Programmer's Manual for Software module, QIFF, to convert ASCII IFF Files to CDF Files

A.J. Allen
Queen Mary & Westfield College
Mile End Road,
London E1 4NS
e-mail A.J.Allen@qmw.ac.uk

Document Status Sheet			
1. Document Title: **QIFF Manual**			
2. Document Reference Number: **CU-QMW-MA-0004**			
3. Issue	4. Revision	5. Date	6. Reason for Change
1	0	6 Jan 95	First Issue.
1	1	10 Jan 95	After Internal Review.
1	2	2 Feb 95	Addition of -log flag. Clean exit on ERROR detection.
1	2	3 June 97	Allow T-offset, var name extn and data period overrides. For public release.

0.1 Applicable Documents

The following documents are applicable to this manual and provide details not explicitly set out here.

- 'Interface Control Document between the UK CDHF and the UK PIs', CU-RAL-ID-0001
- 'Reference Document for CSDS CDF Implementation', DS-QMW-TN-0003
- 'Definition of the Interface between Data Reduction and CDF Creation Software for the UK CSDS Data Products', CU-QMW-TN-0003
- 'CSDS Standard Skeleton Files', DS-QMW-TN-0005
- 'Process Control for the Routine Production of the UK CSDS Data Products', CU-QMW-TN-0005
- 'Detailed Design Document for the UK CDHF', DS-RAL-DD-0001
- 'Test Report on UK CDHF Software Module, QIFF', CU-QMW-TR-0001

1 Introduction

This manual provides support for the use of the CDHF software module QIFF which converts ascii 'Interface File Format', iff, files into 'Common Data Format', cdf, files.

This manual is provided in the format of a Unix man page for inclusion with the software distribution.

2 Man Page

NAME

QIFF - Module to convert ascii .iff files to CSDS .cdf files.

SYNOPSIS

as an executable utility...

```
iff_to_cdf [-nodelete] [-log  log_path] iff_path cdf_path file_stem
```

or as a function...

```
long QiIFFtoCDF (char *iff_fname,
                 char *cdf_fname,
                 char *log_fname,
                 struct QiSFName *QiSname)
```

DESCRIPTION

AS A UTILITY

The executable utility is provided for use in the UK CDHF pipeline data reduction system. A simple main function calls the function QiIFFtoCDF with argument derived from the input arguments.

The required argument iff_path provides the full path to the directory containing the input iff file. Separate iff files must be provided for each spacecraft for each instrument at each resolution.

The required argument cdf_path provides the full path to the directory containing the empty cdf file. This file must have been created from the CSDS skeleton files for the instrument and spacecraft covered by the iff file. The cdf file is not moved or renamed by the utility, but is populated with the data from the iff file.

The required argument file_stem is the unique file name stem of the form SC_RR_III_YYYYMMDD where SC is the spacecraft identifier (C1, C2, C3, C4 or CL for Cluster), RR is the resolution (PP or SP for Cluster), III is the three character instrument identifier and YYYYMMDD is the date of the start of the data in the file as the eight character string year month day. This file stem is used to construct the name of the expected input iff file, the name of the empty cdf file and the name of a logfile

```
    used for processing messages.

    If a fatal error is detected from either the software or the
    input iff file, the partially formed cdf file is automatically
    deleted before the utility exits. This ensures that pipeline
    operation does not continue with a badly formed file. For debug
    purposes this may be turned off using the optional -nodelete flag
    before the required arguments.

    The optional flag -log is used to indicate that the next input argument
    is the path to the directory in which to place the logfile. The log file
    is always derived from the file_stem argument.

 AS A FUNCTION

    Returns a long integer as status. Values are:-
    QMW_OK ( 0 ) on successful completion and
    QMW_ERROR ( -1 ) on failure.

    The arguments are...

IN: char *iff_fname, pointer to character string containing full name
    and path of iff file to be read (including .iff extension).

IN: char *cdf_fname, pointer to character string containing full name
    and path of empty cdf file to be populated (including .cdf extension).

IN: char *log_fname, pointer to character string containing full name
    and path of file to be used for writing log messages.

IN: struct QiSFName *QiSname, pointer to structure containing the
    generic filename and date. This structure is defined in
    IFF_to_CDF.h and takes the following form

    struct QiSFName {
        char fname[NAME_LEN];         /* file name root */
        char sub_strings[NAME_LEN];   /* dummy to hold sc, rr, iii */
        char *sc;                     /* s/c id, e.g. Cn or CL */
        char *rr;                     /* resolution, e.g. SP or PP */
        char *iii;                    /* instrument abbreviation */
        long year;                    /* year */
        long month;                   /* month */
        long day;                     /* day */
        char *extn;                   /* optional file name extension */
    };
```

