FITS EventList format for ACTs

Karl Kosack

2010-08-17

Contents

1 ABOUT

1.1 INTRODUCTION

This document defines the parameters needed in a general atmospheric cherenkov telescope (ACT) array event-list file. An event list is a high-level data product that provides information about each shower detected by the array ("combined" parameters, like *energy* or *hadronness*), and may optionally contain results from individual telescopes (e.g. *impact parameter*). Event list files will be stored in FITS format, and template header/table definitions in the standard `CFITSIO` template format (defined in the `CFITSIO` documentation, chapter 11) will be generated from the description listed below.

Whenever possible, existing standards for column names and header keywords have been used, to ensure maximal compatibility with existing analysis tools (e.g. FTOOLS). Standards exist that are provided by the FITS standards committees and for example OGIP. These standards define how basic to complex data products should be stored in FITS format.

For those not familiar with the FITS format, the data files may contain any number of named **extensions**, each of which contains a **header** plus **data**. The *data* may be a **table** or an **n-dimensional array**.

In the case of a *table*, the columns of the table are described in a standard format in the *header* for that extension (along with any other header information for that table). For each column, one may define a **name**, a **type**, a **unit**, and optionally a **UCD** (universal descriptor defined by the VO standards committee describing the column)

An event-list data file may contain some or all of the following information:

1. observation header information
2. Event Shower Table (may have more than 1)
3. Array description table
4. Monte-Carlo information table
5. Monte-Carlo thrown energy distribution histogram

1.2 STANDARDS

Here are the standards documents which define columns marked as standard below.

- OGIP1 = OGIP standard for Event Lists
- HEASARC = http://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/docs/spectra/ogip_92_007/node
  http://heasarc.gsfc.nasa.gov/docs/journal/fits6.html
- OGIP-T = Angelini et al 1992 (time standards)
- HFWG.R3 = ftp://legacy.gsfc.nasa.gov/fits_info/ofwg_recomm/r3.txt
- HFWG.R7 = ftp://legacy.gsfc.nasa.gov/fits_info/ofwg_recomm/r7.txt

2 HEADERS

2.1 Observation Header

field	type	standard	comment / unit
CREATOR	string	HFWG.R7	‘progname v1.2.3’
TELESCOP	string	OGIP2	“CTA” (mission name)
EXTNAME	string	HEASARC	“events”
RUN_ID	int		observation run number or identifier
DATE_OBS	string	HEASARC	yy-mm-dd (user readable time)
TIME_OBS	string	HEASARC	hh:mm::ss
DATE_END	string	HEASARC	yy-mm-dd
TIME_END	string	HEASARC	hh:mm::ss
TSTART	double	HEASARC	mission time of start of obs(s)
TSTOP	double	HEASARC	mission time of end of obs(s)
MJDREFI	int	HEASARC	integer part of start MJD [s]
MJDREFF	double	HEASARC	fractional part of start MJD
TIMEUNIT	string	HEASARC	time unit of MJD [’days’]
TIMESYS	string	HEASARC	‘UTC’
TIMEREF	string	HEASARC	‘local’
TELAPSE	double	HFWG.R11	diff of start and end times
ONTIME	double	HFWG.R11	tot good time (incl deadtime)
LIVETIME	double	HFWG.R11	(deadtime=ONTIME/LIVETIME)
DEADC	double	HFWG.R11	deadtime fraction
TIMEDEL	double	HEASARC	time resolution (e.g. 1.0)
OBJECT	string	HEASARC	observed object (if applicable)
RA_OBJ	double	HFWG.R3	target position *1*
RA_OBJ	double	HFWG.R3	
RA_PNT	double	HFWG.R3	observation position
DEC_PNT	double	HFWG.R3	
ALT_PNT	double		average altitide of pointing
AZ_PNT	double		average azimuth of pointing
RADECSYS	string	HFWG.R3	‘fk5’
EQUINOX	real	HEASARC	(2000.0 for J2000)
CONV_DEP	real		convergence depth of telescopes
CONV_RA	real		convergence position RA [deg]
CONV_DEC	real		convergence position Dec [deg]
OBSERVER	string	HEASARC	proposer name?
N_TELS	int		
TELLIST	string		comma-separated list of tel IDs *1*
GEOLAT	double		latitude of array center [deg]
GEOLON	double		longitude of array center [deg]
ALTITUDE	double		altitude of array center [km]
EUNIT	string	HEASARC	energy unit ‘TeV’
EVTVER	string		event-list version number

Comments:

1 this list corresponds to the order of telescopes in the TELMASK column and the TELARRAY table

3 EVENTS TABLE

The EVENTS table (stored in an extension called *EVENTS*) is a binary table containing information for each triggered shower event. It does not contain detailed pixel-information for each telescope, but rather single reconstructed shower parameters or parameters that are calculated for each telescope.

The EVENTS table is intended to be a simple-to-work-eith, flat table that contains a base set of columns plus an number of optional columns that are specific to a particular analysis. Since there is only one set of shower-reconstruction parameters in the table, only one type of analysis should be included in each event-list, and separate lists generated for different analysis techniques.

