Technical documentation

libfmcol User Manual

28. February 2008

Norwegian
Meteorological Institute
met.no

Revision history

Version	**Date**	**Comment**	**Responsible**
Version 0.1	02.05.2007	First documentation.	Øystein Godøy
Version 0.2	28.02.2008	Added some more description.	Øystein Godøy

Table of Contents

1 Introduction

1.1 What

`libfmcol` is a C library that is used for collocation of various data sources (e.g. AVHRR imagery, OSISAF, NWCSAF, NWP and in situ observations. It is specifically developed for the needs at the Norwegian Meteorological Institute, but might be useful to other as well.

1.2 Why

`libfmcol` have been developed to collect training and validation data when developing remote sensing products.

1.3 Where

`libfmcol` have been developed for the following purposes:

- Research and development at the Norwegian Meteorological Institute
- Operational processing of remote sensing data at the Norwegian Meteorological Institute

1.4 Who

`libfmcol` have been developed by scientists employed by the Norwegian Meteorological Institute. However, the library is free to be used by anyone that would find it useful. It is distributed using GNU General Public License when it is built without PROJ.4 support. PROJ.4 is distributed using a MIT license. See the PROJ.4 homepage for details on the PROJ.4 license.

The following persons have been involved in the software development:

Name	Function	Affiliation
Øystein Godøy	Coordinator, Software development	Norwegian Meteorological Institute (http://www.met.no/)

2 Compilation and implementation

2.1 Requirements

`libfmcol` have some functionality that relies on the PROJ.4, `libosihdf5` and `libsmhi_saf` libraries. However, the library can be built without these libraries. The functionality requiring PROJ.4 will then be disabled. `libhdf5`, `libfmutil` and `libfmio`

are mandatory libraries.

`libfmcol` have been developed on a Linux system running Fedora Core 3 and Fedora Core 5. Although being developed for portability, this has not been thoroughly tested yet.

2.2 Compilation

The following steps are required to build the software:

1. Unpack the library (distributed as a gzipped tarball) using `tar xvzf libfmcol.tgz`
2. execute `configure`. The `help` option of `configure` will list options. If e.g. PROJ.4 support is not needed, use `without-proj`. If reconfiguration of the system is required, remember to run `make distclean`.
3. `make`
4. `make install`

2.3 Known problems

The library is still not very robust. If read functions are being used to retrieve objects not found in the file this may result in a segmentation fault when operated through e.g. R-software.

3 Description

Basically the library creates an interface between various meteorological data and a collocation file format implemented in NCSA HDF5. The library use other libraries to read e.g. Numerical Weather Prediction Model output from METNO feltfiles, AVHRR information from METSAT-files, NWCSAF PPS Cloud type from HLHDF HDF5 files, OSISAF SSI and DLI from OSIHDF5 files and SYNOP from ASCII files generated using METNO ROBS library (generation of the ASCII files is a pre-processing step).

The following objects are defined within the HDF5 files:

Name	ID#	Description
AVHR	1	AVHRR data as generated by polsatproc using AAPP. Potentially KSPT MEOS files could be read, but support for that format is removed from libfmio due to binary version only of reading functions.
NWP	2	HIRLAM NWP data as generated by METNOs HIRLAM model and stored in METNO feltfiles. remember to check vertical levels if that is to be used at a later point.

STDAT	3	SYNOP or DRAU data as converted to ASCII by ROBS dependent functions.
SAFCM	4	SAFNWC cloud type product in HLHDF format.
SAFSS	5	SAFOSI SSI product in OSIHDF5 format.
SAFDLI	6	SAFOSI DLI product in OSIHDF5 format.

These objects are grouped together within the HDF5 file. Functions that access these objects usually have names of the type `dump123456` or read123456, where the numbers refer to the ID# in the table above.

