
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCC HHH HHH
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCC HHH HHH
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCC 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 AAAAAAAAAAAAAAA TTT CCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCC HHH HHH

FILEID**PATPRE

```
PPPPPPPP      AAAAAA    TTTTTTTTTT  PPPPPPPP    RRRRRRRR    EEEEEEEEEE
PPPPPPPP      AAAAAA    TTTTTTTTTT  PPPPPPPP    RRRRRRRR    EEEEEEEEEE
PP      PP   AA      AA     TT      PP      PP  RR      RR  EE
PP      PP   AA      AA     TT      PP      PP  RR      RR  EE
PP      PP   AA      AA     TT      PP      PP  RR      RR  EE
PP      PP   AA      AA     TT      PP      PP  RR      RR  EE
PPPPPPPP     AA      AA     TT      PPPPPPPP    RRRRRRRR    EEEEEEEE
PPPPPPPP     AA      AA     TT      PPPPPPPP    RRRRRRRR    EEEEEEEE
PP           AAAAAAAAAA     TT      PP          RR  RR      EE
PP           AAAAAAAAAA     TT      PP          RR  RR      EE
PP           AA      AA     TT      PP          RR    RR    EE                ....
PP           AA      AA     TT      PP          RR    RR    EE                ....
PP           AA      AA     TT      PP          RR      RR  EEEEEEEEEE        ....
PP           AA      AA     TT      PP          RR      RR  EEEEEEEEEE        ....
```

```
RRRRRRRR    EEEEEEEEEE    QQQQQQ
RRRRRRRR    EEEEEEEEEE    QQQQQQ
RR      RR  EE           QQ      QQ
RR      RR  EE           QQ      QQ
RR      RR  EE           QQ      QQ
RR      RR  EE           QQ      QQ
RRRRRRRR    EEEEEEEE     QQ      QQ
RRRRRRRR    EEEEEEEE     QQ      QQ
RR  RR      EE           QQ  QQ  QQ
RR  RR      EE           QQ  QQ  QQ
RR    RR    EE           QQ    QQ
RR    RR    EE           QQ    QQ
RR      RR  EEEEEEEEEE     QQQQ  QQ
RR      RR  EEEEEEEEEE     QQQQ  QQ
```

```
        P A T P R E . R E Q -- REQUIRE FILE FOR PATCH

 Version:      'V04-000'
```



```
++

MODULE: PATPRE.REQ

FACILITY: PATCH

ABSTRACT:
              REQUIRE FILE TO DEFINE LOCALLY SOME
              INTERESTING GENERAL SYMBOLS, STRUCTURES AND MACROS

HISTORY:


      AUTHOR: K.D. MORSE 4-OCT-77

Modified by:

      V03-001 MTR0025          Mike Rhodes                8-Aug-1983
              Add flags bit PAT$S_ABSOLUTE to support patching files in
              absolute mode (by virtual address).  NOTE that any file may
              be patched in absolute mode (eg. ISAM files etc.).
              Also, bit PAT$S_NEW_VERSION has been added which is used in
              conjunction with PAT$S_ABSOLUTE to determine the mapping
              performed and the generation of a new version of the patched
              file or if it is patched 'in place'.

      V0103   CNH0017          Chris Hume                 10-Oct-1979     12:00
```

```
!               Added OFP bit to PAT$GL_COMFAB.  Removed support for /COMMAND.
!               Added support for /VOLUME.  (PATMAI.B32 02.27, PATSTO.B32 01.17,
!               PATWRT.B32 02.06, [VMSLIB]QUADEF.MAR 01.20)
!
! MODIFICATIONS:
!
!  NO.  DATE            PROGRAMMER      PURPOSE
!  ---  ----            ----------      -------
!
!  01   7-MAR-78        K.D. MORSE      ADD PATCH AREA STRUCTURE, PAL.
!  02   27-APR-78       K.D. MORSE      ADD ASSEMBLER DIRECTIVE TABLE
!                                       STRUCTURE, ASD.
!  03   17-MAY-78       K.D. MORSE      ADD MESSAGE SEVERITY LEVELS, MSG$.
!  04   25-MAY-78       K.D. MORSE      ADD FORWARD REFERENCE TABLE STRUCTURE, FWR.
!
!--
```

```
!++
! FUNCTIONAL DESCRIPTION:
!
