```
/*
** @(#)pdfl.h   5.1 06/22/94
**
** data structures usefull for tucson pdfl files
**
*/

#ifndef PDFL_H

typedef struct  {
        short   ndir;           /* No. of blocks in directory table */
        short   maxent;         /* Max no. of entries in dir table  */
        short   bytperent;      /* bytes per dir table entry */
        short   nextent;        /* next available dir table entry */
        short   lastblk;        /* block of next available entry */
        short   nextblk;        /* next block location in file for data */

                                /* these 6 are not used by unipops */
        short   nxtb;           /* last dir table block currently used */
        short   nbcal;          /* first block where cal data are stored */
        short   nbsv;           /* first block where save scans stored */
        short   nbwrk;          /* first block of integration work area */
        short   nbscr;          /* first block of link task scratch area */
        short   pftype;         /* POPS file type */

        char    obsname[16];    /* observer name */
        char    pid[8];         /* Project ID */

        char    binfmt_id[16];  /* Binary number format ID */

        short   ifiller[224];   /* fills it out to 512 bytes */
        }       pdfl_bs_block;  /* bootstrap block */

/*      WARNING:  This gets padded out to 64 bytes !!! */

typedef struct {
        long    scan_number;
        char    mode[4];
        long    blk1;           /* block location of first "feed" */
        long    nblk1;          /* number of blocks for first "feed" */
        long    blk2;           /* block location of second "feed" */
        long    nblk2;          /* number of block for second "feed" */
        char    source_name[16];
        double  velocity, rest_freq;
        short   subscan;
} pdfl_index_entry;

typedef struct {
        long    scan_number;
        char    mode[4];
        long    blk1;           /* block location of first "feed" */
        long    nblk1;          /* number of blocks for first "feed" */
        long    blk2;           /* block location of second "feed" */
        long    nblk2;          /* number of block for second "feed" */
        char    source_name[16];
        double  velocity, rest_freq;
} old_pdfl_index_entry;

typedef struct {
        long    scan_number;
        }       gzfl_index_entry;   /* old, pre Sept 91 gzfl index format */

typedef struct {
        long    scan_number;
```

```
        char    type[2];        /* ED, edit scans, KP, keep scans */
        char    mode[2];        /* LI, line, CO, continuum */
        short   blk;
        short   nblk;
        }       pkfl_index_entry;

#define PDBUFSIZE sizeof(pdfl_bs_block)
#define MAXENT 2048
/*              see warning above for reason why sizeof won't work here */
#define BYTPERENT 58
#define OLD_BYTPERENT sizeof(old_pdfl_index_entry)
#define NDIR ((MAXENT * BYTPERENT) / PDBUFSIZE)
#define OLD_NDIR ((MAXENT * OLD_BYTPERENT) / PDBUFSIZE)
#define INDEX_BLOCK(ENTRY) (((ENTRY)*(BYTPERENT))/(PDBUFSIZE))
#define BLOCK_OFFSET(ENTRY)(((ENTRY)*(BYTPERENT)) % (PDBUFSIZE))
#define ISNEW(A) ((A).bs.bytperent == BYTPERENT)

typedef struct {
        int fd;         /* file descriptor */
        char name[256]; /* name of file */
        pdfl_bs_block bs;       /* boot strap block */
        pdfl_index_entry index[MAXENT]; /* index, gains if GZFL file */
        old_pdfl_index_entry old_index[MAXENT]; /* used only for old type */
        } pdfl_file;

#define PDFL_H
#endif
```

```
/*
 *  Cactus file %W%
 *        Date %G%
 *
 */

#ifndef HEADER_H

/* sdd.h -- Defined HEADER structures for POPS I/O */

#define getheads(a) swaps(a)

struct HEADER {
  short headcls;
  short oneptr;
  short twoptr;
