


```

PPPPPPPP      AAAAAA      TTTTTTTTTT      PPPPPPPP      RRRRRRRR      EEEEEEEEEEE
PPPPPPPP      AAAAAA      TTTTTTTTTT      PPPPPPPP      RRRRRRRR      EEEEEEEEEEE
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      TTT      RRRRRRRR      EEEEEEEEEEE
PPPPPPPP      AA      AA      TTT      RRRRRRRR      EEEEEEEEEEE
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      EE
PP      AA      AA      TT      PP      RR      RR      EE

```

```

RRRRRRRR      EEEEEEEEEEE      QQQQQQ
RRRRRRRR      EEEEEEEEEEE      QQQQQQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RRRRRRRR      EEEEEEEEEEE      QQ      QQ
RRRRRRRR      EEEEEEEEEEE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EEEEEEEEEEE      QQQQ      QQ
RR      RR      EEEEEEEEEEE      QQQQ      QQ

```

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

Version: 'V04-000'

```

*****
*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

```

```

**
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$$_ABSOLUTE to support patching files in
          absolute mode (by virtual address). NOTE that any file may
          be patched in absolute mode (eg. ISAM files etc.).
          Also, bit PAT$$_NEW_VERSION has been added which is used in
          conjunction with PAT$$_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 OFF 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, MSGS.
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

```

```

PALSX_EXP_PAREA = 1,      ! Expanding patch area flag
PALSX_ADD_PAREA = 0,      ! Adding new patch area flag
PATSK_USER_DEF = 3,       ! Code returned by PATSBUILD_PATH if symbol was user-defined
PATSK_MAX_ECO = 128,      ! Maximum eco level allowed
PATSK_MIN_ECO = 1,        ! Minimum eco level allowed
PATSK_LENPRIV = 20,       ! Length of process private ISD
PATSS_COMMAND = 0,        ! Command file bit for CLI to set
PATSS_JOURNAL = 1,        ! Journal file bit for CLI to set
PATSS_INPUT = 2,          ! Input image file bit for CLI to set
PATSS_OUTPUT = 3,         ! Output image file bit for CLI to set
PATSS_UPDATE = 4,         ! Update qualifier bit for CLI to set
PATSS_VOLUME = 5,         ! Volume qualifier bit for CLI to set
PATSS_ABSOLUTE = 6,       ! Absolute qualifier bit for CLI
PATSS_NEW_VERSION = 7,    ! New Version qualifier bit for CLI
PATSM_COMMAND = 1,        ! Mask of command file CLI bit
PATSM_JOURNAL = 2,        ! Mask of journal file CLI bit
PATSM_INPUT = 4,          ! Mask of input image file CLI bit
PATSM_OUTPUT = 8,         ! Mask of output image file CLI bit
PATSM_UPDATE = 16,        ! Mask of update qualifier CLI bit
PATSM_VOLUME = 32,        ! Mask of volume qualifier CLI bit
PATSM_ABSOLUTE = 64,      ! Mask of absolute qualifier CLI bit
PATSM_NEW_VERSION = 128;  ! Mask of new_version qualifier CLI bit

```

```

++

```

```

Definition of image section table entries

```

```

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

```

```

L_START_ADR,4,      | Starting patch area address
L_END_ADR,4,        | Ending patch area address
L_CS_NAME,4);      | ASCII 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;

```