3.1 Additional and optional parameters

Since the requirements for analysis of CTA data are not fully defined, this format must be extensible (adding more lower-level reconstruction parameters when needed). Generally all analyses need a gamma-hadron separation parameter, and generally there are severl such parameters (e.g. for Hillas-type, 3D model, 2D template, boosted decision tree, or any other type of reconstruction) For example, one might find that the timing information is useful in gamma-hadron separation. In that case, one may define a set of *shower timing parameters* columns that has one entry per event containing a "gammaness-from-timing" parameter that is calculated from the timing parameters of all telescopes in the lower-level analysis chain. This new parameter than can be then trivially used for cutting purposes.

The basic template for the event-list table can be extended by adding columns (using an `\included` template file) corresponding to the new parameters.

Column names for additional parameters should be prefixed by the type of analysis they correspond to (e.g. `MC\_` for Monte-carlo parameters, `HIL\_` for Hillas-style analysis parameters)

3.2 Telescope-wise parameters

Because some useful parameters, like the impact parameter of the shower, are different for each telescope in the array, it is necessary to define a method for storing these parameters. Here, the simplest method is chosen: telescope-wise columns contain an array of length N, where N is the number of telescopes participating in the observation.

The indexing of this array is linked to the list of telescopes in the TELLIST header keyword, or in the TELARRAY binary table (in a separate extension). The order of the telescope-ids listed in these places gives the order of elements in the telescope-wise arrays.

For example if the array consists of 4 telescopes with IDs 1,5,15,22 (e.g. a subset of a larger array), all entries in the `TEL\_IMPACT` column of the eventlist would be length 4 arrays, where the first element corresponds to telescope 1, the second to 5, and so on. If a telescope participating in the observation did not trigger for a given event, the value in it's array element is set to a nominal value (typically 0). Note that when the event-list is compressed (via e.g. gzip), most of the space lost using fixed-length arrays is regained.

The information about which telescopes triggered is stored in the TELMASK column of the eventlist, which is not an array, but a bitmask of length N, with the same telescope ordering. Using this bitmask, it is trivial to extract the values for triggered telescopes from the telescope-wise columns.

For example, using a vector-based language like Python (or e.g. IDL), the following can be used to extract the average impact parameter for telescope 15:

```
eventlist = pyfits.open("evfile.fits")['EVENTS']
impacts = eventlist.data.field("TEL_IMPACT")
mask = eventlist.data.field("TELMASK")

telindex = 2 # corresponding to telid 15 in this example
telimpact = impacts[telindex] # just the values for tel 15
telmask = mask[telindex]      # which of these are triggers for tel 15
avg = numpy.average( telimpact[telmask] )
```

3.3 EVENTS table details:

3.3.1 BASE SHOWER PARAMETERS

The base parameters should always be in every event-list file, regardless of what reconstruction technique produced the list. They contain temporal, spatial, energetic, and trigger information

field	type	standard	comment
EVENT_ID	uint		event number
TIME	double	OGIP1	timestamp of event, elapsed time (1)
TLIVE	double		timestamp of event (livetime so far)
MULTIP	short		multiplicity of tels used in recon *3*
TELMASK	bitmask		bit pattern of triggered tels
RA	real	OGIP1	reconstructed position RA
DEC	real	OGIP1	reconstructed position DEC
DIR_ERR	double		measure of error in position
DETX	double		tangential coord in nominal sys
DETY	double		tangential coord in nominal sys
ALT	double		event altitude *2*
AZ	double		event azimuth *2*
ALT_PNT	double		pointing altitude, for convenience
AZ_PNT	double		pointing azimuth, for convenience
COREX	double		position on ground (M)
COREY	double		position on ground (M)
CORE_ERR	double		error on core reconstruction (M)
XMAX	double		position of shower max (M)
XMAX_ERR	double		error on showermax
ENERGY	real	OGIP1	shower energy (TeV)
ENERGY_ERR	double		error on energy

Comments:

2 ALT and AZ can be stored here for simplicity, or you can let the user calculate them from the RA/DEC + TIME information…

3 In the OGIP memo, TIME is defined in "seconds" stored as a double. Is this an MJD? That would make the most sense, but may not be precise enough.

4 the question here is how much to split this up. A flat table is easier and faster, but multiple sub-tables are more flexible. What is shown above seems a fairly good balance between the two. The only parameter that may be redundant between each reconstuction type is the time (all other parameters are reconstruction-specific)

5 Of course may have more than one of these base shower parameter tables for each event list (one for each type of reconstruction!) So may need the extention name to be something containing a reconstruction type (SHOWER-HILLAS, SHOWER-M3D) or something...

6 need the RADECSYS and EQUINOX keywords in the header of this table

3.3.2 GAMMA-HADRON SEPARATION PARAMETERS

Since VHE gamma-ray data are dominated by backround events caused by cosmic ray (hadronic) induced air showers, no list of events is ever purely gamma-rays. Therefore it is necessary to have some sort of gamma-hadron separation parameter, on which cuts can be made to reduce the hadronic background. Since there are many techniques for doing this, and since these cuts can also be optimized for different energy ranges, it us useful to store one or more "hadronness" parameters in the event-list. This allows analyses optimized for multiple energy ranges and source strengths to be used with a single event list.

