

ctools - Support #1885

GModelSpatialRadialProfile class parents changed in branch 1520-dm-profiles ?

11/24/2016 06:26 PM - Kelley-Hoskins Nathan

Status:	Feedback	Start date:	11/24/2016
Priority:	Normal	Due date:	
Assigned To:	Knödlseeder Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>I've been trying to get my old fork/branch up-to-date, and I think I've succeeded, but now I'm running into a compile problem, complaining about my class:</p> <pre>error: variable type 'const GModelSpatialRadialProfileDMZhao' is an abstract class</pre>			
<p>However, when I cloned a copy of gammalib/gammlib/1520-dm-profiles:</p> <pre>\$ git remote -v origin https://cta-gitlab.irap.omp.eu/gammlib/gammlib.git (fetch) origin https://cta-gitlab.irap.omp.eu/gammlib/gammlib.git (push) \$ git branch * (HEAD detached at origin/1520-dm-profiles) devel \$ git rev-parse HEAD d47536c29c905da1498e582ed2b02c149db4c6c6</pre>			
<p>and tried to compile it, I get a similar error for the example GModelSpatialRadialProfileGauss class:</p> <pre>\$ make -j 2 ... /bin/sh ../libtool --tag=CXX --mode=compile clang++ -DHAVE_CONFIG_H -I. -I../include -I../src -I../src -I/opt/local/include -g -O2 -c -o GModelSpatialRadialProfile.lo GModelSpatialRadialProfile.cpp /bin/sh ../libtool --tag=CXX --mode=compile clang++ -DHAVE_CONFIG_H -I. -I../include -I../src -I../src -I/opt/local/include -g -O2 -c -o GModelSpatialRadialProfileGauss.lo GModelSpatialRadialProfileGauss.cpp libtool: compile: clang++ -DHAVE_CONFIG_H -I. -I../include -I../src -I../src -I/opt/local/include -g -O2 -c GModelSpatialRadialProfile.cpp -fno-common -DPIC -o .libs/GModelSpatialRadialProfile.o libtool: compile: clang++ -DHAVE_CONFIG_H -I. -I../include -I../src -I../src -I/opt/local/include -g -O2 -c GModelSpatialRadialProfileGauss.cpp -fno-common -DPIC -o .libs/GModelSpatialRadialProfileGauss.o GModelSpatialRadialProfileGauss.cpp:41:39: error: variable type 'const GModelSpatialRadialProfileGauss' is an abstract class const GModelSpatialRadialProfileGauss g_radial_disk_seed; ^ ../include/GModelSpatialRadial.hpp:79:34: note: unimplemented pure virtual method 'region' in 'GModelSpatialRadialProfileGauss' virtual GSkyRegion* region(void) const = 0; ^ ../include/GModelSpatialRadial.hpp:69:34: note: unimplemented pure virtual method 'eval' in 'GModelSpatialRadialProfileGauss' virtual double eval(const double& theta, ^ GModelSpatialRadialProfileGauss.cpp:214:16: error: allocating an object of abstract class type 'GModelSpatialRadialProfileGauss' return new GModelSpatialRadialProfileGauss(*this); ^ 2 errors generated. make[3]: *** [GModelSpatialRadialProfileGauss.lo] Error 1 make[3]: *** Waiting for unfinished jobs... libtool: compile: clang++ -DHAVE_CONFIG_H -I. -I../include -I../src -I../src -I/opt/local/include -g -O2 -c GModelSpatialRadialProfile.cpp -o GModelSpatialRadialProfile.o >/dev/null 2>&1</pre>			

```
make[2]: *** [all-recursive] Error 1
make[1]: *** [all-recursive] Error 1
make: *** [all] Error 2
```

So I'm starting to think one of the GModelSpatialRadialProfile* parent classes changed, but these RadialProfile* classes never got updated. Does anyone know how to fix this?

-Nathan

History

#1 - 11/24/2016 08:31 PM - Knödseder Jürgen

The base classes indeed have changed in the meantime. Have you committed all code? If yes I can checkout your branch, fix it, and push it into the 1520-dm-profiles branch of gammalib.

#2 - 11/24/2016 11:09 PM - Kelley-Hoskins Nathan

Yes, my nkelhos/gammalib/1520-dm-profiles has all my code comitted. That'd be a great help, thanks.

#3 - 11/25/2016 09:53 AM - Knödseder Jürgen

- Status changed from New to Feedback
- Assigned To set to Knödseder Jürgen
- % Done changed from 0 to 100

I adapted the code and pushed it into gammalib/gammalib/1520-dm-profiles. You can pull it from there.

I've corrected the code a bit to follow the coding conventions (4 spaces for indentation, no extra spaces in function arguments or before ; etc.). Please try to follow the conventions.

