

ctools - Change request #1353

Review ctools parameter names

10/30/2014 04:25 PM - Knödlseider Jürgen

Status:	Closed	Start date:	10/30/2014
Priority:	High	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.0.0		
Description			
The ctools parameter names should be reviewed before going to release 1.0. Here a discussion thread to converge on a naming scheme: https://cta-redmine.irap.omp.eu/boards/14/topics/209 .			

History

#1 - 01/30/2015 10:30 PM - Knödlseider Jürgen

- *Assigned To* set to Mayer Michael
- *% Done* changed from 0 to 50

Michael, is this issue now basically done with your change? Or do we still to change some parameters that were not affected by the re-organisation of the base class?

#2 - 02/02/2015 11:42 AM - Mayer Michael

In general, all parameter names were changed following the proposal in the above link. Still, the cscripts have to be adapted as well (see #1408). I will take care of that.

#3 - 02/03/2015 01:46 AM - Knödlseider Jürgen

I recognized that ctselect now had caldb and irf as parameters. Not sure that this tool should have these parameters, it's just for event selection. Or was this added for save energy threshold selection? (the tool nevertheless seems to work without these parameters, hence I removed them for the moment).

#4 - 02/03/2015 09:57 AM - Mayer Michael

Yes, I guess these parameters are a relic from testing the energy threshold algorithms. Currently, the caldb and irf parameters are not queried at all. Therefore it is good to remove them. There is an exception thrown, however, if usethres="DEFAULT" and no response is given. Accordingly, the threshold algorithm does only work if a full observations xml-file is provided as inobs. I think it should be ok like this.

#5 - 02/03/2015 10:08 AM - Knödlseider Jürgen

Mayer Michael wrote:

Yes, I guess these parameters are a relic from testing the energy threshold algorithms. Currently, the caldb and irf parameters are not queried at all. Therefore it is good to remove them. There is an exception thrown, however, if usethres="DEFAULT" and no response is given. Accordingly, the threshold algorithm does only work if a full observations xml-file is provided as inobs. I think it should be ok like this.

I was thinking about this possibility. I'm just somewhat reluctant to ask for response parameters in a tool that just performs event selection. This somehow sounds odd. That's why my favorite option would be a specific tool that extracts energy thresholds from the response files and puts that into an XML file before running ctselect. I think it would provide a cleaner separation of things.

#6 - 02/03/2015 10:31 AM - Mayer Michael

- File *csxmlbuilder* added

I agree, this is good point to make the threshold application a separate tool.

Currently, I am working on a tool to ease the creation of the xml file (*csxmlbuilder*). It is written in python for simplicity but can of course be translated to a *ctool*.

Originally, it was intended for the H.E.S.S. collaboration since it uses an input ascii runlist which is usually the starting point of each analysis. The tool then queries for the location of the FITS data. It is still rather simplistic as it only allows one specific threshold determination (depending on a user-defined background model I studied in my thesis). It also directly creates the individual background models as output, which cannot be done by hand, e.g. if you have 200 runs. I attached the file, which is basically a follow-up from *obslist.py* (uploaded to issue #1039).

Do you see the need for such a tool?

#7 - 02/03/2015 11:04 AM - Knödlseeder Jürgen

This is definitely a useful tool.

At some point I was thinking about a GUI (probably based on Python's TkInter for interoperability) that allows manipulating observation definition XML files. This tool may then allow to scan databases and add additional information to the XML files. Observation selection could then be done by a mouse click (one could even think to select zones graphically from a plot showing the pointing directions of all observations).

Then, we certainly need a tool before *ctobssim* that allows to setup pointing sequences (something like *ctpntsim*). This includes taking into account realistic pointing patterns (you cannot point a region that at a given time is below the Earth horizon). Maybe this tool already gets the energy threshold from the response files, but maybe we want another intermediate tool for that so that it can be equally applied to simulated and real data (something like *ctengthres* or *ctethres*).