4 Library variables and structures

4.1 Definitions

4.1.1

```
/*
 * NAME:
 * fmcol.h
 *
 * PURPOSE:
 * Main header file for libfmcol.
 *
 * REQUIREMENTS:
 * NA
 *
 * INPUT:
 * NA
 *
 * OUTPUT:
 * NA
 *
 * NOTES:
 * NA
 *
 * BUGS:
 * NA
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 21/12/2001
 *
 * MODIFIED:
 * Øystein Godøy, DNMI/FOU, 29.04.2003
 * Added specification of geographical position and dummy ids
 * Øystein Godøy, met.no/FOU, 02.04.2004
 * Added support for KSPT libraries on Red Hat.
```

```
 * Øystein Godøy, met.no/FOU, 21.10.2004
 * Øystein Godøy, met.no/FOU, 29.10.2004
 * Øystein Godøy, met.no/FOU, 01.11.2004
 * Øystein Godøy, met.no/FOU, 14.12.2004
 * Implemented and updated NSCENES and NPOSITIONS, added elements to
 * control which tiles to read.
 * Øystein Godøy, met.no/FOU, 23.12.2004
 * See decode_cfg.c for details.
 * Øystein Godøy, met.no/FOU, 28.12.2004
 * Changed keyword from GPOS to DPOS for observations from DIANA,
added
 * GPOS as keyword for strictly geographical positions (equals
STIDS).
 *
 * CVS_ID:
 * $Id: fmcol.h,v 1.4 2007/09/12 14:36:21 steingod Exp $
 */

#ifndef FMCOL_H
#define FMCOL_H

#define STIDSLEN 1024
#define DUMMYIDSLEN 8192
#define FILENAMELEN 256
#define FILELEN 256
#define DUMMYLEN 512
#define NOCHAN 6
#define FMCOLMAXFILES 1024

#define BOXSIZE 13
#define NWPSIZE 1
#define NWPSIZE2D 1

#include <fmcol_config.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <fmutil.h>
#ifdef FMCOL_HAVE_LIBFMIO
#include <fmio.h>
#endif
#include <fmcolaccess.h>

#ifndef FMCOL_BOOL
#define FMCOL_BOOL
typedef enum {FMCOL_FALSE,FMCOL_TRUE} fmcol_boolean;
#endif

#define DBFILES 20 /* Max allowed classes for process */
#define NOWC 8 /* Number of allowed wildcards */
```

```
#define NSCENES 25 /* Initial number of dummy scenes allocated */
#define NPOSITIONS 256 /* Initial number of postions within dummy
scenes allocated */
#define FMCOLIDLEN 50 /* Length of id description in text */
#define FMCOLDESCLEN 50 /* Length of string descriptions */
#define FMCOLAREALEN 10 /* Length of string descriptions for area */

typedef struct {
    fmgeopos pos;
    char desc[FMCOLDESCLEN];
    char area[FMCOLAREALEN];
} gposs;

typedef struct {
    char id[FMCOLIDLEN];
    gposs *pos;
    int npos;
    int nr;
    int ns;
    int gr;
    int at;
} scenes;

typedef struct {
    scenes *scene;
    int nscenes;
} dids;

typedef struct {
    gposs *pos;
    int npos;
} gposses;

typedef struct {
    char avhrr_p[FILENAMELEN];
    char synop_p[FILENAMELEN];
    char nwp_p[FILENAMELEN];
    char safcm_p[FILENAMELEN];
    char safssi_p[FILENAMELEN];
    char safdli_p[FILENAMELEN];
    char avhrr_f[NAMELEN];
    char synop_f[NAMELEN];
    char nwp_f[NAMELEN];
    char safcm_f[NAMELEN];
    char safssi_f[NAMELEN];
    char safdli_f[NAMELEN];
    char lp_f[NAMELEN];
    char dbfile[DBFILES][FILENAMELEN];
    char classname[DBFILES][NAMELEN];
    char *stids; /* String containing station identifications */
    dids did; /* Dummy scenes from UFFDA/DIANA */
```

```
    gposses geopos; /* geogr. positions */
    int nocl; /* Number of classes / storage files */
    fmcol_boolean synop; /* Determines whether to use only synoptic
times */
    fmcol_boolean snow; /* Do I use this?? */
} cfgstruct;

typedef struct {
    time_t unixtime;
    char name[FILENAMELEN];
} fnsstruct;

typedef struct {
    fnsstruct *f;
    int n;
} fstruct;

void usage(void);
void error(char *where, char *what);
int decode_cfg(char cfgfile[],cfgstruct *cfg);
int check_avhrr(char *ipath, char *basename, fstruct *avhrr_f,
      time_t older, time_t newer);
int check_safct(char *ipath, char *basename, fstruct *safct_f,
      time_t older, time_t newer);

#endif /* FMCOL_H */
```

