


```

PPPPPPPP      AAAAAA      SSSSSSSS      FFFFFFFFFF      CCCCCCCC      BBBB88888
PPPPPPPP      AAAAAA      SSSSSSSS      FFFFFFFFFF      CCCCCCCC      BBBB88888
PP      PP      AA      AA      SS      FF      CC      BB      BB
PP      PP      AA      AA      SS      FF      CC      BB      BB
PP      PP      AA      AA      SS      FF      CC      BB      BB
PP      PP      AA      AA      SS      FF      CC      BB      BB
PPPPPPPP      AA      AA      SSSSSS      FFFFFFFF      CC      BBBB88888
PPPPPPPP      AA      AA      SSSSSS      FFFFFFFF      CC      BBBB88888
PP      AAAAAAAAAA      SS      FF      CC      BB      BB
PP      AAAAAAAAAA      SS      FF      CC      BB      BB
PP      AA      AA      SS      FF      CC      BB      BB
PP      AA      AA      SS      FF      CC      BB      BB
PP      AA      AA      SSSSSSSS      FF      CCCCCCCC      BBBB88888
PP      AA      AA      SSSSSSSS      FF      CCCCCCCC      BBBB88888

```

```

SSSSSSSS      DDDDDDDD      LL
SSSSSSSS      DDDDDDDD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SSSSSS      DD      DD      LL
SSSSSS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SSSSSSSS      DDDDDDDD      LLLLLLLLLL
SSSSSSSS      DDDDDDDD      LLLLLLLLLL

```



```

AGGREGATE FCB STRUCT STRUCTURE PREFIX FCB$ MARKER FCB$ ORIGIN FCB;
  QUEUE_FLINK LONGWORD UNSIGNED; { Forward link to queue of all open files
  QUEUE_BLINK LONGWORD UNSIGNED; { Backward link to queue of all open files
  SPARE_1 LONGWORD FILL;
  RECORD_NUMBER LONGWORD; { Current record number for textfiles
  KEY_TYPES ADDRESS; { Vector of key types
  NKEYS LONGWORD UNSIGNED; { Number of keys defined
  STATUS LONGWORD; { Status of last operation for STATUS()
  COMPONENT LONGWORD UNSIGNED; { Component number for direct access
  PFV ADDRESS; { Address of PFV
  LINELIMIT LONGWORD UNSIGNED; { Linelimit of this file
  PFD ADDRESS; { Address of PFD
  RECORD_BEG ADDRESS; { Beginning of record buffer
  RECORD_CUR ADDRESS; { Current position in record buffer
  RECORD_END ADDRESS; { Last+1 of record buffer
  RECORD_LEN LONGWORD UNSIGNED; { Length of record buffer
  ATTRIB_OVERLAP UNION;
    ATTRIB_WORD UNSIGNED; { Attributes of this file. This word is to be
    ATTRIB_BITS STRUCTURE; { the same layout as PFDSW_ATTRIB.
      TEXT BITFIELD; { The file is of type TEXT.
      BINARY BITFIELD; { The file is of a non-varying binary type.
      "VARYING" BITFIELD; { The file is of a varying string type. The
      { allocated buffer is two bytes bigger than the
      { maximum record size (PFDSL_LENGTH).
      NOREAD BITFIELD; { The file variable is declared WRITEONLY.
      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.
      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 WORD 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; { LF before, nothing after
  CONSTANT CC_NLCR EQUALS $X8D00 PREFIX FCB TAG $K; { Nothing before, CR after
  CONSTANT CC_LFCR EQUALS $X8D01 PREFIX FCB TAG $K; { LF before, CR after
  OPTIONS1 OVERLAP UNION;
    OPTIONS1 LONGWORD UNSIGNED;
    OPTIONS1 BITS STRUCTURE;
      SEQUENTIAL BITFIELD; { Sequential access specified
      DIRECT BITFIELD; { Direct access specified
      KEYED BITFIELD; { Keyed access specified
      READ_ONLY BITFIELD; { File was opened READONLY
      OLD_FILE BITFIELD; { File already exists
      SAVE BITFIELD; { File is to be saved after close
      DELETE BITFIELD; { File is to be deleted after close
      PRINT BITFIELD; { File is to be printed after close
      SUBMIT BITFIELD; { File is to be submitted as a batch job after close
      USER_ACTION BITFIELD; { A USER_ACTION routine was specified
      LAZY BITFIELD; { Lazy-lookahead is in progress
      INSPECTION BITFIELD; { File is in INSPECTION mode
      GENERATION BITFIELD; { File is in GENERATION mode

```

```
EOF BITFIELD;          ( File is at EOF
PROMPT_ENABLE BITFIELD; ( Prompting is enabled
PARTIAL_LINE BITFIELD; ( Partial line has been written
LOCATE BITFIELD;      ( LOCATE was last operation
DEALLOC BITFIELD;    ( Deallocation of this FCB requested
DYNAMIC_RSN BITFIELD; ( RSN in NAM block is dynamic
DYNAMIC_UBF BITFIELD; ( UBF in RAB block is dynamic
STRING 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;        ( Device is file oriented (DEV$V_FOD)
INITIATE_PROMPT BITFIELD; ( Lookahead on this file starts prompt
$$fill 2 BITFIELD LENGTH 6 FILL;
END OPTIONS1 BITS;
END OPTIONS1_OVERLAP;

CONSTANT BLN_EQUALS ;;          ( Length of FCB
FCB_BYTE_TAG R;                ( Field to reference entire structure

END FCB_STRUCT;
END_MODULE $FCBDEF;
```

