GammaLib - Bug #489

GammaLib unit test fails on Mac OS 10.6 with gcc 4.3 & 4.4 compilers

09/16/2012 09:43 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	09/16/2012
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

The test suite fails on Mac OS 10.6 with gcc 4.3 & 4.4 compilers. Here the test result (excluding the Python tests that were not compiled in this example):

```
* GSupport *
************
Test Environment variable: ..... ok
PASS: test_GSupport

***********
* GVector *
***********
Define vectors: . ok
Test 1: Allocate zero vector: . ok
Test 2: Allocate too large vector: . ok
Test 3: Assign values: ... ok
Test 4: Define vector using copy constructor: .. ok
Test 5: Vector assignment: .... ok
Test 6: Assignment and arithmetics: ..... ok
Test 7: Comparison: ..... ok
PASS: test_GVector

*************************
* GMatrix class testing *
*************************
Test matrix allocation: .. ok
Test value assignment: ..... ok
Test matrix copying: ... ok
Test matrix operations: ..... ok
Test matrix arithmetics: ..... ok
Test matrix functions: ..... ok
Test matrix comparisons: ..... ok
Test matrix printing: .. ok
PASS: test_GMatrix

****************************
* GSymMatrix class testing *
****************************
Test matrix allocation: .. ok
Test value assignment: ..... ok
Test matrix copying: ... ok
Test matrix operations: ..... ok
Test matrix arithmetics: ..... ok
Test matrix functions: ..... ok
Test matrix comparisons: ..... ok
Test matrix Cholesky decomposition: ..... ok
Test matrix printing: .. ok
PASS: test_GSymMatrix

*******************************
* GSparseMatrix class testing *
*******************************
```

```
Test matrix allocation: ......... ok
Test value assignment: ..... ok
Test matrix copying: ... ok
Test matrix operations: ..... ok
Test matrix arithmetics: ..... ok
Test matrix functions: ..... ok
Test matrix comparisons: ..... ok
Test matrix Cholesky decomposition: ..... ok
Test matrix printing: ... ok
PASS: test_GSparseMatrix

*************
* GNumerics *
*************
Test GIntegral: . ok
Test Romberg integration: .... ok
PASS: test_GNumerics

***************
* FITS module *
***************
Test create: ........ ok
Test image byte: ..... ok
Test image ushort: ..... ok
Test image short: ..... ok
Test image ulong: ..... ok
Test image long: ..... ok
Test image longlong: ..... ok
Test image float: ..... ok
Test image double: ..... ok
Test bintable bit: ..... ok
Test bintable logical: ..... ok
Test bintable string: ..... ok
Test bintable double: ..... ok
Test bintable float: ..... ok
Test bintable ushort: ..... ok
Test bintable short: ..... ok
Test bintable ulong: ..... ok
Test bintable long: ..... ok
Test bintable longlong: ..... ok
PASS: test_GFits

**************
* XML module *
**************
Test XML attributes: ..... ok
Test XML elements: ..... ok
Test XML constructors: ..... ok
Test XML load: .... ok
Test XML access: ..... ok
PASS: test_GXml

****************
* GApplication *
****************
Test GLog constructor: . ok
Test stream logger: . ok
Test C logger: . ok
PASS: test_GApplication
GModelSpectralRegistry(const GModelSpatial*): add "PowerLaw" to registry.
GModelSpectralRegistry(const GModelSpatial*): "PowerLaw"

**********
* GModel *
**********
Test model parameter handling: . ok
Test model handling: ....... ok
```

```
Test models: .GModelSpectralRegistry(void): "PowerLaw"
.GModelSpectralRegistry(void): "PowerLaw"
GModelSpectralRegistry(void): "PowerLaw"
GModelSpectralRegistry(void): "PowerLaw"
.. ok
Test spectral model: GModelSpectralRegistry(void): "PowerLaw"
GModelSpectralRegistry(void): "PowerLaw"
EGModelSpectralRegistry(void): "PowerLaw"
GModelSpectralRegistry(void): "PowerLaw"
E. NOK
Test spacial model: EE. NOK
FAIL: test_GModel

********
* GSky *
********
Test GWcslib: . ok
Test Healpix GSkymap constructors: ......... ok
Test Healpix GSkymap I/O: .... ok
Test WCS GSkymap constructors: .EE. NOK
Test WCS GSkymap I/O: E NOK
FAIL: test_GSky

********************
* Optimizer module *
********************
Test unbinned optimization: ... ok
Test binned optimization: ... ok
PASS: test_GOptimizer

**********************
* Observation module *
**********************
Test unbinned optimization (1 thread): ... ok
Test unbinned optimization (10 threads): ... ok
Test binned optimization (1 thread): ... ok
Test binned optimisation (10 threads): ... ok
PASS: test_GObservation

*********************************************************
* Multi-wavelength instrument specific class testing *
*********************************************************
Test observation handling: ..... ok
Test optimizer: .GModelSpectralRegistry(void):
GModelSpectralRegistry(void):
E..GModelSpectralRegistry(void):
GModelSpectralRegistry(void):
E.. NOK
FAIL: test_MWL

*****************************************
* CTA instrument specific class testing *
*****************************************
Test response: .. ok
Test effective area: .. ok
Test PSF: ........ ok
Test integrated PSF: ..... ok
Test diffuse IRF: E NOK
Test diffuse IRF integration: E NOK
Test unbinned observations: ...... ok
Test binned observation: EE. NOK
Test unbinned optimizer: .GModelSpectralRegistry(void):
GModelSpectralRegistry(void):
E NOK
Test binned optimizer: EGModelSpectralRegistry(void):
GModelSpectralRegistry(void):
E NOK
```

```
FAIL: test_CTA

****************************************
* LAT instrument specific class testing *
****************************************
Test P6 response: ..... ok
Test P7 response: ..... ok
Test P6 livetime cube: ..... ok
Test P7 livetime cube: ..... ok
Test P6 unbinned observation: ..... ok
Test P7 unbinned observation: ..... ok
Test P6 binned observation: EE.E NOK
Test P7 binned observation: EE.E NOK
Test P6 binned optimizer: EE NOK
Test P7 binned optimizer: EE NOK
FAIL: test_LAT
============================================
5 of 16 tests failed
Please report to jurgen.knodlseder@irap.omp.eu
============================================
make[2]: *** [check-TESTS] Error 1
make[1]: *** [check-am] Error 2
make: *** [check-recursive] Error 1
```

