Astronomical Data Analysis Software and Systems VIII
ASP Conference Series, Vol. 172, 1999
D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds.

CFITSIO, v2.0: A New Full-Featured Data Interface

William Pence
HEASARC, NASA/Goddard Space Flight Center, Greenbelt, MD 20771

Abstract. Version 2.0 of the CFITSIO subroutine interface for reading and writing FITS format data files is now available from the HEASARC web site. Some of the major new features in this version are the ability to read FITS files over the network, support for reading IRAF format images, and options for filtering the input file at run time.

1. New CFITSIO Features

CFITSIO is a software library that contains both C and Fortran-callable interface routines for reading and writing data files in FITS format. It is fast and efficient and typically produces data throughput rates of 5 – 20 MB/s on modern workstations and PCs. Version 2.0 of the CFITSIO library is available at `http://heasarc.gsfc.nasa.gov/fitsio` and contains many new features:

- Y2000 compliant: supports the new `YYYY-MM-DDThh:mm:ss` FITS DATE keyword format.
- Any FITS file on the internet can be read by supplying the FTP or HTTP URL as the name of the file.
- IRAF format images (*.imh files) can be read. They are converted into FITS format on the fly before being opened by CFITSIO.
- FITS files can be read or written in memory or shared memory for better I/O performance than on magnetic disk.
- FITS files can be piped between tasks on the stdin and stdout streams, thus reducing the need to write temporary intermediate files on disk.
- Compressed FITS files (gzip or Unix compress formats) can be read.
- FITS Tables can be filtered or modified at run time by adding qualifiers to the file name using a general C-like syntax that supports arbitrarily complex expressions.
- Tables can be filtered at run time using Good-Time-Interval files (GTIs) and SAOimage region files. The application only sees the table rows that satisfy the filtering expression.
- Any single table column can be binned into a 1-D FITS image, and any pair of table columns can be binned into a 2-dimensional FITS image at run time. The application program then opens the FITS image, not the original table.
- A single FITS file can be opened multiple times and treated as independent files, e.g., the application program can read or write to different extensions in the FITS file simultaneously.

- New routines for dealing with large sets of hierarchically organized data files have been added.
- New FITS tables or images can be created using ASCII template files which describe the file structure.

2. Extended Filename Syntax

Support for different types of files and for filtering of the input file is provided through an extended filename syntax, as illustrated in the following examples:

- `myfile.fits`: opens the FITS file on disk in the current directory.
- `myfile.imh`: opens the IRAF format image file by converting it into a temporary FITS format image in memory (by default) which is then opened and passed to the application program.
- `myfile.fits.gz[events, 2]`: uncompresses the gzipped file myfile.fits into memory (by default) and moves to the extension which has the keywords `EXTNAME = 'EVENTS'` and `EXTVER = 2`.
- `-`: a dash (minus sign) signifies that the input file is to be read from the stdin file stream, or that the output file is to be written to stdout.
- `ftp://legacy.gsfc.nasa.gov/test/vela.fits`: FITS files in any FTP archive site on the internet can be opened with read-only access. Files with HTTP URLs can be opened in the same way.
- `shmem://h2[events]`: opens the FITS file in a shared memory segment.
- `mem://`: creates a scratch output file in computer memory. The resulting 'file' is deleted when the program exits, so this is mainly useful for testing purposes when a permanent copy of the output file is not needed.
- `myfile.fits[EVENTS][col Rad = sqrt(X**2 + Y**2)]`: creates a temporary file (in memory or on disk) that is identical to myfile.fits except it contains a new column in the EVENTS extension called `Rad` whose value is computed using the indicated expression which is a function of the values in the X and Y columns.
- `myfile.fits[EVENTS][PHA > 5]`: creates and opens a temporary FITS files that is identical to `myfile.fits` except that the EVENTS table will only contain the rows that have values of the PHA column greater than 5. In general, any arbitrary boolean expression using a C or Fortran-like syntax can be used to select rows from a table. GTI and regions files can also be use to select rows.
- `myfile.fits[EVENTS][bin (X,Y)=1,2048,4]`: creates a temporary FITS primary array image by binning (i.e., computing the 2-dimensional histogram) of the values in the X and Y columns of the EVENTS extension. In this case the X and Y coordinates range from 1 to 2048 and the image pixel size is 4 units in both dimensions, so the resulting image is 512 × 512 pixels in size.
- `myfile.fits[EVENTS][PHA  > 5][bin (X,Y)=1,2048,4]`: The column modifiers, row selection, and column histogramming filters can be combined into more complex operations. In this case the EVENTS extension is filtered for rows that have PHA greater than 5, then the X and Y columns are binned into an image.

Acknowledgments. Many people have contributed to the development of version 2.0 of CFITSIO. In particular, members of the Integral Science Data Center (Jurek Borkowski, Bruce O'Neel, and Don Jennings) designed the concept of plug-in I/O drivers now used by CFITSIO, and provided the drivers for accessing files in shared memory and via the FTP, HTTP, and ROOT protocols. Also, Don Jennings wrote the hierarchical grouping routines, and Jurek Borkowski wrote the ASCII template file parser.

Uwe Lammers, from the XMM project at ESTEC, provided the very fast expression parsing algorithm that is used by CFITSIO when calculating new column values or filtering table rows at run time. Peter Wilson, (NASA/GSFC, RSTX) combined this algorithm with CFITSIO's iterator function for even fast I/O throughput. Peter Wilson also wrote the Fortran-callable wrappers for CFITSIO.

And, last but not least, Jonathan McDowell and collaborators at the AXAF Science Center developed a file filtering and binning syntax which formed the basis for the CFITSIO extended filename syntax, and Doug Mink (Smithsonian Astrophysical Observatory) provided the routines for converting IRAF format images into FITS format.