

ctools - Feature #1476

ctbin should complain about missing Rol information

06/09/2015 01:59 PM - Mayer Michael

Status:	Closed	Start date:	06/09/2015
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.0.0		
Description			
<p>At the moment, when running ctbin on a raw event list, the resulting map will be empty. The reason is that the Rol information in the header keywords is missing because ctselect has to be run first to set them.</p> <p>In case a user tries to run ctbin on a raw event list, a warning should be thrown that ctbin requires Rol informations (which are set by ctselect). Currently, ctbin stays silent, assuming an empty Rol (Ra=0, Dec=0, rad=0), leading to an empty map which is quite confusing. A check should be included if the Rol is set correctly, otherwise a meaningful exception should be thrown that ctselect should be used prior to ctbin.</p>			

History

#1 - 06/09/2015 02:02 PM - Knödseder Jürgen

I agree that this needs to be changed.

It probably comes from the fact that in an earlier version, ctbin did not required ROI information.

Whoever wants to implement that, just take the issue and go ahead.

#2 - 06/09/2015 02:06 PM - Mayer Michael

- Assigned To set to Mayer Michael

There are two options to check if the Rol is empty:

1. Straight forward: Check if Ra=0 && DEC==0 && rad==0
2. Implement a is_empty function in GCTARol, which does the check

What would you prefer?

#3 - 06/09/2015 02:30 PM - Knödseder Jürgen

Good question.

Since an ROI with a radius of 0 is not making sense, so far the assumption was that an ROI with a radius of 0 is an empty ROI. So the logic would be to check simply on radius.

It has one pitfall, as in principle you can set the ROI radius to 0 in the header of an event file, hence ROI information would be formally being provided but not correctly recognized as such by the tools. This could be avoided by explicitly requiring that the ROI radius provided in an EVENTS file needs to be >0. Once could add such a test in the gammalib::read_ds_rol function in GCTASupport.cpp. Alternatively, one could enforce in the radius() that any argument passed needs to be positive.

I would then go with implementing a is_valid() method in GCTARol so that one could forget about what exactly is meant by having no valid ROI (is_valid() is better than is_empty(), as the later is for container classes signaling that no element exists in the contained class).

#4 - 06/09/2015 04:25 PM - Mayer Michael

- Status changed from New to Pull request

- % Done changed from 0 to 100

It has one pitfall, as in principle you can set the ROI radius to 0 in the header of an event file, hence ROI information would be formally being provided but not correctly recognized as such by the tools. This could be avoided by explicitly requiring that the ROI radius provided in an EVENTS file needs to be >0. One could add such a test in the `gammalib::read_ds_roi` function in `GCTASupport.cpp`. Alternatively, one could enforce in the `radius()` that any argument passed needs to be positive.

Actually, this requires some more modifications if we throw an exception if a non-positive value is passed to `GCTARoi::radius(const double&)`. The reason is that in `ctselect`, observation files get written and read in. Usually, the ROI will evolve in `ctselect` as follows:

- read original file without DS-keywords (-> empty ROI)
- save temporary file (including empty ROI -> radius=0.0 in file)
- load temporary file (would throw exception since ROI radius=0.0)
- Set ROI information from selection parameter (ROI is set properly)

To take care of that, we would have to modify `GCTAEventList::write_ds_keys()` to only write the ROI information if they are valid.

I did those things on two branches:

- `gammalib: 1472-add-is_valid-function-to-GCTARoi` (sry mixed up the issue number)
- `ctools: 1476-add-meaningful-exception-to-ctbin`

#5 - 06/10/2015 12:39 AM - Knödseder Jürgen

- Status changed from Pull request to Closed

- Target version set to 1.0.0

Both changes are merged into the respective devel branches.