In this example I enabled the debugging flag of GModelSpectralRegistry so that one sees which spectra are registered. Obviously, only the powerlaw spectrum is registered. The other spectral models did not register.

History

#1 - 09/17/2012 12:06 AM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 10

After improving the OpenMP configuration and correcting a bug in the compiler setting (see Bug #488), which is implemented in the release branch commit:07659949, the compilation using gcc 4.4 worked fine, but the unit test produced a number of errors. It is interesting that these errors are not the same as before. At a first glance, they look like uncatched exceptions.

```
************
* GSupport *
************
Test Environment variable: ..... ok
PASS: test_GSupport

***********
* GVector *
***********
Define vectors: . ok
Test 1: Allocate zero vector: . ok
Test 2: Allocate too large vector: . ok
Test 3: Assign values: ... ok
Test 4: Define vector using copy constructor: .. ok
Test 5: Vector assignment: .... ok
Test 6: Assignment and arithmetics: ..... ok
Test 7: Comparison: ..... ok
PASS: test_GVector

*************************
* GMatrix class testing *
*************************
Test matrix allocation: /bin/sh: line 1: 52104 Abort trap              DYLD_LIBRARY_PATH=../src/.libs:/opt/local/lib:/opt/local/lib/root:
 PYTHONPATH=../pyext:../inst/mwl/test:../inst/cta/test:../inst/lat/test:/Users/jurgen/dev/glast/python:/opt/local/lib/root: ${dir}$tst
FAIL: test_GMatrix

****************************
* GSymMatrix class testing *
```

```
*****************************
Test matrix allocation: /bin/sh: line 1: 52123 Abort trap              DYLD_LIBRARY_PATH=../src/.libs:/opt/local/lib:/opt/local/lib/root:
 PYTHONPATH=../pyext:../inst/mwl/test:../inst/cta/test:../inst/lat/test:/Users/jurgen/dev/glast/python:/opt/local/lib/root: ${dir}$tst
FAIL: test_GSymMatrix

*******************************
* GSparseMatrix class testing *
*******************************
Test matrix allocation: /bin/sh: line 1: 52141 Abort trap              DYLD_LIBRARY_PATH=../src/.libs:/opt/local/lib:/opt/local/lib/root:
 PYTHONPATH=../pyext:../inst/mwl/test:../inst/cta/test:../inst/lat/test:/Users/jurgen/dev/glast/python:/opt/local/lib/root: ${dir}$tst
FAIL: test_GSparseMatrix

*************
* GNumerics *
*************
Test GIntegral: . ok
Test Romberg integration: .... ok
PASS: test_GNumerics

***************
* FITS module *
***************
Test create: ........ ok
Test image byte: ..... ok
Test image ushort: ..... ok
Test image short: ..... ok
Test image ulong: ..... ok
Test image long: ..... ok
Test image longlong: ..... ok
Test image float: ..... ok
Test image double: ..... ok
Test bintable bit: ..... ok
Test bintable logical: ..... ok
Test bintable string: ..... ok
Test bintable double: ..... ok
Test bintable float: ..... ok
Test bintable ushort: ..... ok
Test bintable short: ..... ok
Test bintable ulong: ..... ok
Test bintable long: ..... ok
Test bintable longlong: ..... ok
PASS: test_GFits

**************
* XML module *
**************
Test XML attributes: ...../bin/sh: line 1: 52206 Abort trap              DYLD_LIBRARY_PATH=../src/.libs:/opt/local/lib:/opt/local/lib/root:
 PYTHONPATH=../pyext:../inst/mwl/test:../inst/cta/test:../inst/lat/test:/Users/jurgen/dev/glast/python:/opt/local/lib/root: ${dir}$tst