4.1.2 fmcolaccess.h

```
/*
 * NAME:
 * fmcolaccess.h
 *
 * PURPOSE:
 * Header file for the HDF5 software used to store collocated
stattionwise
 * data.
 *
 * REQUIREMENTS:
 * NA
 *
 * INPUT:
 * NA
 *
 * OUTPUT:
 * NA
 *
 * NOTES:
 * NA
 *
 * BUGS:
```

```
 * NA
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 27/07/2000
 *
 * MODIFIED:
 * Øystein Godøy, DNMI/FOU, 19/09/2002: Modularized HDF access, and
added
 * support for SAFCM...
 * Øystein Godøy, DNMI/FOU, 29.04.2003: Added storage without STDAT
 * Øystein Godøy, MI/FOU, 06.11.2003: Due to errors when extracting
 * safcm_data data, a dummy interface is created to circumvent the
problem
 * in the current situation. This involves the use of a slimmer data
 * structure not containing the character strings describing the
class
 * names. Functions affected are the high level read* functions
accessing
 * the safcm_data structure. In this fix, data are transferred from
 * safcm_data to safcm_data2.
 * Øystein Godøy, met.no/FOU, 20.12.2004: Added dump124.
 * Øystein Godøy, met.no/FOU, 23.12.2004: Added support for SAFDLI.
 * Øystein Godøy, METNO/FOU, 10.09.2007: Rewrote hader file use.
 * Øystein Godøy, METNO/FOU, 28.02.2008: Added read124
 *
 * CVS_ID:
 * $Id: fmcolaccess.h,v 1.2 2007/09/17 07:17:50 steingod Exp $
 */

#include <fmcol_config.h>

#ifndef _HDFACCESS_H
#define _HDFACCESS_H

#ifdef FMCOL_HAVE_LIBHDF5
/*
 * Header files that are required.
 */
#include <hdf5.h>
#include <fmutil.h>
/*
#include <avhrr_stdat.h>
#include <safcm_stdat.h>
#include <safssi_stdat.h>
#include <safdli_stdat.h>
#include <nwp_stdat.h>
#include <std_stdat.h>
*/

#define MAXGROUPS 10000
#define MAXDATASETS 10
```

```
#define NAMELEN 256
#define BOXSIZE2D 169
#define FMCOL_NO_CLOUDTYPE_VALUES 21

/*
 * Variable declarations...
 */
typedef struct {
    char classname[NAMELEN];
    char station[NAMELEN];
    time_t t_start;
    time_t t_end;
} skeys;

typedef struct {
    int noobs;
    skeys *scrit;
    char (*group)[NAMELEN];
} cinfo;

typedef struct {
    char source[20];
    int nochan;
    int nopix;
    time_t vtime;
    fmucsref nav;
    fmangles ang;
    float ch1[BOXSIZE2D];
    float ch2[BOXSIZE2D];
    float ch3a[BOXSIZE2D];
    float ch3b[BOXSIZE2D];
    float ch4[BOXSIZE2D];
    float ch5[BOXSIZE2D];
} as_data;

typedef struct {
    char stID[10];
    char stType[5]; /* ?? */
    fmgeopos pos;
    int year;
    short month;
    short day;
    short hour;
    short min;
    float TTT;
    float TdTdTd;
    float TwTwTw;
    float PPPP;
    short ff;
    int dd;
```

```
    float RRR;
    short E;
    short sss;
    short N;
    short Nh;
    short Cl;
    short Cm;
    short Ch;
    short VV;
    short ww;
    short W1;
    short W2;
} obsstruct;

typedef struct {
    char type[5];
    fmtime timeid0;
    fmtime timeid1;
    fmtime timeid2;
    int unixtime0;
    int unixtime1;
    int unixtime2;
    int noobs;
    int nostat;
    obsstruct *obs;
} stdatstruct;

typedef struct {
    int nopar;
    int nopix;
    time_t vtime;
    int ltime;
    fmucsref nav;
    float t0m[BOXSIZE2D]; /* 0m temp */
    float t2m[BOXSIZE2D]; /* 2m temp */
    float t950hpa[BOXSIZE2D]; /* temp at 950 hPa */
    float t800hpa[BOXSIZE2D]; /* temp at 800 hPa */
    float t700hpa[BOXSIZE2D]; /* temp at 700 hPa */
    float t500hpa[BOXSIZE2D]; /* temp at 500 hPa */
    float ps[BOXSIZE2D]; /* mean sea level pressure */
    float topo[BOXSIZE2D]; /* model topography */
    float pw[BOXSIZE2D]; /* precipitable water */
    float rh[BOXSIZE2D]; /* relative humidity at surface */
} ns_data;

typedef struct {
    char source[20];
    int nopix;
    time_t vtime;
    fmucsref nav;
    unsigned char data[BOXSIZE2D];
```

```
    char description[FMCOL_NO_CLOUDTYPE_VALUES][FMSTRING256];
} safcm_data;

typedef struct {
    char source[20];
    int nopix;
    time_t vtime;
    unsigned char data[BOXSIZE2D];
} safcm_data2;

typedef struct {
    char source[20];
    int nopix;
    time_t vtime;
    fmucsref nav;
    float data[BOXSIZE2D];
    unsigned short qflg[BOXSIZE2D];
} safssi_data;

typedef struct {
    char source[20];
    int nopix;
    time_t vtime;
    fmucsref nav;
    float data[BOXSIZE2D];
    unsigned short qflg[BOXSIZE2D];
} safdli_data;


/*
 * Top level functions to use in main program
 * The top level functions are named according to a numbering of
data
 * sources (see fmcolaccess.c for correct list at any time).
 *    AVHRR - 1 - AVHRR data as given by MEOS
 *    NWP   - 2 - HIRLAM NWP data
 *    STDAT - 3 - SYNOP or DRAU data
 *    SAFCM - 4 - SAFNWC cloud type product
 *    SAFSSI      - 5 - SAFOSI SSI product
 *    SAFDLI      - 6 - SAFOSI DLI product
 */
int dump123(char *filename, char *class,
      as_data aobs, ns_data nobs, obsstruct sobs);
#ifdef FMCOL_HAVE_LIBPPS_COMMON
int dump1234(char *filename, char *class,
      as_data aobs, ns_data nobs, obsstruct sobs, safcm_data cobs);
int dump124(char *filename, char *class,
      as_data aobs, ns_data nobs, safcm_data cobs);
#ifdef FMCOL_HAVE_LIBOSIHDF5
int dump12345(char *filename, char *class,
      as_data aobs, ns_data nobs, obsstruct sobs,
```

```
      safcm_data cobs, safssi_data ssi);
int dump123456(char *filename, char *class,
      as_data aobs, ns_data nobs, obsstruct sobs,
      safcm_data cobs, safssi_data ssi, safdli_data dli);
int dump1245(char *filename, char *class,
      as_data aobs, ns_data nobs,
      safcm_data cobs, safssi_data ssi);
int dump12456(char *filename, char *class,
      as_data aobs, ns_data nobs,
      safcm_data cobs, safssi_data ssi, safdli_data dli);
#endif
#endif
#ifdef FMCOL_HAVE_LIBPPS_COMMON
int dump34(char *filename, char *class,
      obsstruct sobs, safcm_data cobs);
#ifdef FMCOL_HAVE_LIBOSIHDF5
int dump456(char *filename, char *class,
      safcm_data cobs, safssi_data iobs, safdli_data dobs);
#endif
#endif
#ifdef FMCOL_HAVE_LIBPPS_COMMON
#ifdef FMCOL_HAVE_LIBOSIHDF5
int read12345(char *filename, skeys *scrit,
      as_data **aobs, ns_data **nobs, obsstruct **sobs,
      safcm_data2 **cobs, safssi_data **iobs);
int read123456(char *filename, skeys *scrit,
      as_data **aobs, ns_data **nobs, obsstruct **sobs,
      safcm_data2 **cobs, safssi_data **iobs, safdli_data **dobs);
int read1245(char *filename, skeys *scrit,
      as_data **aobs, ns_data **nobs,
      safcm_data2 **cmobs, safssi_data **iobs);
int read12456(char *filename, skeys *scrit,
      as_data **aobs, ns_data **nobs,
      safcm_data2 **cmobs, safssi_data **iobs, safdli_data **dobs);
#endif
int read1234(char *filename, skeys *scrit,
      as_data **aobs, ns_data **nobs, obsstruct **sobs,
      safcm_data2 **cobs);
int read124(char *filename, skeys *scrit,
      as_data **aobs, ns_data **nobs, safcm_data2 **cobs);
#endif
int read123(char *filename, skeys *scrit,
      as_data **aobs, ns_data **nobs, obsstruct **sobs);
#ifdef FMCOL_HAVE_LIBPPS_COMMON
int read34(char *filename, skeys *scrit, obsstruct **sobs,
safcm_data2 **cobs);
#ifdef FMCOL_HAVE_LIBOSIHDF5
int read456(char *filename, skeys *scrit,
      safcm_data2 **cmobs, safssi_data **iobs, safdli_data **dobs);
#endif
#endif
```

```
int checknorec(char *filename, skeys *scrit);

/*
 * Functions below are only used by the top level functions, not in
the
 * main program.
 * Data dump to file
 */
hid_t openFile(char *filename);
herr_t read_items(hid_t loc_id, const char *name, void *opdata);
herr_t count_basegroups(hid_t loc_id, const char *name, void
*opdata);
herr_t count_datasets(hid_t loc_id, const char *name, void *opdata);
herr_t dumpAVHRR2HDF(hid_t file, hid_t grp, as_data aobs);
herr_t dumpNWP2HDF(hid_t file, hid_t grp, ns_data nobs);
herr_t dumpSTDAT2HDF(hid_t file, hid_t grp, obsstruct sobs);
#ifdef FMCOL_HAVE_LIBPPS_COMMON
herr_t dumpSAFCM2HDF(hid_t file, hid_t grp, safcm_data cmobs);
#endif
#ifdef FMCOL_HAVE_LIBOSIHDF5
herr_t dumpSAFSSI2HDF(hid_t file, hid_t grp, safssi_data ssiobs);
herr_t dumpSAFDLI2HDF(hid_t file, hid_t grp, safdli_data dliobs);
#endif
herr_t readSTDAT(hid_t file, hid_t grp, obsstruct *obs);
herr_t readAVHRR(hid_t file, hid_t grp, as_data *obs);
herr_t readNWP(hid_t file, hid_t grp, ns_data *obs);
#ifdef FMCOL_HAVE_LIBPPS_COMMON
/*herr_t readSAFCM(hid_t file, hid_t grp, safcm_data *obs);*/
herr_t readSAFCM(hid_t file, hid_t grp, safcm_data2 *obs);
herr_t readSAFCMRAW(hid_t file, hid_t grp, safcm_data2 *obs);
#endif
#ifdef FMCOL_HAVE_LIBOSIHDF5
herr_t readSAFSSI(hid_t file, hid_t grp, safssi_data *obs);
herr_t readSAFDLI(hid_t file, hid_t grp, safdli_data *obs);
#endif

/*
 * Compound datatypes required
 */
hid_t compoundAVHRR(int nopix);
hid_t compoundNWP(int nopix);
hid_t compoundSTDAT(void);
hid_t compoundSAFCM(int nopix);
hid_t compoundSAFCMRAW(int nopix);
hid_t compoundSAFSSI(int nopix);
hid_t compoundSAFDLI(int nopix);
hid_t compoundUCS(void);
hid_t compoundIMGA(void);
hid_t compoundLATLON(void);

/*
```

```
 * End function prototypes.
 */

#endif /* FMCOL_HAVE_LIBHDF5 */
#endif /* _HDFACCESS_H */
```