  short thrptr;
  short fourptr;
  short fiveptr;
  short sixptr;
  short sevptr;
  short eigptr;
  short nineptr;
  short tenptr;
  short elvptr;
  short twlptr;
  short trnptr;
  short align1; /* makes class 1 align */
  short align2; /* makes class 1 align */

/* Class 1 */

  double headlen;
  double datalen;
  double scan;
  char obsid[8];
  char observer[16];
  char telescop[8];
  char projid[8];
  char object[16];
  char obsmode[8];
  char frontend[8];
  char backend[8];
  char precis[8];

/* Class 2 */

  double xpoint;
  double ypoint;
  double uxpnt;
  double uypnt;
  double ptcon[4];
  double orient;
  double focusr;
  double focusv;
  double focush;
  char pt_model[8];

/* Class 3 */

  double utdate;
  double ut;
  double lst;
  double norchan;
```

```
  double noswvar;
  double nophase;
  double cyclLen;
  double samprat;
  char cl11type[8];

/* Class 4 */

  double epoch;
  double xsource;
  double ysource;
  double xref;
  double yref;
  double epocra;
  double epocdec;
  double gallong;
  double gallat;
  double az;
  double el;
  double indx;
  double indy;
  double desorg[3];
  char coordcd[8];

/* Class 5 */

  double tamb;
  double pressure;
  double humidity;
  double refrac;
  double dewpt;
  double mmh2o;

/* Class 6 */

  double scanang;
  double xzero;
  double yzero;
  double deltaxr;
  double deltayr;
  double nopts;
  double noxpts;
  double noypts;
  double xcell0;
  double ycell0;
  char frame[8];

/* Class 7 */

  double bfwhm;
  double offscan;
  double badchv;
  double rvsys;
  double velocity;
  char veldef[8];
  char typecal[8];

/* Class 8 */

  double appeff;
  double beameff;
  double antgain;
  double etal;
  double etafss;
```

```
/* Class 9 - Mt. Graham */

  double synfreq;
  double lofact;
  double harmonic;
  double loif;
  double firstif;
  double razoff;
  double reloff;
  double bmthrow;
  double bmorent;
  double baseoff;
  double obstol;
  double sideband;
  double wl;
  double gains;
  double pbeam[2];
  double mbeam[2];
  double sroff[4];
  double foffsig;
  double foffref1;
  double foffref2;

 /* Class 10 */

  double openpar[10];

 /* Class 11 */

/* removed 288 bytes of unused variables, or 36 doubles */

  double current_disk;
  double bologain;
  double sptip_start;
  double sptip_stop;
  double ramp_up;
  double tatms;
  double taus;
  double taui;
  double tatmi;
  double tchop;
  double tcold;
  double gaini;
  double count[3];
  char   linename[16];
  double refpt_vel;
  double tip_humid;
  double tip_ref_flag;
  double refract_45;
  double ref_correct;
  double beam_num;                          /* which beam of array rcvr this is */
  double burn_time;       /* the time at which the scan was acutally recorded */
  double parallactic;                             /* angle between north and up */
  double az_offset;                         /* az offset during spec five point */
  double el_offset;                         /* el offset during spec five point */
  double nutate_rate;                                          /* nutate freq */

/* if you add a variable above remove the appropriate number of doubles below */

  double spares04[2];
  double spares05[6];
```

```
 /* Class 12 */

  double obsfreq;
  double restfreq;
  double freqres;
  double bw;
  double trx;
  double tcal;
  double stsys;
  double rtsys;
  double tsource;
  double trms;
  double refpt;
  double x0;
  double deltax;
  double inttime;
  double noint;
  double spn;
  double tauh2o;
  double th2o;
  double tauo2;
  double to2;
  char polariz[8];
  double effint;
  char rx_info[16];

/* Class 13 */

  double nostac;
  double fscan;
  double lscan;
  double lamp;
  double lwid;
  double ili;
  double rms;
  double align3[4];
};

#define HEADER_H

#endif
```