#8 - 02/03/2015 04:54 PM - Mayer Michael

This is definitely a useful tool.

Currently, it is a bit hand-made, with a lot of hard coded values. However, in the long run, there should be no ambiguity about the background model. It should come with the IRF as given. The code in *csxmlbuilder* is only tested and valid for HESS, but making this background model algorithm part of the IRF directly in the FITS exporters in the HESS Software seems a bit too soon to me. Therefore, I don't know if it is necessary to create a *ctool* out of *csxmlbuilder*. My feeling is to better to keep this specific to H.E.S.S. for the moment. A more generic tool which converts a runlist to an observation and background xml-file might be more useful (taking the background out of the IRF). What do you think?

At some point I was thinking about a GUI (probably based on Python's TkInter for interoperability) that allows manipulating observation definition XML files. This tool may then allow to scan databases and add additional information to the XML files. Observation selection could then be done by a mouse click (one could even think to select zones graphically from a plot showing the pointing directions of all observations).

Yes, that totally makes sense. For CTA, this should increase the usability a lot. If we want to implement this soon, we probably have to setup such databases for other IACTs?

Then, we certainly need a tool before *ctobssim* that allows to setup pointing sequences (something like *ctpntsim*). This includes taking into account realistic pointing patterns (you cannot point a region that at a given time is below the Earth horizon). Maybe this tool already gets the

energy threshold from the response files, but maybe we want another intermediate tool for that so that it can be equally applied to simulated and real data (something like `ctengthres` or `ctethres`).

I agree that this is necessary. Especially when we move towards pointing tables. I still like the idea to have the energy range definition in the observation xml-file, which then gets applied in `ctselect`. To have something like `ctethres`, there has to be a pre-defined algorithm to determine the energy threshold. Often the user has to test different approaches to account for systematic uncertainties.

#9 - 02/06/2015 02:25 PM - Knödlseider Jürgen

We may think about removing the `axisrot` parameter from the tools. This was initially for compliance with Fermi/LAT, but the parameters are not used and supported.

We probably should only keep parameters that are really used for release 1.0 as it's always easy to add parameters later (but more tricky to remove them).

#10 - 02/06/2015 03:14 PM - Mayer Michael

I fully agree to that. I was further thinking about the `edisp` parameter. I never tried what happens if it was set to true, but as far as I know energy dispersion computation in `ctools` is not mature enough, is it?

#11 - 02/06/2015 03:18 PM - Knödlseider Jürgen

`edisp` is in principle working. But this still needs testing.

There will be a student starting to work here at IRAP on the energy dispersion in about one week from now. So I think we should keep this parameter.

#12 - 02/06/2015 03:18 PM - Knödlseider Jürgen

- *Status changed from New to In Progress*

#13 - 02/06/2015 03:27 PM - Mayer Michael

`edisp` is in principle working. But this still needs testing. There will be a student starting to work here at IRAP on the energy dispersion in about one week from now. So I think we should keep this parameter.

Ok great, happy to keep it then.

#14 - 02/06/2015 05:18 PM - Knödlseider Jürgen

- *% Done changed from 50 to 70*

I removed now the axisrot parameter from all ctools.

Can we close this issue now or are there still some pending actions on the cscripts?

#15 - 02/06/2015 05:22 PM - Mayer Michael

I think we are fine on the ctools side. The cscripts still have to be adapted to use the new functions from the ctool class (#1408). Concerning the parameter names, however, I don't see any open issues left there.

#16 - 02/26/2015 08:36 AM - Knödlseider Jürgen

- % Done changed from 70 to 100

I went also over the cscript parameters and I think this is fine now. Can we close that?

#17 - 02/26/2015 09:22 AM - Mayer Michael

yes, looks good.

#18 - 04/10/2015 10:50 PM - Knödlseider Jürgen

- Status changed from In Progress to Closed

Closed.

Files

csxmlbuilder	9.18 KB	02/03/2015	Mayer Michael
--------------	---------	------------	---------------