!       This require file defines some general, local values and macros
!       which cannot or are best not defined globally.
!
!--

LITERAL
        PAL$K_EXP_PAREA = 1,                    ! Expanding patch area flag
        PAL$K_ADD_PAREA = 0,                    ! Adding new patch area flag
        PAT$K_USER_DEF = 3,                     ! Code returned by PAT$BUILD_PATH if symbol was user-defined
        PAT$K_MAX_ECO = 128,                    ! Maximum eco level allowed
        PAT$K_MIN_ECO = 1,                      ! Minimum eco level allowed
        PAT$K_LENPRIV = 20,                     ! Length of process private ISD
        PAT$S_COMMAND = 0,                      ! Command file bit for CLI to set
        PAT$S_JOURNAL = 1,                      ! Journal file bit for CLI to set
        PAT$S_INPUT = 2,                        ! Input image file bit for CLI to set
        PAT$S_OUTPUT = 3,                       ! Output image file bit for CLI to set
        PAT$S_UPDATE = 4,                       ! Update qualifier bit for CLI to set
        PAT$S_VOLUME = 5,                       ! Volume qualifier bit for CLI to set
        PAT$S_ABSOLUTE = 6,                     ! Absolute qualifier bit for CLI
        PAT$S_NEW_VERSION = 7,                  ! New_Version qualifier bit for CLI
        PAT$M_COMMAND = 1,                      ! Mask of command file CLI bit
        PAT$M_JOURNAL = 2,                      ! Mask of journal file CLI bit
        PAT$M_INPUT = 4,                        ! Mask of input image file CLI bit
        PAT$M_OUTPUT = 8,                       ! Mask of output image file CLI bit
        PAT$M_UPDATE = 16,                      ! Mask of update qualifier CLI bit
        PAT$M_VOLUME = 32,                      ! Mask of volume qualifier CLI bit
        PAT$M_ABSOLUTE = 64,                    ! Mask of absolute qualifier CLI bit
        PAT$M_NEW_VERSION = 128;                ! Mask of new_version qualifier CLI bit

!++
! Definition of image section table entries
!--

BYTEBLOCKFIELDS(ISE,
                L_NXTISE,4,                     ! Link to next image section entry
                L_IMGVST,4,                     ! Start virtual address in image section
                L_IMGVEND,4,                    ! Ending virtual address in image section
                L_MAPVST,4,                     ! Starting virtual address of mapped image section
                L_MAPVEND,4);                   ! Ending virtual address of mapped image section

!++
! Definition of PATCH command text block.
!--

BYTEBLOCKFIELDS(TXT,
                L_NXTBLK,4);                    ! Pointer to next block

!++
! Definition of patch area list entry, PAL.
!--
BYTEBLOCKFIELDS(PAL,                            ! Prefix name
                L_FLINK,4,                      ! Forward link
```

```
                L_START_ADR,4,                                  ! Starting patch area address
                L_END_ADR,4,                                    ! Ending patch area address
                L_CS_NAME,4);                                   ! ASCIC name for patch area

!++
! Definition of Assembler Directive table structure, ASD.
!--
BYTEBLOCKFIELDS(ASD,
                L_PC,4,                                         ! Unmapped PC of assembler directive
                L_OPINFO,4,                                     ! Address of instruction opcode table entry for directive
                B_NUM_OPRND,1);                                 ! Number of operands on directive

!++
! Generate the data structure for the ForWard Reference table, FWR$.
!--
BYTEBLOCKFIELDS(FWR,
                L_FLINK,4,                                      ! Forward link to next entry
                L_PC,4,                                         ! PC at which operand is to be placed
                W_OPRNDLNG,2,                                   ! Length of operand string
                B_NUMBYTES,1,                                   ! Number of bytes operand will take in encoded form
                B_NTHOPRND,1,                                   ! Number of operand in instruction, i.e., nth operand
                A_OPRNDADR,4,                                   ! Address of unreduced operand ascii string
                A_OPINFO,4,                                     ! Index into OPINFO table for instruction's opcode
                B_BUFOFF,4);                                    ! Offset into PAT$GL_TEMP_BUF to hold encoded operand

!++
! Define PATCH message severity levels.
!--
LITERAL
        MSG$K_INFO = 3,
        MSG$K_WARN = 0,
        MSG$K_SEVERE = 2,
        MSG$K_FATAL = 4,
        MSG$K_SUCCESS = 1;
```