4.1.3 avhrr_stdat.h

```
/*
 * NAME:
 * avhrr_stdat.h
 *
 * PURPOSE:
 * Header file for avhrr_stdat.c, see avhrr_stdat.c for details.
 *
 * REQUIREMENTS:
 * NA
 *
 * INPUT:
 * NA
 *
 * OUTPUT:
 * NA
 *
 * NOTES:
 * NA
 *
 * BUGS:
 * NA
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 06/11/2000
 *
 * MODIFIED:
 * Øystein Godøy, MI/FOU, 11.04.2003
 * Øystein Godøy, met.no/FOU, 22.12.2004
 * Changed storage structure, this needs to be reflected by
fmcolaccess.c as
 * well...
 * Øystein Godøy, METNO/FOU, 25.04.2007: Adapted for
libfmutil/libfmio.
 *
 * CVS_ID:
 * $Id: avhrr_stdat.h,v 1.6 2007/09/12 14:36:21 steingod Exp $
 */

#include <stdio.h>
#include <stdlib.h>

#include <fmcol.h>
#include <fmcolaccess.h>
```

```
/*
#include <fmcoord.h>
#include <fmsubtrack.h>
*/

#ifndef _AVHRR_STDAT_H
#define _AVHRR_STDAT_H

#define OUTOFIMAGE -40100
#define MISVAL -99999
#define MAXCH 5

int avhrr_stdat(fmgeopos gpos, fmprojspec myproj, fmio_img img,
as_data *a);
int avhrr_init(int boxsize, int nochan, as_data *a);
int avhrr_free(as_data *a);

#endif /* _AVHRR_STDAT_H */
```

