Astronomical Data Analysis Software and Systems VII
ASP Conference Series, Vol. 145, 1998
R. Albrecht, R. N. Hook and H. A. Bushouse, eds.

IRAF Multiple Extensions FITS (MEF) Files Interface

Nelson Zarate

National Optical Astronomy Observatories

Tucson, AZ 85719 (zarate@noao.edu)

Abstract. The Multiple Extension FITS (MEF) file interface is an IRAF library providing facilities for general file operations upon FITS multi-extension files. The MEF library has been used as the basis for a set of new IRAF tasks providing file level operations for multi-extension files. These operations include functions like listing extensions, extracting, inserting, or appending extensions, deleting extensions, and manipulating extension headers. MEF supports extensions of any type since it is a file level interface and does not attempt to interpret the contents of a particular extension type. Other IRAF interfaces such as IMIO (the FITS image kernel) and STSDAS TABLES are available for dealing with specific types of extensions such as the IMAGE extension or binary tables.

1. Introduction

The Multiple Extensions FITS (MEF) interface consists of a number of routines to mainly read a FITS Primary Data Unit or an Extension Unit and manipulate the data at a file level. It is up to the application to take care of any details regarding data structuring and manipulation. For example, the MEF interface will read a BINTABLE extension and give to the calling program a set of parameters like dimensionality, datatype, header buffer pointer and data portion offset from the beginning of the file.

Currently the routines available to an SPP program are:

- `mef = mef_open (fitsfile, acmode, oldp)`
- `mef_rdhdr (mef, group, extname, extver)`
- `mef_rdhdr_exnv (mef, extname, extver)`
- `mef_wrhdr (mefi, mefo, in_phdu)`
- `[irdb]val = mefget[irdb] (mef, keyword)`
- `mefgstr (mef, keyword, outstr, maxch)`
- `mef_app_file (mefi, mefo)`
- `mef_copy_extn (mefi, mefo, group)`

- `mef_dummyhdr (fd, hdrfname)`

  [irdb]: int, real, double, boolean.

2. Initializing Routine

2.1. mef = mef_open (fitsfile, acmode, oldp)

Initializes the MEF interface. Should be the first routine to be called when performing operations on FITS files using this set of routines. Returns a pointer to the MEF structure.

fitsfile Pathname to the FITS file to be open. The general syntax is:

dir$root.extn[group]

- dir: Directory name where the file resides
- root: Rootname
- extn: (optional) Extension name — can be any extension string
- group: Extension number to be opened

The '[group]' string is optional and is not part of the disk filename. It is used to specified which extension number to open. The extension number is zero based — zero for the primary extension, 1 for the first extension, and so on.

acmode The access mode of the file. The possible values are:

READ_ONLY, READ_WRITE, APPEND, NEW_FILE

oldp Not used. Reserve for future use.

3. Header Routines

3.1. mef_rdhdr (mef, group, extname, extver)

Read the FITS header of a MEF file that matches the EXTNAME or EXTVER keyword values or if not specified, read the extension number 'group'. If no extension is found an error is posted. After reading the header the file pointer is positioned at the end of the last data FITS block (2880 bytes).

mef The MEF pointer returned by mef_open. When the routine returns, all of the elements of the MEF structure will have values belonging to the header just read.

group The extension number to be read — zero for the Primary Data Unit, 1 for the first extension, and so on. If you want to find out an extension by the value of extname and/or extver then 'group' should be -1.

extname The string that will match the EXTNAME value of any extension. The first match is the extension header returned.

extver The integer value that will match the EXTVER value of any extension. If 'extname' is not null then both values need to match before the routine returns. If there are no values to match then 'extver' should be INDEFL.

3.2. mef_rdhdr_gn (mef,group)

Read extension number 'group'. If the extension number does not exist, an error is posted.

mef The MEF pointer returned by mef_open. When the routine returns, all of the elements of the MEF structure will have values belonging to the header just read.

group The extension number to be read — zero for the Primary Data Unit, 1 for the first extension, and so on.

3.3. mef_rdhdr_exnv (mef,extname, extver)

Read group based on the Extname and Extver values. If the group is not encountered, an error is posted.

mef The MEF pointer returned by mef_open. When the routine returns, all of the elements of the MEF structure will have values belonging to the header just read.

extname The string that will match the EXTNAME value of any extension. The first match is the extension header returned.

extver The integer value that will match the EXTVER value of any extension. If 'extname' is not null then both values need to match before the routine returns. If there are no value to match then 'extver' should be INDEFL.

3.4. mef_wrhdr (mefi, mefo, in_phdu)

Append the header from an input PHU or EHU to output file.

mefi The input file MEF pointer returned by mef_open. The header should have been read by now.

mefo The output file MEF pointer returned by mef_open.

in_phdu Boolean value (true, false) stating whether the input header is the primary header or not.

3.5. [irdb]val = mefget[irdb] (mef, keyword)

[irdb]: integer, real, double or boolean.

Get a FITS header keyword value of the specified datatype; for example 'imgeti (mef, "NCOMB")' will return an integer value from the keyword 'NCOMB'.

mef The input file MEF pointer returned by mef_open. The header should have been read by now.

keyword The input string (case insensitive) keyword from which to return its value.

3.6. mefgstr (mef, keyword, outstr, maxch)

Get the string value of a FITS encoded card. Strip leading and trailing whitespace and any quotes.

mef The input file MEF pointer returned by mef_open. The header should have been read by now.

keyword The input string (case insensitive) keyword from which to return its value.

outstr The output string with the value of input keyword.

maxch Length in chars of **outstr**.

4. File Operations

4.1. mef_app_file (mefi, mefo)

Appends a FITS file to an output file. If the file does not exist, a dummy Primary Header Unit is first created.

mefi The input file MEF pointer returned by mef_open. The header should have been read by now.

mefo The output file MEF pointer returned by mef_open.

4.2. mef_copy_extn (mefi, mefo, group)

Copy a FITS extension given by its number 'group' into an output file. If the file does not exists, this extension becomes a Primary Header Data Unit of the output FITS file. If the output file already exists, the input extension gets appended.

mefi The input file MEF pointer returned by mef_open. The header should have been read by now.

mefo The output file MEF pointer returned by mef_open.

group The input extension number to be appended to the output file.

4.3. mef_dummyhdr (fd, hdr_fname)

Write a dummy Primary Header Unit with no data to a new file. Optionally a header file with user keywords can be used.

fd The output file descriptor.

hdr_fname The header filename. This is text file with a FITS header syntax that will be appended to the file. Each FITS card does not have to be 80 characters long. The routine takes care of the correct padding.