

ctools - Feature #994

allow arbitrary energy binning in ctbin

11/19/2013 01:34 PM - Mayer Michael

Status:	Closed	Start date:	11/19/2013
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	00-07-00		
Description At the moment, ctbin automatically bins events logarithmically in energy. Already available in \$PFILES/ctbin.par, there is a parameter defining the energy binning algorithm: <i>ebinalg</i> . The two other possibilities "FILE" and "LIN" still have to be implemented. The only code change would be in line 489ff of ctbin.cpp, where the GEbounds object would be constructed according to the different options. Is there a format in which a energy binning file should be given?			

History

#1 - 11/19/2013 01:55 PM - Knödseder Jürgen

I was thinking about the Fermi/LAT format (gtbin in the Fermi/LAT science tools has the same parameter). I never looked in the type of format needed for Fermi/LAT. We can of course also support multiple formats, but I would start with the Fermi/LAT one.

#2 - 11/19/2013 03:27 PM - Knödseder Jürgen

- Target version set to 00-07-00

#3 - 11/19/2013 03:44 PM - Mayer Michael

I had a quick look how the binning is read in Fermi:
The user has to provide a simple ascii-file containing the ebounds, e.g.:

```
0.1 0.2
0.2. 0.5
0.5 20.0
```

This ascii-file has to be passed to gtbindef which creates a fits-file with exactly the same values in a FITS-bintable. This fits-file in turn, is passed to gtbin. For now, I cannot see the reason for this intermediate step. Probably since gtbindef also allows to make a time binning. I think as a first approach an ascii-file should be fine. I guess the easiest way to implement this is to extend the GEbounds::read method checking whether an ascii-file or a fits-file is present.

#4 - 11/19/2013 04:03 PM - Knödseder Jürgen

Interesting.

I'm not sure we want to dump code for the ASCII file in the GEbounds method as the ASCII file format apparently does not provide a mechanism for specifying the energy units. I think they added this intermediate step for the lazy people that do not want to edit FITS files.

So I would propose to go with the standard EBOUNDS format, hence we have everything in place that is needed for feeding these energy boundaries into ctbin.

#5 - 11/19/2013 04:39 PM - Mayer Michael

- File `gtbindef_output.fits` added
- Assigned To set to Mayer Michael

Ok, let's use FITS-files. I agree that we have to specify an energy unit. I attached the `gtbindef` output file. We probably should allow both extension names ("ENERGYBINS" and "EBOUNDS").
Furthermore, I guess we need to add the protected members `std::string m_ebinalg` and `std::string m_ebinfile` to `ctbin.hpp`

#6 - 11/19/2013 04:51 PM - Knödseder Jürgen

Thanks. Interestingly, the `gtbindef` executable does not fill the `CHANNEL` vector correctly. No idea if anyone is using this stuff ...

The `GEbounds` file constructor and load methods allow to specify the extension name, hence we can check whether one of `ENERGYBINS` or `EBOUNDS` exists. I think the `EBOUNDS` is more standard (at least for `HEASARC` and `XSPEC`), thus I would always write the energy boundaries out into an `EBOUNDS` extension. But we can be flexible for reading ...

I agree that `ctbin` needs to be extended. When I started writing the code I basically copied the `gtbin` interface to limit interface changes in the future, but much of the code is not yet there ...

#7 - 11/19/2013 05:36 PM - Mayer Michael

- % Done changed from 0 to 90

ok the code is almost ready. One questions though: We probably need to modify `$CTOOLS/syspfiles/ctbin.par` because right now `ebinalg` and `ebinfile` are hidden parameters - I guess both should be available to the user? (`ebinfile` only if `ebinalg=="FILE"`)

#8 - 11/19/2013 08:51 PM - Knödseder Jürgen

Mayer Michael wrote:

ok the code is almost ready. One questions though: We probably need to modify `$CTOOLS/syspfiles/ctbin.par` because right now `ebinalg` and `ebinfile` are hidden parameters - I guess both should be available to the user? (`ebinfile` only if `ebinalg=="FILE"`)

Agree that we should switch `ebinalg` and `ebinfile` to a.