4.1.4 nwp_read.h

```
/*
 * NAME:
 * nwp_read.h
 *
 * PURPOSE:
 * To provide an interface between libusenwp and libfmcol.
 *
 * REQUIREMENTS:
 * NA
 *
 * INPUT:
 * NA
 *
 * OUTPUT:
 * NA
 *
 * NOTES:
 * NA
 *
 * BUGS:
 * NA
 *
 * AUTHOR:
 * Øystein Godøy, met.no/FOU, 18.10.2004
 *
 * MODIFIED:
 * Øystein Godøy, METNO/FOU, 17.10.2007: Removed hardcoded
filenames.
 *
 * CVS_ID:
```

```
 * $Id: nwp_read.h,v 1.6 2007/10/17 11:43:44 steingod Exp $
 */

#include <stdio.h>
#include <stdlib.h>

#include <fmutil.h>
#include <fmcol.h>
#include <fmcolaccess.h>

#ifndef NWP_READ
#define NWP_READ

#define NOFIELDS1 9
#define NOFIELDS2 1
#define NOFIELDS NOFIELDS1+NOFIELDS2

typedef struct {
    fmsec1970 validtime;
    int leadtime;
    fmucsref refucs;
    float *t950hpa; /* temp at 950 hPa */
    float *t800hpa; /* temp at 800 hPa */
    float *t700hpa; /* temp at 700 hPa */
    float *t500hpa; /* temp at 500 hPa */
    float *t0m; /* surface temp */
    float *t2m; /* 2m temp */
    float *ps; /* mean sea level pressure */
    float *topo; /* model topography */
    float *pw; /* precipitable water */
    float *rh; /* relative humidity at surface */
} nwpdata;

int nwpdata_init(nwpdata *nwp);
int nwpdata_read(char *nwppath, fmfilelist surflist, fmfilelist
levlist,
      fmtime reqtime, fmucsref refucs, nwpdata *nwp);
int nwpdata_free(nwpdata *nwp);

#endif /* NWP_READ */
```

