

GammaLib - Bug #619


Segfault in @GPythonTestSuite.test_value()@

12/05/2012 11:32 PM - Deil Christoph

Status:	New	Start date:	12/05/2012
Priority:	Normal	Due date:	
Assigned To:	Knödseder Jürgen	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>I was playing with GPythonTestSuite on the command line to see how it works and what it can do, and got this segfault:</p>			
<pre>\$ cat test_GPythonTestSuite.py import gammalib test_suite = gammalib.GPythonTestSuite() test_suite.test_value(0, 0) \$ python test_GPythonTestSuite.py Segmentation fault: 11</pre>			
<p>Is this a bug or am I supposed to not call it with Python integers as input?</p>			
<pre>In [3]: test_suite.test_value? Type: instancemethod String Form:<bound method GPythonTestSuite.test_value of <gammalib.test.GPythonTestSuite; proxy of <Swig Object of type 'GPythonTestSuite **' at 0x10f3a74b0> >> File: /Users/deil/code/install/lib/python2.7/site-packages/gammalib/test.py Definition: test_suite.test_value(self, *args) Docstring: test_value(GTestSuite self, int const & value, int const & expected, std::string const & name="", std::string const & message="") test_value(GTestSuite self, int const & value, int const & expected, std::string const & name="") test_value(GTestSuite self, int const & value, int const & expected) test_value(GTestSuite self, double const & value, double const & expected, double const & eps=1.0e-10, std::string const & name="", std::string const & message="") test_value(GTestSuite self, double const & value, double const & expected, double const & eps=1.0e-10, std::string const & name="") test_value(GTestSuite self, double const & value, double const & expected, double const & eps=1.0e-10) test_value(GTestSuite self, double const & value, double const & expected)</pre>			

History

#1 - 12/06/2012 02:50 AM - Knödseder Jürgen

A segmentation fault is always a bug 

Yet, the class is not intended to be used directly, but it should serve as base class from which the test classes should be derived. Yet I had to make it non abstract.

I have to track down what exactly is happening when the class is used directly.

For the record, here is the crash report:


```
Exception Type: EXC_BAD_ACCESS (SIGSEGV)
Exception Codes: KERN_INVALID_ADDRESS at 0x0000000000000000
Crashed Thread: 0 Dispatch queue: com.apple.main-thread
```

Thread 0 Crashed: Dispatch queue: com.apple.main-thread

```
0 libstdc++.6.dylib      0x000000010029b8eb std::string::assign(std::string const&) + 11
1 libgamma.0.dylib      0x000000010111c83c GTestSuite::test_value(int const&, int const&, std::string const&, std::string const&) + 1068
2 _test.so              0x0000000101aec822 _wrap_GTestSuite_test_value + 2162
3 org.python.python     0x00000001000b7cb2 PyEval_EvalFrameEx + 22626
4 org.python.python     0x00000001000ba215 PyEval_EvalCodeEx + 2197
5 org.python.python     0x00000001000b81ad PyEval_EvalFrameEx + 23901
6 org.python.python     0x00000001000ba215 PyEval_EvalCodeEx + 2197
7 org.python.python     0x00000001000ba336 PyEval_EvalCode + 54
8 org.python.python     0x00000001000dec2c PyRun_InteractiveOneFlags + 380
9 org.python.python     0x00000001000dee8e PyRun_InteractiveLoopFlags + 78
10 org.python.python    0x00000001000df701 PyRun_AnyFileExFlags + 161
11 org.python.python    0x00000001000eef9d Py_Main + 2909
12 org.python.python    0x0000000100000f14 0x100000000 + 3860
```

#2 - 12/06/2012 09:43 AM - Deil Christoph

Jürgen Knödseder wrote:

A segmentation fault is always a bug 

I didn't know that. So swig always does type checking on inputs?

Apparently it is possible to create gammalib Python classes / methods that do crash.

So an `api-sanity-checker` on the Python side would be nice to catch all the trivial ways to make gammalib crash from Python.

I couldn't find such a tool, but Jürgen, if you agree that would be useful I'll make a ticket for myself and find or write something in the future.


#3 - 12/06/2012 09:44 AM - Deil Christoph

- Assigned To set to Knödseder Jürgen

#4 - 12/06/2012 12:54 PM - Knödseder Jürgen

Christoph Deil wrote:

Jürgen Knödseder wrote:

A segmentation fault is always a bug 

I didn't know that. So swig always does type checking on inputs?

Apparently it is possible to create gmmalib Python classes / methods that do crash.
So an `api-sanity-checker` on the Python side would be nice to catch all the trivial ways to make gmmalib crash from Python.
I couldn't find such a tool, but Jürgen, if you agree that would be useful I'll make a ticket for myself and find or write something in the future.

This sounds like a good idea. The Python interface is not yet very well tested, a lot of unit tests are still missing. As most of the functionality is tested using the C++ unit tests, I was thinking to focus mostly on the interface for the Python tests, but as you will see, little is done so far.

So a Python sanity checker would certainly help to improve the code quality.