#4 - 11/28/2016 05:33 PM - Kelley-Hoskins Nathan

I was trying to compile a merged nkelhos/gammalib/1520-dm-profiles , but I ran into problems. To try and trace them I tried to compile gammalib/gammalib/1520-dm-profiles, but I get an error:

```
$ ./autogen.sh && ./configure --prefix=$PWD && make clean && make -j 1 && make install
...
...
building '_model' extension
/usr/bin/clang -Wno-unused-result -Wsign-compare -Wunreachable-code -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-prototypes
-l/Users/nkelhos/Software/miniconda3/include -arch x86_64 -I../include -I/opt/local/include -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include
-l../inst/com/include -l/Users/nkelhos/Software/miniconda3/include/python3.5m -c gammalib/model_wrap.cpp -o
build/temp.macosx-10.6-x86_64-3.5/gammalib/model_wrap.o
gammalib/model_wrap.cpp:3907:43: error: no member named 'eval_gradients' in 'GModelSpatialRadial'
    return self->GModelSpatialRadial::eval_gradients(photon);
                                   ~~~~~^
1 error generated.
error: command '/usr/bin/clang' failed with exit status 1
make[3]: *** [build] Error 1
make[2]: *** [all] Error 2
make[1]: *** [all-recursive] Error 1
make: *** [all] Error 2
```

My setup:

```
$ git remote -v
origin https://cta-gitlab.irap.omp.eu/gammalib/gammalib.git (fetch)
origin https://cta-gitlab.irap.omp.eu/gammalib/gammalib.git (push)
```

```
$ git branch -vv
* 1520-dm-profiles 58af4ad [origin/1520-dm-profiles] Adapt DM radial profile code to new spatial model interface
  devel          c48c64a [origin/devel] Bugfix.
```

```
$ git pull
Already up-to-date.
```

#5 - 11/28/2016 10:10 PM - Knödseder Jürgen

You have probably still the old Python interface.

Try to type `make clean`, and then `./configure` and then `make clean` again, and then try to recompile.

I changed some time ago the logic for the Python interface to avoid that a user will delete the SWIG wrappers. Now, when you type `./configure` with the SWIG wrappers existing, they will never be deleted, even if the interface changes. There is an interface change with the new code, but since the old wrappers are never deleted, you have a conflict. I need to change this behavior for developers ...

#6 - 11/29/2016 01:53 AM - Kelley-Hoskins Nathan

I tried this with `gammalib/gammalib/1520-dm-profiles` where `make clean + configure + make clean + make` worked. But when I try it with my fork (`nkelhos/gammalib/devel`), it doesn't work, and I still get the same error.

```
/usr/bin/clang -Wno-unused-result -Wsign-compare -Wunreachable-code -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-prototypes
-I/Users/nkelhos/Software/miniconda3/include -arch x86_64 -I../include -I/opt/local/include -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include
-I../inst/com/include -I/Users/nkelhos/Software/miniconda3/include/python3.5m -c gammalib/linalg_wrap.cpp -o
build/temp.macosx-10.6-x86_64-3.5/gammalib/linalg_wrap.o
gammalib/linalg_wrap.cpp:3697:12: warning: unused function 'var_tuple_to_index' [-Wunused-function]
static int var_tuple_to_index(PyObject *input, int *ptr, int dim) {
      ^
1 warning generated.
/usr/bin/clang++ -bundle -undefined dynamic_lookup -L/Users/nkelhos/Software/miniconda3/lib -arch x86_64
build/temp.macosx-10.6-x86_64-3.5/gammalib/linalg_wrap.o -L../src/.libs -L/opt/local/lib -L/Users/nkelhos/Software/miniconda3/lib -L../src/.libs
-L/opt/local/lib -lgamma -lcfitsio -ledit -lcurses -o build/lib.macosx-10.6-x86_64-3.5/gammalib/_linalg.cpython-35m-darwin.so
-headerpad_max_install_names
building '_model' extension
/usr/bin/clang -Wno-unused-result -Wsign-compare -Wunreachable-code -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-prototypes
-I/Users/nkelhos/Software/miniconda3/include -arch x86_64 -I../include -I/opt/local/include -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include
-I../inst/com/include -I/Users/nkelhos/Software/miniconda3/include/python3.5m -c gammalib/model_wrap.cpp -o
build/temp.macosx-10.6-x86_64-3.5/gammalib/model_wrap.o
gammalib/model_wrap.cpp:3907:43: error: no member named 'eval_gradients' in 'GModelSpatialRadial'
    return self->GModelSpatialRadial::eval_gradients(photon);
           ~~~~~^
1 error generated.
error: command '/usr/bin/clang' failed with exit status 1
make[3]: *** [build] Error 1
make[2]: *** [all] Error 2
make[1]: *** [all-recursive] Error 1
make: *** [all] Error 2
```

But the merge command seems to think everything is up to date.

```
$ git branch -vv
* devel a8c7750 [origin/devel] merged NEWS for some conflic
```

```
$ git remote -v
origin https://cta-gitlab.irap.omp.eu/nkelhos/gammalib.git (fetch)
origin https://cta-gitlab.irap.omp.eu/nkelhos/gammalib.git (push)
```

```
upstream https://cta-gitlab.irap.omp.eu/gammalib/gammalib.git (fetch)
upstream https://cta-gitlab.irap.omp.eu/gammalib/gammalib.git (push)
```

```
$ git merge upstream/master
Already up-to-date.
```

#7 - 11/29/2016 09:08 AM - Knödseder Jürgen

That's because the spatial model interface changed (the `eval_gradient()` methods don't exist anymore; `gradient` is now a boolean flag), and your code is incompatible with this change.