4.1.5 nwp_stdat.h

```
/*
 * NAME:
 * nwp_stdat.h
 *
 * PURPOSE:
 * Header file for nwp_stdat.c
 *
 * REQUIREMENTS:
```

```
 * NA
 *
 * INPUT:
 * NA
 *
 * OUTPUT:
 * NA
 *
 * NOTES:
 * NA
 *
 * BUGS:
 * NA
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 28/01/2002
 *
 * MODIFIED:
 * Øystein Godøy, MI/FOU, 16.04.2003
 * Øystein Godøy, met.no/FOU, 18.10.2004: NWP interface is changed.
 *
 * CVS_ID:
 * $Id: nwp_stdat.h,v 1.5 2007/09/12 14:36:22 steingod Exp $
 */

#include <stdio.h>
#include <stdlib.h>
#include <fmutil.h>
#include <fmcol.h>
#include <nwp_read.h>
#include <fmcolaccess.h>

/*
#include <fmcoord.h>
*/

#ifndef _NWP_STDAT_H
#define _NWP_STDAT_H

#define OUTOFIMAGE -40100
#define MISVAL -99999
#define UNDEF +1.E+35

int nwp_stdat(fmgeopos gpos, fmprojspec myproj, fmucsref refucs,
      nwpdata nwp, ns_data *n);
int nwp_init(int boxsize, float boxresx, float boxresy, int nopar,
ns_data *n);
int nwp_free(ns_data *n);

#endif /* _NWP_STDAT_H */
```