The simplest parametrization of an air-shower event is a moment-analysis of cleaned shower images (the resulting set of moments are known as the Hillas parameters [TODO:citation]). In a Hillas-parameter based analysis, the gamma-hadron separation parameter is usualy a combination of the *mean-reduced-scaled-width* and *mean-reduced-scaled-lenght* parameters (defined in e.g. [TODO: cite]).

The following gives examples of parameters that may be included in an event list for several types of gamma-hadron separation techniques (Hillas-style, 2D Model template, and 3D model). In each case, a prefix for the analysis type is appended, to avoid conflicting column names. Alternately, one could stipulate that all analyses provide a "HADRONNESS" value in a defined range.

- HILLAS PARAMETER COLUMNS

field	type	standard	comment
HIL_MSW	double		mean scaled width
HIL_MSL	double		mean scaled length
HIL_MSW_ERR	double		error on MSW
HIL_MSL_ERR	double		error on MSL

- MODEL PARAMETER COLUMNS

field	type	standard	comment
LIKELIHD	double		likelihood for being a gamma-ray
likelihoodErr	double		error on likelihood
...			

- Telescope-wise parameters
  As mentioned earlier, some parameters are specific to each telescope. For generating response matrices, for example, one needs the impact parameter of a shower with respect to each telescope. Although in principle this could be calucalted from the telescope location and shower reconstruction parameters, it is a relatively complex computation, involving a number of coordinate transformations. For this reason, it is easiest to have impact parameters pre-calculated and proved in the event-list.

field	type	standard	comment
TEL_IMPACT	double[N]		impact parameter of shower with each tel

For a particuar analysis (E.g. a Hillas-style analysis), one may also store other useful per-telescope parameters, such as the non-reduced Hillas parameters (LENGTH, WIDTH, SIZE, ASYMMETRY, etc). These can be used for reconstruction the shower's geometry or energy for example.

field	type	standard	comment
TEL_IMPACT	double[N]		impact parameter of shower with each tel

3.3.3 MONTE-CARLO SHOWER PARAMETERS

field	type	standard	comment
MC_EVENTID	uint		event number from simulation
MC_SHOWERID	uint		shower id from simulation
MC_PRIMID	uint		type of primary particle
MC_ENERGY	double		true energy
MC_ALT	double		true direction
MC_AZ	double		true direction
MC_XMAX	double		true showerMax [g/cm^2]
MC_COREX	double		true core X pos of shower axis
MC_COREY	double		true core Y pos of shower axis
MC_FIRSTINT	double		height of first interaction [m]
MC_XSTART	double		atmos. depth of first interaction [g/cm2]

Comments:

1. May also need simulation "combined" timing parameters here or in a separate table.

3.3.4 SHOWER TIMING PARAMETERS (TBD)

Timing parameters that are not telescope-specific (e.g. average-velocity? Who knows. It may be in the end just a "gammaness" parameter of how well the shower matches the timing characteristics of a hadon vs gamma)

field	type	standard	comment
EVENTID	uint		event number

4 ARRAY CONFIGURATION INFORMATION

4.1 TELESCOPE TABLE (one entry per telescope)

This is optional information (mostly needed by the low-level analysis), but is useful to include here (and doesn't take up much space). It can be used for example for visualization purposes or for identifying different array configurations in detail

field	type	standard	comment / unit
TELID	int		telescope number
TELCLASS	string		telescope type (HESS, CTA1,) *1*
TELPOSX	double		x pos rel to center of array (M)
TELPOSY	double		y pos rel to center of array (M)
TELPOSZ	double		z (height) of telescope (M)
TELFOV	double		fov in deg
TELMIRAREA	double		mirror area (m^2)
TELCAMAREA	double		camera area m^2
TELFNUM	double		F-number or focal length

5 MONTE-CARLO INFORMATION TABLES

5.1 MCINFO table

5.2 MCENERGY table

6 Implemetation notes

6.1 Storage of pointing information

6.1.1 Run-wise

6.1.2 Globally

6.2 Keyword names

in FITS, keyword names may only be 8 characters long, so this should be taken into account when defining this format in the template files.

6.2.1 Hierarchical keywords

The latest FITS standards support the usage of Hiarachical keywords (e.g. ARRAY.LOCATION.ALT). These could be used to simplify some of the header information

6.3 long strings in headers

Now supported by FITS and `CFITSIO` (see the `fits\\_*\\_key\\_longstr()` functions). The `CFITSIO` routines will automatically combine "continued" keywords into a single long string, overcomeing the 68-character limit for single key/values. They are stored in the FITS header as:

```
KEYWORD = 'this is a test of long strings. It can&'
CONTINUE= 'continue over multiple&'
CONTINUE= 'lines using the CONTINUE keyword'
```

6.4 Units

Units are defined for tables using the TUNITn keyword in the table definition, and for header values should be encoded in brackets as the first token of the comment string: e.g.

```
LAMBDA =                  5400.0 / [angstrom] this is the wavelength
```