP_CNT, LOOP_MAX                            ; 1203
                                05   11 00043            BRB     4$                                            :
      54       08  A3               01   C3 00045 3$:        SUBL3   #1, 8(LIST_PTR), LOOP_MAX                     ; 1205
                  52            54   D1 0004A 4$:        CMPL    LOOP_MAX, LOOP_CNT                            ; 1210
                                09   18 0004D            BGEQ    5$                                            :
                     006D80AA   8F   DD 0004F            PUSHL   #7176362                                      ; 1212
                  66            01   FB 00055            CALLS   #1, LIB$SIGNAL                                :
         0000007F  8F           54   D1 00058 5$:        CMPL    LOOP_MAX, #127                                ; 1213
                                13   15 0005F            BLEQ    6$                                            :
                                57   DD 00061            PUSHL   R7                                            ; 1215
                  7E            68   9A 00063            MOVZBL  PAT$GL_OLDNBK+3, -(SP)                        :
                          01    A4   9F 00066            PUSHAB  1(LOOP_MAX)                                   :
                                03   DD 00069            PUSHL   #3                                            :
                     006D809A   8F   DD 0006B            PUSHL   #7176346                                      :
                  66            05   FB 00071            CALLS   #5, LIB$SIGNAL                                :
                                52   D5 00074 6$:        TSTL    LOOP_CNT                                      ; 1217
                                13   18 00076            BGEQ    7$                                            :
                                      57  DD 00078          PUSHL   R7                                            ; 1219
                  7E                  68  9A 0007A          MOVZBL  PAT$GL_OLDNBK+3, -(SP)                        ;
                              01      A2  9F 0007D          PUSHAB  1(LOOP_CNT)                                   ;
                                      03  DD 00080          PUSHL   #3                                            ;
                        006D809A      8F  DD 00082          PUSHL   #7176346                                      ;
                  66                  05  FB 00088          CALLS   #5, LIB$SIGNAL                                ;
                  54                  52  D1 0008B 7$:      CMPL    LOOP_CNT, LOOP_MAX                            ; 1222
                                      03  15 0008E          BLEQ    8$                                            ;
                                    008D  31 00090          BRW     15$                                           ;
      46 00000000G EF                 02  E1 00093 8$:      BBC     #2, PAT$GL_CONTEXT+2, 10$                     ; 1230
      4E           65                 52  E0 0009B          BBS     LOOP_CNT, (ECO_PTR), 11$                      ; 1233
                   50                 69  9A 0009F          MOVZBL  PAT$GB_ECOLVL, R0                             ; 1242
                                      10  13 000A2          BEQL    9$                                            ;
                              01      A2  9F 000A4          PUSHAB  1(LOOP_CNT)                                   ; 1244
                                      50  DD 000A7          PUSHL   R0                                            ;
                                      02  DD 000A9          PUSHL   #2                                            ;
                        006D80F2      8F  DD 000AB          PUSHL   #7176434                                      ;
                  66                  04  FB 000B1          CALLS   #4, LIB$SIGNAL                                ;
                  50          01      A2  9E 000B4 9$:      MOVAB   1(R2), R0                                     ; 1245
                  69                  50  90 000B8          MOVB    R0, PAT$GB_ECOLVL                             ;
                  6A                  01  90 000BB          MOVB    #1, PAT$GB_EXEC_CMD                           ; 1246
      55 00000000G EF                 04  E1 000BE          BBC     #4, PAT$GL_FLAGS, 14$                         ; 1247
      4D 00000000G EF                 52  E0 000C6          BBS     LOOP_CNT, PAT$GL_ECO_UPD, 14$                 ; 1250
                                      69  94 000CE          CLRB    PAT$GB_ECOLVL                                 ; 1253
                                      6A  94 000D0          CLRB    PAT$GB_EXEC_CMD                               ; 1254
                                      50  DD 000D2          PUSHL   R0                                            ; 1255
                                      01  DD 000D4          PUSHL   #1                                            ;
                        006D803B      8F  DD 000D6          PUSHL   #7176251                                      ;
                  66                  03  FB 000DC          CALLS   #3, LIB$SIGNAL                                ;
                                      3A  11 000DF          BRB     14$                                           ; 1230
      16 00000000G EF                 01  E1 000E1 10$:     BBC     #1, PAT$GL_CONTEXT, 12$                       ; 1265
      2E           65                 52  E1 000E9          BBC     LOOP_CNT, (ECO_PTR), 14$                      ; 1268
                                      57  DD 000ED 11$:     PUSHL   R7                                            ; 1271
                  7E                  68  9A 000EF          MOVZBL  PAT$GL_OLDNBK+3, -(SP)                        ; 1272
                              01      A2  9F 000F2          PUSHAB  1(LOOP_CNT)                                   ; 1271
                                      03  DD 000F5          PUSHL   #3                                            ;
                        006D804B      8F  DD 000F7          PUSHL   #7176267                                      ;
                                      14  11 000FD          BRB     13$                                           ;
      18           65                 52  E0 000FF 12$:     BBS     LOOP_CNT, (ECO_PTR), 14$                      ; 1279
                                      57  DD 00103          PUSHL   R7                                            ; 1282
                  7E                  68  9A 00105          MOVZBL  PAT$GL_OLDNBK+3, -(SP)                        ; 1283
                              01      A2  9F 00108          PUSHAB  1(LOOP_CNT)                                   ; 1282
                                      03  DD 0010B          PUSHL   #3                                            ;
                        006D8043      8F  DD 0010D          PUSHL   #7176259                                      ;
                  66                  05  FB 00113 13$:     CALLS   #5, LIB$SIGNAL                                ;
                                      69  94 00116          CLRB    PAT$GB_ECOLVL                                 ; 1284
                                      6A  94 00118          CLRB    PAT$GB_EXEC_CMD                               ; 1285
                                          04 0011A          RET                                                   ; 1281
                                      52  D6 0011B 14$:     INCL    LOOP_CNT                                      ; 1288
                                    FF6B  31 0011D          BRW     7$                                            ; 1222
                  53                  63  D0 00120 15$:     MOVL    (LIST_PTR), LIST_PTR                          ; 1290
                                    FF0D  31 00123          BRW     1$                                            ; 1194
                                          04 00126          RET                                                   ; 1293

; Routine Size:  295 bytes,    Routine Base:  _PAT$CODE + 0000
;  282          1294  1 END                               !End of module
;  283          1295  0 ELUDOM




                                                                .EXTRN  LIB$SIGNAL

;                                 PSECT SUMMARY
;
;       Name                     Bytes                      Attributes
;
;   _PAT$CODE                        295  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         6         0       1000          00:01.8




;                                  COMMAND QUALIFIERS

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

; Size:          295 code + 0 data bytes
; Run Time:         00:19.0
; Elapsed Time:     00:54.9
; Lines/CPU Min:     4100
; Lexemes/CPU-Min: 47759
; Memory Used:  190 pages
; Compilation Complete
```

0300 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PATARI
LIS

PATCMD
LIS

PATECO
LIS

PATCON
LIS

PATENC
LIS

PATBAS
LIS

PATBLD
LIS