

# Exploring Space, Time, and Data with WCSTools

Jessica Mink

Smithsonian Astrophysical Observatory

## Extending FITS Header Capabilities

Since the first paper describing the FITS (Flexible Image Transport System) in 1979, a FITS HDU (Header Data Unit), hereafter referred to as a "FITS header", has consisted of

*"one or more logical records giving header information in the form of 80-character ASCII card images, 36 per record. Each card image contains an 8-character keyword (in upper case), usually followed by an equals sign and a value...required keywords are followed, in any order, by optional keywords, some of which are described in the FITS papers, and are terminated by an END keyword."* (Greisen, 2002)

I wrote my first FITS reader in Fortran many years ago. While learning how to write compilers as an undergraduate, I became comfortable with parsing character strings, so I access a FITS header as a single character string, breaking out keywords and their values and comments by their position on each 80-character line and the "=" and "/" markers rather than the columns in which they occur. When C's built-in string search turned out to be much slower than that of the Fortrans I had been using, I wrote my own and use FITS headers as random access databases with 80-character loosely-formatted records. This turns out to provide a lot of flexibility. It also has until recently, made it possible to rapidly follow the original direction that *"required keywords are followed, in any order, by optional keywords"* [emphasis mine].

## Extending keywords beyond 8 characters

The 1981 paper, "FITS - a Flexible Image Transport System", by Wells, Greisen, and Harten, defined the standard rows for the components of a keyword line:

```
COLUMN          1111111111222222222233333333334
NUMBERS         1234567890123456789012345678901234567890
Card 1 SIMPLE =          T / comment
Card 2 BITPIX =          16 / comment
Card 3 NAXIS =           2 / comment
Card 4 NAXIS1 =         190 / comment
Card 5 NAXIS2 =         244 / comment
Card 6 END
```

### ESO Solution

Over time, it has been hard to give such 8-character keywords an adequately descriptive name, so ESO adopted a new keyword, HIERARCH, which indicates that the real keyword name which follows is of arbitrary length and that the entire line no longer follows the original alignment:

```
HIERARCH CAHA DET CCD5 = 1 / # of CCDs in detector array
HIERARCH CAHA DET ID = 'CCD15' / Detector System ID
HIERARCH CAHA DET EXP TYPE = 'science' / Type of exposure
HIERARCH CAHA DET CCD1 NAME = 'SITE#15' / Detector CCD name
HIERARCH CAHA DET CCD1 ID = '9094ABR05-01' / Detector CCD ID
HIERARCH CAHA DET CCD1 TEMP = -105.0 / dewar temperature in deg. C
HIERARCH CAHA DET CCD1 NX = 2048 / # of pixels along x
HIERARCH CAHA DET CCD1 NY = 2048 / # of pixels along y
```

### My Solution

Because the WCSTools FITS header reader already parses the line, and the HIERARCH option requires that the line be parsed using the "=" and "/" characters as field breaks anyway, why not simply allow arbitrary-length keywords:

OLD:

```
$ sethead testbin.fits longkeyword="This is a test"
$ imhead testbin.fits | tail -2
LONGKEYW= 'This is a test'
END
```

NEW:

```
$ imhead testbin.fits | tail -4
LONGKEYW= 'This is a test'
LONGKEYWORD = 'This is a test'
HIERARCH LONGKEYWORD = 'This is a test'
END
```

```
$ gethead testbin.fits longkeyw
This is a test
```

```
$ gethead -e testbin.fits longkeyword
This is a test
```

```
$ gethead -e testbin.fits "hierarch longkeyword"
This is a test
```

## Command line access to SIMBAD

```
$ simpos
SIMPOS 3.9.6, 28 August 2018, Jessica Mink (jmink@cfa.harvard.edu)
Return RA and Dec for object name using SIMBAD, NED, Vizier
Usage: simpos [-idtv][b|e|g] namel name2 ...
        simpos [-idtv][b|e|g] @namelist ...
name(n): Objects for which to search (space -> _)
namelist: File with one object name per line
-a: Print all information returned
-b: Print coordinates in B1950 instead of J2000
-c: Use CDS server in France, default is CFA in US
-d: Print coordinates in degrees instead of sexagesimal
-e: Print coordinates in ecliptic instead of J2000
-g: Print coordinates in galactic instead of J2000
-i: Print ID(s) returned from SIMBAD
-n: Find coordinates using the NED database
-s: Find coordinates using SIMBAD
-t: Print output as tab-separated table
-v: Print extra descriptive info
-z: Find coordinates in Vizier catalog

$ simpos aldebaran
04:35:55.239 +16:30:33.49 J2000
```

The WCSTools package, open-source software since 1996, is best known for its ability to set and utilize the world coordinate systems of image data and to search catalogs, but it contains other tasks to observe and manipulate the data and metadata in FITS files. Here are some of them, plus some upcoming additions to the package, including implementations of arbitrary-length keywords in FITS headers and the UCAC5, Atlas, SDSS, and GAIA catalogs. It's portable because it's all text!