FAIL: test_GXml

****************
* GApplication *
****************
Test GLog constructor: . ok
Test stream logger: . ok
Test C logger: . ok
PASS: test_GApplication

**********
* GModel *
**********
Test model parameter handling: . ok
Test model handling: ..... ok
Test models: .... ok
Test spectral model: ... ok
Test spacial model: ... ok
PASS: test_GModel

********
* GSky *
********
Test GWcslib: ..... ok
Test Healpix GSkymap constructors: ..../bin/sh: line 1: 52256 Abort trap              DYLD_LIBRARY_PATH=../src/.libs:/opt/local/lib:/opt/local/lib/root:
 PYTHONPATH=../pyext:../inst/mwl/test:../inst/cta/test:../inst/lat/test:/Users/jurgen/dev/glast/python:/opt/local/lib/root: ${dir}$tst
FAIL: test_GSky
```

```
********************
* Optimizer module *
********************
Test unbinned optimization: ... ok
Test binned optimization: ... ok
PASS: test_GOptimizer

**********************
* Observation module *
**********************
Test unbinned optimization (1 thread): ... ok
Test unbinned optimization (10 threads): ... ok
Test binned optimization (1 thread): ... ok
Test binned optimisation (10 threads): ... ok
PASS: test_GObservation

*********************************************************
* Multi-wavelength instrument specific class testing *
*********************************************************
Test observation handling: ..... ok
Test optimizer: ..... ok
PASS: test_MWL

*****************************************
* CTA instrument specific class testing *
*****************************************
Test response: .. ok
Test effective area: .. ok
Test PSF: ........ ok
Test integrated PSF: ..... ok
Test diffuse IRF: .. ok
Test diffuse IRF integration: .. ok
Test unbinned observations: ...... ok
Test binned observation: ... ok
Test unbinned optimizer: ..... ok
Test binned optimizer: ..... ok
PASS: test_CTA

*****************************************
* LAT instrument specific class testing *
*****************************************
Test P6 response: ...... ok
Test P7 response: ...... ok
Test P6 livetime cube: ...... ok
Test P7 livetime cube: ...... ok
Test P6 unbinned observation: ...... ok
Test P7 unbinned observation: ...... ok
Test P6 binned observation: ..... ok
Test P7 binned observation: ..... ok
Test P6 binned optimizer: ..... ok
Test P7 binned optimizer: ..... ok
PASS: test_LAT

****************************
* Python interface testing *
****************************
Test GLog: ..... ok
Test GFits: ..... ok
Test GMatrix: ..... ok
Test GSymMatrix: ..... ok
Test GSparseMatrix: ..... ok
Model module dummy test: . ok
Numerics module dummy test: . ok
Observation module dummy test: . ok
Optimizer module dummy test: . ok
Test HEALPix map: ...../bin/sh: line 1: 52356 Abort trap
 DYLD_LIBRARY_PATH=../src/.libs:/opt/local/lib:/opt/local/lib/root:
 PYTHONPATH=../pyext:../inst/mwl/test:../inst/cta/test:../inst/lat/test:/Users/jurgen/dev/glast/python:/opt/local/lib/root: ${dir}$tst
FAIL: test_python.py
==============================================
6 of 17 tests failed
Please report to jurgen.knodlseder@irap.omp.eu
==============================================
make[2]: *** [check-TESTS] Error 1
```

```
make[1]: *** [check-am] Error 2
make: *** [check-recursive] Error 1
```

