

ctools - Feature #1579

Add csobs2caldb

11/17/2015 06:40 PM - Mayer Michael

Status:	Closed	Start date:	11/17/2015
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>To simplify simulations for current IACTs, I propose to create a new cscript called csobs2caldb. The script could be launched the following:</p> <p>Input observation definition file [obs.xml] Instrument (e.g. prod2) [MyCutConfig] Response name (e.g. South_50h) [Zenith75_Azimuth0]</p> <p>The script could take the first observation in the file obs.xml to create a caldb entry following cta_root2caldb.py. The mission string would be taken from the GObservation::instrument() function. Accordingly, the user could populate the path \$CALDB with own IRFs for simulation.</p> <p>One issue that needs to be fixed for simulations: currently ctobssim uses "cta" as default mission. We should add a hidden parameter mission which allows for the selection of the newly created missions in \$CALDB.</p>			

History

#1 - 11/17/2015 06:47 PM - Mayer Michael

Apparently the loading of the Calbb mission "cta" happens in GCTAResponseIrf. An alternative would be to create all caldb entries under the mission name "cta". The folder structure could then e.g. be: \$CALDB/data/cta/hess/bcf/zenith75_az0/
This would make the input parameter "instrument" obsolete, implying a very simple parameter interface.

#2 - 11/18/2015 10:59 AM - Mayer Michael

- Status changed from New to Pull request
- % Done changed from 0 to 100

I used the latter method (adding all IRFs under umbrella of CTA) leading to a simplified parameter interface. Code is available on [1579-add-csobs2caldb](#) and is very much inspired from cta_root2caldb.py.

#3 - 11/19/2015 09:51 AM - Knödseder Jürgen

In the long run (or even short run?) we would like to add the IRFs directly to the event files as the IRFs change quickly. We won't thus use CALDB anymore.

Note also that the CALDB structure is well defined:

http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/docs/memos/cal_gen_91_001/cal_gen_91_001.html.

Specifically, the path is

\$CALDB/data/<mission>/<instrument>/<class>

where < mission > is a lower-case string (without spaces) naming the mission (for example einstein, rosat, suzaku) and < instrument > is a lower case string (without spaces) which gives the name of the instrument, and < class > is either pcf, bcf or cpf. So \$CALDB/data/cta/hess/bcf/zenith75_az0/ is not a valid CALDB name.

Loading of the IRF happens indeed in GCTAResponseIrf using a hard wired cta:

```
// Set calibration database
GCaldb caldb;
if (gammalib::dir_exists(xml_caldb)) {
    caldb.rootdir(xml_caldb);
}
else {
    caldb.open("cta", xml_caldb);
}
this->caldb(caldb);
```

We could add an optional parameter mission (or maybe better observatory to be more explicit) that if not given is set to CTA. This should enable handling the following structure:

```
$CALDB/data/hess/bcf/zenith75_az0/
```

#4 - 11/19/2015 10:19 AM - Mayer Michael

Thanks for the feedback. I agree that bundling the IRFs to the events is definitely the way to go.

The use case for this feature is only for simulations using ctobssim. A user might want to simulate e.g. 50 hours of HESS observations of a certain detector configuration (zenith, azimuth, muon efficiency, ...) at a certain sky coordinate. How will this be accommodated if we don't use caldb anymore?

Currently the user has to provide an inobs XML file for non-CTA simulations. However, the observation times, pointing etc are already provided in this container. Therefore it is kind of a hack to modify these values in the event list's FITS header by hand.

So \$CALDB/data/cta/hess/bcf/zenith75_az0/ is not a valid CALDB name.

For the moment we have \$CALDB/data/cta/prod2/bcf/South_50h/, which looks quite similar to me (or am I missing something?). Instead of passing caldb=prod2 and irfs=South_50h to ctobssim, the user could pass e.g. caldb=hess and irfs=zenith75.

But I agree that this might not be ideal and as you say it might violate the caldb scheme. If we did it as you proposed above, we would need to add a hidden parameter observatory to ctobssim, right?

Would you propose to continue implementing this tool or what would be the plan for the use case I described above?

#5 - 11/19/2015 11:12 PM - Knödseder Jürgen

You are right, mistake is on my side. So this looks fine.

Also here we would also need:

- a unit test
- a reference manual page

Can you add this?

#6 - 11/20/2015 10:47 AM - Mayer Michael

- *Assigned To set to Mayer Michael*
- *% Done changed from 100 to 90*

Alright, i am working on this. I didn't want to write test cases and docs until we are sure we want to implement this feature smile.png

#7 - 11/20/2015 02:20 PM - Mayer Michael

- *% Done changed from 90 to 100*

I have added a .rst file and included a unit test for csobs2caldb. For the unit test, I added a specific observation XML file to \$CTOOLS/test/data/ since for the moment, the script only works on the latest CTA IRF format. Note that the subsequent unit test which runs cspull now uses the newly created caldb entry to test its validity.

I think we might be ready to merge.

#8 - 11/25/2015 07:22 AM - Knödseder Jürgen

- *Status changed from Pull request to Closed*

Merged into devel.