You have to merge the `gammalib/gammalib/1520-dm-profiles` branch into your branch to make the changes also available on your side.

#8 - 11/29/2016 03:21 PM - Kelley-Hoskins Nathan

I think I've missed something, because when I do that merge, it says its up-to-date.

```
$ git branch -vv
* devel a8c7750 [origin/devel] merged NEWS for some conflic
$ git remote -v
origin https://cta-gitlab.irap.omp.eu/nkelhos/gammalib.git (fetch)
origin https://cta-gitlab.irap.omp.eu/nkelhos/gammalib.git (push)
upstream https://cta-gitlab.irap.omp.eu/gammalib/gammalib.git (fetch)
upstream https://cta-gitlab.irap.omp.eu/gammalib/gammalib.git (push)
$ git pull
Already up-to-date.
$ git merge remotes/upstream/1520-dm-profiles origin/master
Already up-to-date.
```

Is that the right way to merge it?

#9 - 11/29/2016 03:38 PM - Knödseder Jürgen

When I did the merge into `gammalib` I used the actual committed branch in your repo to start with, so in principle you do not have to do any merge on your side. Maybe rename your old `1520-dm-profiles` branch and checkout the `1520-dm-profiles` from `gammalib/gammalib` and the push this branch into your fork.

#10 - 11/29/2016 04:31 PM - Kelley-Hoskins Nathan

Ok, I think I did all of that, but I still get the same error.

- I renamed nkelhos/gammlib/1520-dm-profiles to 1520-dm-profiles-old
- deleted the existing 1520-dm-profiles (locally and with \$ git push origin --delete 1520-dm-profiles)
- checked out remotes/upstream/1520-dm-profiles (though this was only in HEAD mode)
- checked it out as a local branch (\$ git checkout -b 1520-dm-profiles)
- pushed this new branch to nkelhos/gammlib
- deleted the entire gammlib directory to start fresh
- cloned nkelhos/gammlib
- checked out origin/1520-dm-profiles

And when I tried autogen.sh + configure + make clean + configure + make clean + make, I still get the same error.

#11 - 12/01/2016 06:18 PM - Kelley-Hoskins Nathan

I started over with a fresh fork of gammlib/gammlib, and when I checkout 1520-dm-profiles and compiled it, I still got the same problem, even with make clean, configure, and make clean again.

```
building '_model' extension
/usr/bin/clang -Wno-unused-result -Wsign-compare -Wunreachable-code -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-prototypes
-l/Users/nkelhos/Software/miniconda3/include -arch x86_64 -I../include -I/opt/local/include -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include
-I../inst/com/include -l/Users/nkelhos/Software/miniconda3/include/python3.5m -c gammlib/model_wrap.cpp -o
build/temp.macosx-10.6-x86_64-3.5/gammlib/model_wrap.o
gammlib/model_wrap.cpp:3907:43: error: no member named 'eval_gradients' in 'GModelSpatialRadial'
    return self->GModelSpatialRadial::eval_gradients(photon);
                                   ~~~~~^
```

1 error generated.
error: command '/usr/bin/clang' failed with exit status 1

I can get it to go away if I comment out the eval_gradients extension in \$GAMMALIB/pyext/GModelSpatialRadialProfile.i , because (from what I understand) eval_gradients() is being phased out and replaced by eval().

\$GAMMALIB/pyext/GModelSpatialRadialProfile.i :

```
%extend GModelSpatialRadialProfile {
    double eval(const GPhoton& photon) const {
        return self->GModelSpatialRadial::eval(photon);
    }
    /*
    double eval_gradients(const GPhoton& photon) const {
        return self->GModelSpatialRadial::eval_gradients(photon);
    }
    */
};
```

I am not sure if this has other impacts, but I'm going to use it for now in my own branch of my new fork.

#12 - 12/01/2016 10:20 PM - Knödseder Jürgen

Sorry for that. Yes, this code should be removed. I somehow did not recognise that when I pushed the modifications.

In fact, the code should be changed as follows:

```
double eval(const GPhoton& photon, const bool& gradients) const {  
    return self->GModelSpatialRadial::eval(photon, gradients);  
}
```

i.e. a bool argument needs to be added.