PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP		AAAAAAA AAAAAAA A AAAAA		\$	RRRRRRRRRRR RRRRRRRRRRR RRRRRRRRRRRR		LLL LLL LLL
PPP	PPP	AAA	AAA	SSS	RRR RRR	ŤŤŤ	iii
PPP	PPP	AAA	AAA	ŠŠŠ	RRR RRR	ŤŤŤ	ΙΙΙ
PPP	PPP	AAA	AAA	SSS	RRR RRR	ŤŤŤ	ΙΙΙ
PPP	PPP	AAA	AAA	ŠŠŠ	RRR RRR	ŤŤ	<u>ו</u> וו
PPP	PPP	AAA	AAA	SSS	RRR RRR	ŤŤŤ	<u>ו</u> וו
PPP	PPP	AAA	AAA	ŠŠŠ	RRR RRR	ŤŤŤ	iii
PPPPPPPP	PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP		AAA	SSSSSSSS	RRRRRRRRRRR	ŤŤŤ	iii
PPPPPPPP	PPP	AAA	AAA	\$\$\$\$\$\$\$\$\$	RRRRRRRRRRR	TTT	ίίί
PPPPPPPP	PPPP	AAA	AAA	ŠSSSSSSS	RRRRRRRRRRR	TTT	ίίί
PPP		AAAAAAAA	AAAAA	SSS	RRR RRR	TTT	LLL
PPP		AAAAAAAA	AAAAA	SSS	RRR RRR	TTT	ĹĹĹ
PPP		AAAAAAAA	AAAAA	SSS	RRR RRR	TTT	LLL
PPP		AAA	AAA	SSS	RRR RRR	TTT	LLL
PPP		AAA	AAA	SSS	RRR RRR	TTT	LLL
PPP		AAA	AAA	SSS	RRR RRR	TTT	LLL
PPP		AAA	AAA	SSSSSSSSSS	RRR RRR	TTT	LLLLLLLLLLLLLL
PPP		AAA	AAA	SSSSSSSSSS	RRR RRR	TTT	LLLLLLLLLLLLLL
PPP		AAA	AAA	\$\$\$\$\$\$\$\$\$\$\$\$\$\$	RRR RRR	TTT	

Sym

_\$2

PAS

PA

PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP	AAAAA AA AA AA AA	\$	FFFFFFFF FF FFFFFFFFFFFFFFFFFFFFFFFFFF	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	••••
\$	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD					

{ Pascal File Control Block (FCB\$) field definitions
{ Fice: PASFCB.SDL, Edit: SBL1006 MODULE SECROPE IDENT "1-006";

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.

Author: Steven B. Lionel, 1-April-1981

(1-001 - Original. SBL 1-April-1981 (1-002 - Replace "." with ":" for SDL V1.5. SBL 16-July-1982 (1-003 - Add FCB\$V_RND. SBL 29-July-1982

€ 1-004 - Replace FCB\$V_RND with FCB\$V_FOD. SBL 29-July-1982

(1-005 - Add FCB\$V_INITIATE_PROMPT. SBL 15-Dec-1982

(1-006 - Use MARKER. SBL 10-Feb-1983

{ The Pascal File Control Block (FCB\$) is dynamically allocated by the { Run-Time Library and contains internal information necessary to the { RTL to control file processing. The allocated block actually contains { the FCB followed by the RMS RAB, FAB and NAM blocks. The pointer { to this storage is identified as FCB inside the RTL and points to the RAB. The FCB is at negative offsets from this position.

The layout is as follows:

