

ctools - Feature #1807

Example script to visualise observation container

06/23/2016 05:03 PM - Mayer Michael

Status:	Closed	Start date:	06/30/2016
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.1.0		
Description			
<p>I have written an example script which allows to visualise an observation XML container. The script plots the zenith angle distribution, the azimuth distribution, the energy threshold, and the offset distribution from the target position.</p> <p>The script basically runs csobsinfo and demonstrates how to reuse some functions of this tool.</p> <p>If we want to have such a script in the software package, I will add some documentation as well.</p>			

History

#1 - 06/23/2016 05:06 PM - Mayer Michael

... I added the script on branch `1807-show_obs-example-script`.

#2 - 06/29/2016 12:48 AM - Knödlseher Jürgen

- Status changed from New to In Progress
- Assigned To set to Mayer Michael
- % Done changed from 0 to 50

I think this is useful. I did some commenting of the code on GitLab. Please note that we now have Python coding conventions (see <http://cta.irap.omp.eu/ctools-devel/coding/python.html>). For example, use from now on always ' hyphens, so that we can use the " hyphens in a string when needed, e.g. "This is the "parameter"".

I also added unit tests for all example scripts (see latest devel branch). They are exercised using make check. Please add also a unit test to the test_example.py.in script for show_obs.py. All matplotlib scripts should now have an optional argument that allows plotting in a file, which allows to execute the script in a unit test without having a window pop up.

About the "show" scripts: for now they have a different user interface than the cscripts. I'm still wondering whether we should do this in the long run (keep it for release 1.1 as is). Users will at some point get confused about different user interfaces, so I'm wondering whether we should also use the par interface for the scripts.

#3 - 06/29/2016 01:47 PM - Mayer Michael

- % Done changed from 50 to 100

I have implemented all the suggested changes. Thanks for the feedback.

I also added unit tests for all example scripts (see latest devel branch). They are exercised using make check. Please add also a unit test to the test_example.py.in script for show_obs.py. All matplotlib scripts should now have an optional argument that allows plotting in a file, which allows to execute the script in a unit test without having a window pop up.

Since the script creates four different plots, the script now looks for the argument "save" if the plots shall be saved to files or shown instead. The unit test of course requires them to be saved instead of shown.

About the "show" scripts: for now they have a different user interface than the cscripts. I'm still wondering whether we should do this in the long run (keep it for release 1.1 as is). Users will at some point get confused about different user interfaces, so I'm wondering whether we should also use the par interface for the scripts.

I agree this is somewhat irritating. However, I am not sure how far we want to go in supporting of plotting but user I/O via parfiles seems to be the obvious choice to go for.

#4 - 06/29/2016 01:48 PM - Mayer Michael

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

#5 - 06/30/2016 02:45 PM - Knödseder Jürgen

- Status changed from *Pull request* to *Closed*

- Target version set to *1.1.0*

- Start date set to *06/30/2016*

Merged into devel (refactored the code a bit and put all plots into a single figure)