## WCSTools Programs

(from the package command `wcstools`)

WCSTools 3.9.6 Programs

```
addpix: Add a constant value(s) to specified pixel(s)
bincat: Bin a catalog into a FITS image in flux or number
char2sp: Replace this character with spaces in output (default=_ )
conpix: Operate on all of the pixels of an image
cphead: Copy keyword values between FITS or IRAF images
crf: Change CR's to newlines in text file (for imwcs, imstar logs)
delhead: Delete specified keywords from FITS or IRAF image file headers
delwcs: Delete the WCS keywords from an image
filename: Drop directory from pathname, returning just the file name
fileroot: Drop filename from path name, returning directory path
fileroot: Drop file name extension, returning path name without it
edhead: Edit the header of a FITS or IRAF file
getcol: Extract specified fields from a space-separated ASCII table file
getdate: Convert between two date formats
getfits: Extract portion of a FITS file into a new FITS file, preserving WCS
gethead: Return values for keyword(s) specified after filename
getpix: Return value(s) of specified pixel(s)
gettab: Extract values from tab table data base files
httpget: Send contents returned from URL to standard output
i2f: Read two-dimensional IRAF image file and write FITS image file
imcat: List catalog sources in the area of the sky covered by an image.
imextract: Extract 1D file from 2D file or 2D file from 3D file
imfill: Replace bad pixels in image files with 2-D Gaussian, mean, or median
imhead: Print FITS or IRAF header
immatch: Match catalog and image stars using the WCS in the image file.
imrot: Rotate and/or reflect FITS or IRAF image files
imresize: Block sum or average a file by integral numbers of columns and rows
imsize: Print center and size of image using WCS keywords in header
imsmooth: Filter FITS and IRAF image files with 2-D Gaussian, mean, or median
imstack: Stack 1-dimensional images into a 2-dimensional image
imstar: Find and list stars in an IRAF or FITS image
imwcs: Match FITS or IRAF image stars to catalog stars and fit a WCS
isfits: Return 1 if argument is a FITS file, else 0
isnum: Return 1 if argument is an integer, 2 if it is floating point, else 0
isrange: Return 1 if argument is a range of the format n1[-n2[xs]]...
keyhead: Change keyword names in headers of FITS or IRAF images
newfits: Create blank FITS files (dataless by default with BITPIX=0)
remap: Rebin an image from its current WCS to a new one
scat: Search a source catalog given a region on the sky
sethead: Set header keyword values in FITS or IRAF images
setpix: Set specified pixel(s) to specified value(s)
simpos: Return RA and Dec for object name(s) from SIMBAD, NED, VizierR
sky2xy: Print image pixel coordinates for given sky coordinates
skycoor: Convert between J2000, B1950, galactic, and ecliptic coordinates
sp2char: Replaces space in string with specified character (default=_ )
subpix: Subtract a constant value(s) from specified pixel(s)
sumpix: Total pixel values in row, column, or specified area
wcshead: Print basic world coordinate system information for images
xy2sky: Print sky coordinates for given image pixel coordinates
* <program name> help lists possible arguments
* <program name> version lists version of program
```

## Extending keyword values beyond one line

WCSTools supports two methods for extending a keyword's entry beyond a single 80-character line:

### IRAF

This is not official or especially well documented, but is needed for WCSTools to support the IRAF TNX and ZPX polynomial image<->sky coordinate projections. It consists of a series of lines with the same keyword before an underscore and a sequence number afterward. Each stands alone as [keyword]=[value], but they only make sense when assembled. For example:

```
CTYPE1 = 'RA---TNX'
CTYPE2 = 'DEC---TNX'
...
WAT1_002= '44519483406535 -0.09397437792791556 0.1451290667953512 -4.3273459822'
WAT1_003= '97552E-5 4.525801441223172E-4 0.009912907609377613 0.185709640211661'
WAT1_004= '6 1.874028266281251E-5 0.01797451858969125 0.04459426127294047 0.010'
WAT1_005= '87988458562529 -0.1525431073290788 -0.00297118661606969 ''
WAT2_002= '044519483406535 -0.09397437792791556 0.1451290667953512 -4.808546205'
WAT2_003= '327283E-5 4.148865918405720E-4 0.01180811673900729 0.087140524550311'
WAT2_004= '24 -9.417303440113919E-4 0.03105863709218507 0.3049773562603359 0.01'
WAT2_005= '469707839868787 0.0440181026974359 0.1257086385879256 ''
```

### FITS Standard

Breaking the "in any order" rule, string values to be continued past 80 characters end with a "&" and are followed by a line starting with "CONTINUE". Keywords for world coordinate systems have not needed to take more than one line, so setting and reading this format is being implemented for the first time in the next version of WCSTools. For example:

```
COMMENT This FITS file may contain long string keyword values that are
COMMENT continued over multiple keywords. The HEASARC convention uses the &
COMMENT character at the end of each substring which is then continued
COMMENT on the next keyword which has the name CONTINUE.
DLSNAKE = '/home/ian/bin/dlsmake.new F2p11R, version $Revision: 1.6 $, invoked@'
CONTINUE ' Mon Feb 16 09:34:04 EST 2004'
```