#9 - 11/20/2013 10:33 AM - Mayer Michael

- Status changed from New to Pull request

ok, I changed `ebinalg` and `ebinfile` to a. Furthermore, I left two todo in the code. The first one is because I wasn't sure how to get the correct fitsio error status if the HDU `EBOUNDS` or `ENERGYBINS` isn't found in the FITS-file. The second is more a question: Do we want to have an additional check if `ebinalg` is none of `LOG/LIN/FILE` and throw an exception if none of them is present. Currently, if neither `LIN` nor `FILE` ist found, `LOG` is used instead. Since my last commit to `ctools` was a bit ago, I tried to push to the repository as stated on the `ctools-wiki`. However, this did not work because of the pushing permissions. Probably, we should link to the `gammalib-wiki` for how to contribute to `ctools`?

#10 - 11/20/2013 10:33 AM - Mayer Michael

- % Done changed from 90 to 100

#11 - 11/20/2013 11:30 AM - Knödlseider Jürgen

Mayer Michael wrote:

ok, I changed ebinalg and ebinfile to a. Furthermore, I left two todo in the code. The first one is because I wasn't sure how to get the correct fitsio error status if the HDU EBOUNDS or ENERGYBINS isn't found in the FITS-file.

The GFits class has two methods

```
bool    hashdu(const std::string& extname) const;
bool    hashdu(int extno) const;
```

for this purpose.

The second is more a question: Do we want to have an additional check if ebinalg is none of LOG/LIN/FILE and throw an exception if none of them is present. Currently, if neither LIN nor FILE ist found, LOG is used instead.

The parameter file class GPars/GPar should in principle allow to handle a predefined number of options, yet this is not implemented so far. I think your present logic is fine.

Since my last commit to ctools was a bit ago, I tried to push to the repository as stated on the ctools-wiki. However, this did not work because of the pushing permissions. Probably, we should link to the gammalib-wiki for how to contribute to ctools?

This depends where you want to push to. Normally, you should be able to push into a specific feature branch. Can tell me what branch you're trying to push?

#12 - 11/20/2013 11:43 AM - Mayer Michael

The GFits class has two methods

```
bool    hashdu(const std::string& extname) const;
bool    hashdu(int extno) const;
```

Yes, I am using `hashdu(const std::string& extname)` to check for the extensions. I implemented that an exception is thrown if neither "EBOUNDS" nor "ENERGYBINS" extension are found. For this I used `throw GException::fits_hdu_not_found(G_BIN_EVENTS, "\'EBOUNDS\' or \\'ENERGYBINS\'", status);`. My question was about the status which is passed to this exception. For now I set the status to zero which should be fine since the error message is meaningful already.

This depends where you want to push to. Normally, you should be able to push into a specific feature branch. Can tell me what branch you're trying to push?

I wanted to merge my branch into devel which is forbidden. On https://cta-redmine.irap.omp.eu/projects/ctools/wiki/Contributing_to_ctools however, the user is told to do so.

Therefore, like in `gammalib`, I created a feature branch `994-extend-ebinning-options-for-ctbin` and pushed it into the repo. This branch should be available and ready to get merged.

#13 - 11/20/2013 12:20 PM - Knödlseeder Jürgen

Mayer Michael wrote:

The `GFits` class has two methods
[...]

Yes, I am using `hashdu(const std::string& extname)` to check for the extensions. I implemented that an exception is thrown if neither "EBOUNDS" nor "ENERGYBINS" extension are found. For this I used `throw GException::fits_hdu_not_found(G_BIN_EVENTS, "\'EBOUNDS\' or \\'ENERGYBINS\'", status);`. My question was about the status which is passed to this exception. For now I set the status to zero which should be fine since the error message is meaningful already.

I'm in fact moving away from the "one class per exception" model, limiting the number of exceptions to a handful of standard ones (see <https://cta-redmine.irap.omp.eu/projects/gammalib/wiki/Exceptions>).

This depends where you want to push to. Normally, you should be able to push into a specific feature branch. Can tell me what branch you're trying to push?

I wanted to merge my branch into devel which is forbidden. On https://cta-redmine.irap.omp.eu/projects/ctools/wiki/Contributing_to_ctools however, the user is told to do so.

This is indeed outdated. Pushing into devel is forbinned.

Therefore, like in gammalib, I created a feature branch 994-extend-ebinning-options-for-ctbin and pushed it into the repo. This branch should be available and ready to get merged.

That's indeed the way to go!

#14 - 11/20/2013 01:08 PM - Knödlseeder Jürgen
- Status changed from Pull request to Closed

Merged into devel.

Files

gtbindef_output.fits	11.3 KB	11/19/2013	Mayer Michael
----------------------	---------	------------	---------------