

ctools - Feature #1408

Adapt cscripts to use ctool-base class functionality in get_parameters()

01/19/2015 01:43 PM - Mayer Michael

Status:	Closed	Start date:	01/19/2015
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.0.0		
Description After making the parameter interface to the ctools more homogenous (#1360), the same thing has to be done for all cscripts. Specifically we should use the functionality like setting up the the observations and other parameters from the ctool-base class in python			

History

#1 - 02/06/2015 03:25 PM - Mayer Michael

- Status changed from New to In Progress

I am trying to change the inheritance of the cscripts from gammalib.GApplication to ctools.ctool. However, calling the ctool.__init__() function in the cscript constructor raises an error:

```
AttributeError: No constructor defined - class is abstract
Exception TypeError: "in method 'GApplication_logTerse', argument 1 of type 'GApplication const *' in <bound method csresmap.__del__ of
<__main__.csresmap; >> ignored
```

I am a bit lost how to tackle that. Is it due to the definition of the ctool-class, or should it be called differently from python?

#2 - 02/06/2015 05:39 PM - Knödlseider Jürgen

- Assigned To set to Mayer Michael
- Target version set to 1.0.0
- % Done changed from 0 to 10

Can you push your current working branch into git so that I can have a look?

#3 - 02/06/2015 06:08 PM - Mayer Michael

I just started to edit csresmap when I realised the problem.
The branch is *1408-adapt-cscripts-to-ctool-base-class*

#4 - 02/07/2015 12:05 AM - Knödlseider Jürgen

- % Done changed from 10 to 50

This is indeed a tricky problem that I encountered already when I wrote the GTestSuite class.

My understanding of the problem is that Python does not know the concept of an abstract class, hence you cannot derive directly from ctool.i. The work around to this is to create a non-abstract class that derives from ctool.i and then to make a Python interface for this non-abstract class.

I did this by adding the cscript class to the ctool.i file. You will see that the cscript class exists twice in ctool.i. Once at the top within the { }; brackets

which is basically C++ code that is copy-pasted into the SWIG interface. Hence in this top section I create a C++ cscript class that implements dummy versions of the pure virtual methods that basically do nothing. In that way, cscript is a non-abstract class that shares all methods with ctool. The at the bottom there is the Python interface to this class that SWIG needs to create the corresponding Python classes.

Now check csresmap. You will see

```
class csresmap(ctools.cscript):
```

and later

```
# Initialise application
if len(argv) == 0:
    ctools.cscript.__init__(self, self.name, self.version)
elif len(argv) == 1:
    ctools.cscript.__init__(self, self.name, self.version, *argv)
else:
    raise TypeError("Invalid number of arguments given.")
```

That's all that is needed and it seems to work.

There was a conflict when I merged in the code which I corrected. Hence when you continue with adapting all cscripts please get a fresh checkout from devel.

I also added the

```
pars.append(gammalib.GApplicationPar("logfile","f","h","csresmap.log","","","Log filename"))
```

parameter similar to the ctools so that the logfile name can be changed.

#5 - 02/08/2015 02:32 PM - Mayer Michael

- % Done changed from 50 to 80

Excellent, thanks for your effort and support to correct this issue. csresmap now works without a problem, using the new functionality from the ctool-class.

Going over the individual cscripts, I realised that not every cscript would benefit by deriving from ctool. I am trying to comment on each of the cscripts in the following:

- cresmap: already done
- cscaldb: doesn't use any functionality to set up observations or maps. Therefore, it could stay as it is, inheriting from gammalib.GApplication?
- cspull, cstdist: They used different approaches to setup the observation container (all using obsutils functionality). I made them derived classes from cscript and slightly changed the parameter structure. Now the observation setup is more homogeneous like in ctool. I however kept the original philosophy by setting some parameters to hidden (e.g. "ra", "dec" in cstdist, or "inobs" in cspull). Furthermore I hid the parameter "tmin" (with 0.0 as default). At the same time, "duration" had to be changed to "tmax" to be conform with ctool nomenclature.
- cssens: I left this tool untouched because observations are setup in a for-loop with different energy ranges. Accordingly, using ctool-functionality here might complicate this tool in my view.

Generally, we have to decide whether all cscripts should derive from ctools.cscript or only if the functionality of the base class is used after all. I adapted the test cases according to the new parameters and pushed the changed to the working branch

#6 - 02/08/2015 03:24 PM - Knödlseider Jürgen

Thanks Michael. I would opt for deriving all scripts from cscript so that we could eventually adapt this class so that all cscripts fulfill some common interface or behavior. Some of the actual scripts are quite old, and it may be worth to see if it makes sense to rewrite them with the current new logic. This is however not an urgent issue.

#7 - 02/09/2015 10:50 AM - Mayer Michael

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

- % Done changed from 80 to 90

I would opt for deriving all scripts from cscript so that we could eventually adapt this class so that all cscripts fulfill some common interface or behavior.

Agreed, I adapted the code accordingly.

Some of the actual scripts are quite old, and it may be worth to see if it makes sense to rewrite them with the current new logic. This is however not an urgent issue.

I realised that when going through the code  . E.g. for cstdist, the ts computation functionality was not used so far. However, into the output file, some more values are written which would have been lost if we used the new algorithm (e.g. npred_nullhypothesis).

Apart from that everything should work fine now with the ctool.cscript inheritance.

#8 - 02/09/2015 11:09 AM - Knödseder Jürgen

- *Status changed from Pull request to Closed*

- *% Done changed from 90 to 100*

Merged into devel.