## Updates to the WCSTools Package since 2013

(from the package `NEWS` file)

Version 3.9.6 (November 5, 2018)

```
getfits: Fix command line parsing so ranges work (2018-07-26)
webread.c: Fix bug that failed to reset buffer length reading table returned by
scat (2018-11-05)
ucacread.c: Correct bug to read UCAC 4 high proper motion stars correctly
using local path (2018-08-03)
gsc2read.c: Read GSC2 ID as extra tab-separated-table keyword (2018-08-07)
webread.c: Always use "\r\n" instead of "\n" when writing to or reading from
socket (2018-01-19)
```

Version 3.9.5 (March 30, 2017)

```
sethead: Fix bug so that coordinates with : and dates with - are strings
getcol: Set default maximum number of lines to process from 100000 to 500000
ty2read: Add fifth digit to star numbers because 256 are over 9999
```

Version 3.9.4 (August 2, 2016) (after Ole Streicher)

```
getdate: Fix input list so it works correctly (2016-08-02 for Armin Rest)
imstar: Append to output line rather than overwriting it
bincat: Fix mis-reading of coordinates from command line
cphead: Increase number of keywords to handle distortion polynomials
filext: If no file extension, print null string, avoiding seg fault
getfits: Decrement argument counter after reading center RA
imcat: Fix typo in UCAC output format
imstar: Fix sprint of headline
findstar.c: Fix bug in trim section parsing
fitsfile.c: Add one to header string length for trailing NULL
wcs.c: Clean up use of ptype so it is always 3 characters long
sortstar.c: Initialize haspm flag if zero
Update all man pages
```

Version 3.9.3 (June 23, 2016)

```
wcs.h, wcs.c: Increase ctype string length to 15 (+null=16) for distortion
wcslib.h, wcslib.c: Increase ctype string length to 15 (+null=16) for distortion
scat.c, ucacread.c: Typos in code fixed
matchstar.c, hget.c, fitsfile.c: Fix isnum() tests for added coloned times and
dashed date
```

Version 3.9.2 (December 10, 2015)

```
isnum: Add return of 4 for yyyy-mm-dd dates (2015-11-06)
isnum: Add -n option to output number without linefeed
imcat, imstar: Print two decimal places on PROS region file coordinates
remap: Declare GetFITSWCS()
simpos: Switch to CFA Vizier server as default (2015-05-05)
simpos: Use NED name resolver if called as nedpos or using -n argument
simpos: Look up in VIZIER catalogs if called as vizpos or using -z argument
simpos: Use CDS server if called as cdspos or using -c argument
sky2xy: Add xy option to -o to print only computed x and y coordinates
nedpos, nedname: Drop tasks because IPAC has dropped interface; use simpos
-n
getdate: Fix bug with FITS date conversion (2015-08-24)
fileroot: Add option to read file pathnames from stdin
catutil.c: Add tab as an assignment character in agets()
hget.c: Add return of 4 for yyyy-mm-dd dates to isnum() (2015-11-06)
```

Version 3.9.1 (March 24, 2015)

```
remap: Declare GetFITSWCS() as well as GetWCSFITS()
cphead: Fix bug freeing header string more than once
imcat, scat, immatch, imwcs: Fix bugs with web access to GSC2 and SDSS
catalogs
gsc2read.c: Drop concatenation of "blank" string in middle of search URL
webread.c: Fix bug in URL filtering
```

Version 3.9.0 (December 19, 2014)

```
gethead: Add -r option to return multiple values as a single word separated
by the first character of the keyword the value of which follows
(2014-12-19)
remap: Update usage to -f WCSfile suggested by Steve Willner (2014-02-19)
sethead: Fix bug dealing with setting the same keyword in multiple file
extensions
imcat, scat, immatch, imwcs: Implement UCAC4 catalog
fileutil.c: Add next_line() to return one line of file
fitsfile.c: fix buffer reallocation bug in fitsread()
tabread.c: fix file tests
gsc2read.c and tabread.c: adjust to GSC2.3.3 format
webbuff.c: Add filter to convert characters in web URLs
```

Version 3.8.9 (September 16, 2013)

```
imcat, imwcs, immatch, scat: Work around SDSS server failure
sdsread.c: Add alternate UK and US catalog servers
```

## Upcoming updates to WCSTools

It is my intention to write a single paper describing all of WCSTools over the next year or so. Despite being in development for almost 25 years, this has been something of a side project, and there is quite a bit to be done before all of the necessary parts of the package are ready to undergo refereeing. Here are some of my plans for the immediate future:

- Improved Catalog Support: Remote access to the GAIA, Atlas, most recent SDSS catalogs, hopefully via both SQL and VO interfaces
- Improved WCS fitting algorithm
- HEALPIX projection support
- Python wrapping of all of the tasks
- Your needs?



Jessica Mink  
jmink@cfa.harvard.edu  
<http://tdc-www.harvard.edu/mink/>  
@MinkJessica



WCSTools

Jessica Mink