4.1.6 safssi_stdat.h

```
/*
 * NAME:
 * safssi_stdat.h
 *
 * PURPOSE:
 * Header file for safssi_stdat.c
 *
 * REQUIREMENTS:
 * NA
 *
 * INPUT:
 * NA
 *
 * OUTPUT:
 * NA
 *
 * NOTES:
 * NA
 *
 * BUGS:
 * NA
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 27/09/2002
 *
 * MODIFIED:
 * Øystein Godøy, MI/FOU, 11.04.2003
 *
 * CVS_ID:
 * $Id: safssi_stdat.h,v 1.5 2007/09/12 14:36:22 steingod Exp $
 */

#ifndef _SAFSSI_STDAT_H
#define _SAFSSI_STDAT_H

#include <dirent.h>

#include <fmutil.h>
#include <fmcol.h>
#include <fmcolaccess.h>
#ifdef FMCOL_HAVE_LIBOSIHDF5
#include <safhdf.h>

#define SAFSSI_MISVAL -999.

int safssi_stdat(fmgeopos gpos, fmprojspec myproj,
      osihdf *safssi, safssi_data *ssi);
int safssi_init(int boxsize, safssi_data *ssi);
int safssi_free(safssi_data *ssi);
```

```
char *safssi_prod(char *filename);
int safssi_name(fmtime refdate, char *basename, char *path, char
*filename);
int safssi_check(fmio_img refdata, osihdf *safssi);

#endif /* FMCOL_HAVE_LIBOSIHDF5 */
#endif /* _SAFSSI_STDAT_H */
```