SAMPLE USE

```
#/bin/csh
set IFFPATH=/cluster/devel/pipeline/iff/DWP
set CDFPATH=.
set SKTPATH=/cluster/devel/CSDS_CDF/csds_cdf_skeletons/DWP
set BIN=.
set CDFBIN=/cluster/cdf/bin
set FILE=C3_PP_DWP
set DAY=_19970123
set CDFFILE=$FILE$DAY

#remove existing cdf file of same name if present

if(-e $CDFPATH/$CDFFILE'_V00'.cdf)  'rm'   $CDFPATH/$CDFFILE'_V00'.cdf;

#generate prototype cdf from .skt file

$CDFBIN/skt2cdf  -cdfname $CDFPATH/$CDFFILE'_V00' -fillval $SKTPATH/$FILE;

# transfer content of .iff file to .cdf file

$BIN/iff_to_cdf $IFFPATH $CDFPATH  $CDFFILE ;
```

CUSTOMIZATION

The included file IFF_to_CDF.h contains modifiable definitions that may be changed for customization to different Data Centres.

In particular the definition

```
#define GEN_BY  "Generated by UK CDHF."
```

MUST be changed to identify the Data Centre or instrument team running the software.

Several array dimensions are specified as definitions and may be changed. Particularly the number of variables allowed. If these are increased the number of words should also be increased. Defaults are:-

```
#define NUMBER_OF_WORDS 240
#define MAX_NUMBER_OF_VARS 40
#define MAX_NUMBER_OF_DIMS 10
#define NAME_LEN 30
#define WORD_LEN 120
#define FULL_NAME_LEN 240
```

No limit is placed on the number of records in an input file, nor the period of time it covers. Input lines may be of any length provided that the number of distinct words or variables contained

by each line does not exceed the above limits.

'Words' are items on the input line, such as values in records. A quoted string is treated as a single word, so that attribute entries should not exceed the length WORD_LEN - 1.

To extend use of this pipeline software to other mission data some optional override flags are provided, see syntax in CU-QMW-TN-0003.

Time offset: This may optionally be specified. If present it is used as the start time to be added to every millisecond epoch value in place of the start of day as derived from the file name.

Data Period: The half interval is determined automatically for Cluster SP and PP data. For other missions, inclusion of a data periodicity line becomes essential to override the Cluster default. See CU-QMW-TN-0003.

Variable names: Cluster variables comprise a stem identifying the data product and an extension for spacecraft, resolution and instrument. This extension is taken from the skeleton file, except for the Half_interval variable and attributes for the Epoch variable where it must be calculated internally. It is possible to specify an alternative variable extension in the iff file, and this is used in place of that derived from the file name.

File Names: This software requires the file name to be of the Cluster syntax as part of the pipeline validation control. This restriction can only be avoided by renaming the cdf file after processing and by using the above override options. The software will then warn in the logfile that the Logical_file_id differs from the file name, but this may be safely ignored.

WARNING/ERROR MESSAGES

The following occurences cause the program to exit with an error message:

Specified iff file not found.
Specified cdf file not found.
Fatal Error flag received from PI software via iff file.
Time tags not monotonic increasing.

The following occurences cause a warning message to be written to the log file. They arise from syntax errors in the input iff file, but an output cdf file will be generated. Such warnings should always be investigated as they may result in loss of data or

```
    corrupt/missing attribute information.

      Unable to attach logfile for writing (stderr used as default).
      File name and Logical_file_id attribute incompatible.
      Null input file (no records present in input file).
      Incorrect syntax or number of fill values.
      Incorrect V attribute line syntax.
      Incorrect G attribute line syntax.
      Unknown V or G attribute or variable name.
      Incorrect data record syntax.
      Data value impossible for data type.
      Incorrect number of data entries in record line.
      Unmarked data gaps (time tag spacing outside tolerance).
      Number of words exceeds limit set in IFF_to_CDF.h
      Length of word (or quoted string) exceeds limit set in IFF_to_CDF.h
      End of data marker inconsistent with number of records supplied.
      Records appear after end of data marker.

    Other messages may be generated by the software, these represent
    internal or cdf errors and should be reported to QMW for rectification.


AUTHOR
     A.J.Allen QMW 2-3-95

DOCUMENTATION
     'Interface Control Document between the UK CDHF and the UK PIs',
     CU-RAL-ID-0001

     'Reference Document for CSDS CDF Implementation',
     DS-QMW-TN-0003

     'Definition of the Interface between Data Reduction and
      CDF Creation Software for the UK CSDS Data Products',
     CU-QMW-TN-0003

     'CSDS Standard Skeleton Files',
     DS-QMW-TN-0005

     'Process Control for Routine Production of UK CSDS Data Products',
     CU-QMW-TN-0005

     'Detailed Design Document for the UK CDHF',
     DS-RAL-DD-0001

BUGS
```

Segmentation faults may arise from files which exceed the defined maximum number of variables or number of words in each input line. These are not always trapped.

FILES

IFF_to_CDF.h
iff_to_cdf.c
CSDS skeletons and prototype cdf files
cdf library