

GammaLib - Change request #2339

Ignore additional table column(s) with differing shape in fits files instead of throwing exception

02/15/2018 10:49 AM - Specovius Andreas

Status:	Closed	Start date:	02/15/2018
Priority:	Normal	Due date:	
Assigned To:	Knödlseider Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.6.0		
Description			
Currently in ctools it is expected that fits files consist of columns that end with "_LO" and "_HI" and other table columns which all have to feature an exact same shape.			
It would be good if additional table columns (meaning ones which are not used within ctools but are present in the fits file) which feature a shape that differs from the shape of the other table columns would be skipped instead of throwing an exception.			
In simple words: allow/ skip extra table columns with differing shape			

History

#1 - 08/01/2018 10:50 PM - Knödlseider Jürgen

- Assigned To set to Knödlseider Jürgen
- Target version set to 1.6.0

#2 - 08/01/2018 10:50 PM - Knödlseider Jürgen

- Project changed from ctools to GammaLib
- Target version deleted (1.6.0)

#3 - 08/01/2018 10:50 PM - Knödlseider Jürgen

- Target version set to 1.6.0

#4 - 08/02/2018 09:27 AM - Knödlseider Jürgen

- Status changed from New to In Progress

Here is the current error message:

```
>>> import gammalib
>>> b=gammalib.GCTABackground3D('hess_bkg_example.fits')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "/usr/local/gamma/lib/python2.7/site-packages/gammalib/cta.py", line 4238, in __init__
      this = _cta.new_GCTABackground3D(*args)
ValueError: *** ERROR in GCTAResponseTable::read_tables(GFitsTable&): Invalid value. Table column "NEVENTS" contains 20 elements while 50000 elements were expected from the axis definitions. Please check the response table for consistency.
>>>
```

#5 - 08/02/2018 04:48 PM - Knödseder Jürgen

- Status changed from *In Progress* to *Pull request*

- % Done changed from 0 to 100

This was a tricky one since the exception is thrown at the level of the `GCTAResponseTable::read()` method which does not know anything about the file structure since this is a generic method. Removing the exception at that level remove the check for the file structure consistency. If for some reason the background matrix would be incompatible with the axis definitions, the class would simply ignore the matrix, and `GCTABackground3D` would pretend that the matrix was not found, although it exists in the file.

I therefore implemented a kludge at the level of the `GCTABackground3D::read()` where the expected file structure is known:

```
// Copy response table and skip all columns that are not mandatory. This
// kludge is needed to get rid of additional columns in the HESS
// background model.
GFitsTable *ptr = table.clone(); // Make sure that table is of same type
for (int i = 0; i < table.ncols(); ++i) {
    std::string colname = table[i].name();
    if ((colname != "DETX_LO") && (colname != "DETX_HI") &&
        (colname != "DETY_LO") && (colname != "DETY_HI") &&
        (colname != "ENERG_LO") && (colname != "ENERG_HI") &&
        (colname != "BKG") && (colname != "BGD")) {
        ptr->remove(colname);
    }
}

// Read background table from reduced FITS table
m_background.read(*ptr);

// Delete reduced FITS table
delete ptr;
```

The method clones the FITS table and removes all columns from the table that are not mandatory. The issue was that the `remove()` method did not yet exist, so I had to implemented a `GFitsTable::remove()` method that allows removal of a column from a table.

I also added a protected `GFitsTable::update_header()` method that makes sure that the header is up to date after any row or column operations. Before there was a risk for a discrepancy. Hence FITS column manipulation is now much cleaner.

#6 - 08/02/2018 04:50 PM - Knödseder Jürgen

It works now:

```
>>> import gammalib
>>> b=gammalib.GCTABackground3D('hess_bkg_example.fits')
>>> print(b)
=== GCTABackground3D ===
Filename .....: hess_bkg_example.fits
Number of DETX bins .....: 50
Number of DETY bins .....: 50
Number of energy bins .....: 20
DETX range .....: 64.8959468343793 - 72.2621799966754 deg
DETX range .....: -49.7521667480469 - -44.7521667480469 deg
Energy range .....: 0.1 - 100 TeV
>>>
```

#7 - 08/02/2018 07:58 PM - Knödseder Jürgen

- Status changed from Pull request to Closed

Code merged into devel.

#8 - 08/06/2018 09:39 AM - Specovius Andreas

user#3 wrote:

Code merged into devel.

Looks good! smile.png