GammaLib - Change request #616

How to debug python unit tests

12/05/2012 08:12 PM - Deil Christoph

Status:	Closed	**Start date:**	12/05/2012
Priority:	Normal	**Due date:**	
Assigned To:	Deil Christoph	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

I'm adding a unit test in test_GSky.py at the moment.
There are errors in my code that throw Python exceptions, but I'm having a hard time identifying them and getting my otherwise simple test to work, because all I see is this:

```
$ python test_python.py

...
Optimizer module dummy test: . ok
Test HEALPix map: E NOK
Test CAR projection map: ..... ok
...
```

The cause is that GTestSuite::run catches all exceptions.

You can reproduce my problem by e.g. inserting 1 / 0, which throws a ZeroDivisionError here:
https://cta-redmine.irap.omp.eu/projects/gammalib/repository/revisions/1985a10ddffafc19a29c03f82d3fd17822e70f31/entry/test/test_GSky.py#L128

There is some documentation on how to run and debug unit tests here, but as far as I can see the debugging info doesn't apply to tests written in Python?
https://cta-redmine.irap.omp.eu/projects/gammalib/wiki/Contributing_to_GammaLib

I can get my test working for now by inserting

```
import IPython; IPython.embed()
```

statements in my Python code, but it's tedious.

History

#1 - 12/05/2012 09:06 PM - Knödlseder Jürgen

The unit tests all create an XML file that is located in test/reports. Look for the GPython.xml file, it should contain the error messages that are thrown.

The XML files are nicely displayed in Jenkins. Maybe there exist other tools to display the content conveniently. The XML format is JUnit compatible (so I guess there should be tools around, maybe in Eclipse?)

#2 - 12/05/2012 09:15 PM - Deil Christoph

Jürgen Knödlseder wrote:

> The unit tests all create an XML file that is located in test/reports. Look for the GPython.xml file, it should contain the error messages that are thrown.
>
> The XML files are nicely displayed in Jenkins. Maybe there exist other tools to display the content conveniently. The XML format is JUnit compatible (so I guess there should be tools around, maybe in Eclipse?)

I just want the normal Python stack trace on my console.
I need this all the time in my edit / run cycle.

In reports/GPython.xml I see this:

```
  <testcase assertions="" classname="Python interface testing" name="Test FK5 to Galactic coordinate conversion" status="" time="0.371438">
    <error message="*** ERROR in GPythonTestSuite::test: <type 'exceptions.ZeroDivisionError'>
integer division or modulo by zero" type="N16GPythonException10test_errorE" />
  </testcase>
```

No traceback, no line numbers. :-(

#3 - 12/05/2012 09:22 PM - Knödlseder Jürgen

I know. I remember that I looked into this, but apparently it's not trivial to access the entire stack using the C API. Here is a relevant link:
http://stackoverflow.com/questions/1796510/accessing-a-python-traceback-from-the-c-api

You may open a feature about this, asking for the traceback to be added to the test suite output. In the meanwhile, the best thing is to test the Python script outside the test suite, so that you get things logged onto the screen. Once it's doing what you expect, you can embed it within the test suite.

#4 - 12/05/2012 09:49 PM - Deil Christoph

I think running the Python tests via your C test runner is not a good idea, it's too complicated.

Couldn't they simply be Python unit tests run with nose or py.test.
This is what all other mixed C / C++ / Python libraries do that I know of, e.g. numpy, scipy, ...

Over time gammalib will probably develop a huge Python unit test codebase, so it's important to set this up well.
Jürgen, if you would consider this I could set up a new forum thread to discuss this.

For now I'll add your comment from above on writing test code outside the unit test code and then copying it there in the end here:
https://cta-redmine.irap.omp.eu/projects/gammalib/wiki/Contributing_to_GammaLib
(gotta run to the bus, will do it tomorrow)

#5 - 12/05/2012 10:06 PM - Knödlseder Jürgen

The purpose of the test suite implementation is automatic unit testing. For this we don't need console output, but output in a standard test suite format. That was the entire purpose of implement the JUnit format in GammaLib. And it should run without any third party software, as everyone who

installs GammaLib should be able to run the unit tests. Therefore, it must be GammaLib native.

I understand that this is complicated for developing, but not all tests need to be unit tests.

#6 - 12/05/2012 11:53 PM - Deil Christoph

https://cta-redmine.irap.omp.eu/projects/gammalib/wiki/Contributing_to_GammaLib#Writing-and-debugging-Python-unit-tests

#7 - 12/05/2012 11:54 PM - Deil Christoph

- Status changed from New to Closed

- Assigned To set to Deil Christoph