FCB RAB FAB NAM

<-- PFV\$A_FCB (in Pascal File Variable)</pre>

{-

```
AGGREGATE FCB_STRUCT STRUCTURE PREFIX FCB$ MARKER FCB$ ORIGIN FCB:
       QUEUE_FLINK LUNGWORD UNSIGNED:
                                                            { Forward link to queue of all open files { Backward link to queue of all open files
      QUEUE_BLINK LONGWORD UNSIGNED;
SPARE_1 LONGWORD FILL;
RECORD_NUMBER_LONGWORD;
                                                                  Current record number for textfiles
      KEY TYPES ADDRESS;
NKEYS LONGWORD UNSIGNED;
                                                                  Vector of key types
                                                                  Number of keys defined
      STATUS LONGWORD:
                                                                  Status of last operation for STATUS()
       COMPONENT LONGWORD UNSIGNED;
                                                                  Component number for direct access
      PFV ADDRESS;
LINELIMIT LONGWORD UNSIGNED;
                                                                  Address of PFV
                                                                  Linelimit of this file
      PFD ADDRESS;
                                                                   Address of PFD
      RECORD_BEG ADDRESS;
RECORD_CUR ADDRESS;
RECORD_END ADDRESS;
RECORD_LEN LONGWORD UNSIGNED;
ATTRIB_OVERLAP_UNION;
ATTRIB_BITS_STRUCTURE;
TEXT_BITFIELD;
RINARY_RITFIELD;
                                                                  Beginning of record buffer
                                                                  Current position in record buffer
                                                                  Last+1 of record buffer
                                                               { Length of record buffer
                                                               { Attributes of this file. This word is to be
                                                                  the same layout as PFD$W_ATTRIB.
                                                                  The file is of type TEXT.
                                                                 The file is of type IEAT.

The file is of a non-varying binary type. The file is of a varying string type. The allocated buffer is two bytes bigger than the maximum record size (PFD$L_LENGTH).

The file variable is declared WRITEONLY.
                    BINARY BITFIELD: 'VARYING' BITFIELD;
                    NOREAD BITFIELD:
                    NOWRITE BITFIELD:
                                                                  The file variable is declared READONLY
                    EXTERN BITFIELD:
                                                                  The file is external; i.e. it has been declared
                                                                  in the PROGRAM or MODULE header.
                                                               { The file is statically allocated. { The file is heap allocated via NEW.
                    STATIC BITFIELD;
     STATIC BITFIELD; { The file is statically allocated. HEAP BITFIELD; { The file is heap allocated via NEW. $$fill_1 BITFIELD LENGTH 8 FILL; END ATTRIB_BITS; END ATTRIB_OVERLAP; PROMPT_CC GORD UNSIGNED; { Carriage control bytes for prompting CONSTANT CC_NULL EQUALS $$X0000 PREFIX FCB TAG $$K; { No control CONSTANT CC_LFNL EQUALS $$X0001 PREFIX FCB TAG $$K; { If before, no CONSTANT CC_NLCR EQUALS $$X8D00 PREFIX FCB TAG $$K; { Nothing befor CONSTANT CC_LFCR EQUALS $$X8D01 PREFIX FCB TAG $$K; { Lf before, CR OPTIONS1 OVERLAP UNION; OPTIONS1 LONGWORD UNSIGNED; OPTIONS1 BITS STRUCTURE:
                                                                                                         { LF before, nothing after
                                                                                                         ( Nothing before, CR after
                                                                                                        { Lf before, CR after
             OPTIONS1 BITS STRUCTURE;
SEQUENTIAL BITFIELD;
DIRECT BITFIELD;
                                                                  Sequential access specified
                                                                  Direct access specified
                    KEYED BITFIELD;
                                                                  Keyed access specified
                    READ ONLY BITFIELD;
OLD FILE BITFIELD;
SAVE BITFIELD;
                                                                  file was opened READONLY
                                                                  file already exists
file is to be saved after close
                    DELETE BITFIELD:
                                                                  File is to be deleted after close
                    PRINT BITFIELD;
SUBMIT BITFIELD;
                                                                  file is to be printed after close
                                                                  File is to be submitted as a batch job after close
                    USER_ACTION BITFIELD;
LAZY BITFIELD;
INSPECTION BITFIELD;
GENERATION BITFIELD;
                                                                  A USER_ACTION routine was specified
                                                                 Lazy-lookahead is in progress File is in INSPECTION mode
                                                               { file is in GENERATION mode
```

E

E

```
EOF BITFIELD; { file is at EOF PROMPT ENABLE BITFIELD; { Prompting is enabled PARTIAL LINE BITFIELD; { LCCATE was last operation DEALLOC BITFIELD; { LCCATE was last operation DEALLOC BITFIELD; { Coallocation of this FCB requested DYNAMIC RSN BITFIELD; { RSN in NAM block is dynamic DYNAMIC TUBF BITFIELD; { File is a string (READV/WRITEV) ON QUEUE BITFIELD; { FCB is on queue of files INDEXED BITFIELD; { File is indexed organization PROMPT ALWAYS BITFIELD; { Always write partial records FOD BITFIELD; { Lookahead on this file starts prompt $$fill 2 BITFIELD LENGTH 6 FILL; END OPTIONS1 BITS; END OPTIONS1 BITS;
                                                                                                                                                                                                                                  { Length of FCB
```

CONSTANT BLN EQUALS :;

FCB BYTE TAG R:

{ Field to reference entire structure

END FCB_STRUCT; END_MODULE \$FCBDEF;

EX

0293 AH-BT13A-SE

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