4.1.7 safcm_stdat.h

```
/*
 * NAME:
 * safcm_stdat.h
 *
 * PURPOSE:
 * Header file for safcm_stdat.c
 *
 * REQUIREMENTS:
 *
 * INPUT:
 *
 * OUTPUT:
 *
 * NOTES:
 *
 * BUGS:
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 06/11/2000
 *
 * MODIFIED:
 * Øystein Godøy, MI/FOU, 06.11.2003
 * Due to errors when extracting these data, a dummy interface is
 * created to circumvent the problem in the current situation. This
 * involves the use of a slimmer data structure not containing the
 * character strings describing the class names.
 *
 * CVS_ID:
 * $Id: safcm_stdat.h,v 1.7 2007/09/12 14:36:22 steingod Exp $
 */

#ifndef _SAFCM_STDAT_H
#define _SAFCM_STDAT_H

#include <fmutil.h>
#include <dirent.h>
#include <fmcol.h>
#include <fmcolaccess.h>

#ifdef FMCOL_HAVE_LIBPPS_COMMON
```

```
#include <pps_cloudproducts_io.h>

#define M2KM(x) (x/1000.00)

#define SAFCM_MISVAL 255

/*
 * Probably a flag to determine whether debugging info should be
output to
 * logs or not, a non-zero number is TRUE probably...
 */
/*int sm_debug = 1;*/

int safcm_stdat(fmgeopos gpos, fmprojspec myproj,
      CTy_t *ctype, safcm_data *cm);
int safcm_init(int boxsize, safcm_data *cm);
int safcm_free(safcm_data *cm);
char *safcm_prod(char *filename);
int safcm_name(fmtime refdate, char *basename, char *path, char
*filename);
int safcm_check(fmio_img refdata, CTy_t *ctype);

#endif /* FMCOL_HAVE_LIBPPS_COMMON */
#endif /* _SAFCM_STDAT_H */
```

4.1.8 stdats4hdf.h

```
/*
 * PURPOSE:
 * Header file for the HDF5 software used to store collocated
stattionwise
 * data.
 *
 * NOTE:
 * See stdats2hdf.c for details.
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 27/07/2000
 */

#ifndef _STDATS4HDF_H
#define _STDATS4HDF_H

/*
 * Header files that are required.
 */
#include <fmutil.h>
#include <hdf5.h>
#include <avhrr_stdat.h>
#include <std_stdat.h>
```

```
#include <nwp_stdat.h>

/*
 * Function prototypes.
 */
int stdats2hdf(char *filename, char *class,
      obsstruct sobs, as_data aobs, ns_data nobs);
/*
short read_hdf5_product(char *filename, osihdf *f, short mode);
short init_osihdf(osihdf *f);
short malloc_osihdf(osihdf *f, osi_dtype *dfs, char **desc);
short free_osihdf(osihdf *f);
*/

/*
 * End function prototypes.
 */

#endif /* _STDATS4HDF_H */
```

4.1.9 std_stdat.h

```
/*
 * NAME:
 * std_stdat.h
 *
 * PURPOSE:
 * Header file for std_stdat.c software.
 *
 * REQUIREMENTS:
 * NA
 *
 * INPUT:
 * NA
 *
 * OUTPUT:
 * NA
 *
 * NOTES:
 * NA
 *
 * BUGS:
 * NA
 *
 * AUTHOR:
 * Øystein Godøy, DNMI/FOU, 25/01/2002
 *
 * MODIFIED:
 * NA
 *
 * CVS_ID:
```

```
 * $Id: std_stdat.h,v 1.4 2007/09/12 14:36:22 steingod Exp $
 */

#ifndef _STD_STDAT_H
#define _STD_STDAT_H

#include <fmutil.h>
#include <fmcolaccess.h>

/*
 * Time differences are given in seconds
 */
#define SDIFF 5400 /* Accepted time difference from synop time */
#define TSDIFF 1800 /* Generally acceppted time difference syno */
#define TDDIFF 2700 /* Generally acceppted time difference drau */
#define STDATSTRLEN 1024

int read_stdat(char *ipath, char *basename, fmcol_boolean synop,
      time_t timeid, stdatstruct *pobs);
int get_stdat_h(char *infile, stdatstruct *obsl);
int fill_stdat_h(char *infile, time_t vtime, stdatstruct *obsl);
int init_stdat(stdatstruct *obsl);
int free_stdat(stdatstruct *obsl);

#endif /*  _STD_STDAT_H */
```

5 Library functions

5.1 Writing

5.2 Reading

5.3 Internal