#2 - 09/17/2012 12:13 AM - Knödlseder Jürgen

- % Done changed from 10 to 20

Running the gdb debugger on test_GSky gave the following program stack:

```
Program received signal SIGABRT, Aborted.
0x00007fff897f40b6 in __kill ()
(gdb) where
#0  0x00007fff897f40b6 in __kill ()
#1  0x00007fff898949f6 in abort ()
#2  0x000000010080ab02 in uw_init_context_1 ()
#3  0x000000010080af48 in _Unwind_Resume ()
#4  0x000000010009bc67 in GWcs::coordsys (this=<value temporarily unavailable, due to optimizations>, coordsys=@0x1009066a0) at
 GWcs.cpp:213
#5  0x00000001000b160b in GWcsHPX (this=0x100906260, nside=@0x7fff5fbfef4c, order=@0x7fff5fbfe9c0, coords=@0x7fff5fbfe9d0) at
 GWcsHPX.cpp:113
#6  0x00000001000995b5 in GSkymap (this=0x7fff5fbfe400, wcs=@0x7fff5fbfe9e0, coords=@0x7fff5fbfe9d0, nside=<value temporarily unavailable,
 due to optimizations>, order=<value temporarily unavailable, due to optimizations>, nmaps=1) at GSkymap.cpp:144
#7  0x00000001000051d3 in std::string::_M_rep () at /opt/local/include/gcc44/c++/bits/basic_string.h:443
#8  0x00000001000051d3 in ~basic_string [inlined] () at /opt/local/include/gcc44/c++/bits/basic_string.h:503
#9  0x00000001000051d3 in TestGSky::test_GSkymap_healpix_construct (this=<value temporarily unavailable, due to optimizations>) at
 test_GSky.cpp:443
#10 0x000000010012b74c in GTestSuite::run (this=0x7fff5fbff260) at GTestSuite.cpp:294
#11 0x000000010012f82a in GTestSuites::run (this=0x7fff5fbff320) at GTestSuites.cpp:257
#12 0x00000001000039a0 in main () at test_GSky.cpp:716
```

The section of test_GSky.cpp where the program aborts is given below:

```
    // Test invalid coordsys in constructor
    test_try("Test invalid coordsys in constructor");
    try {
        GSkymap map("HPX", "HOR", 1, "RING", 1);
        test_try_failure();
    }
    catch (GException::wcs_bad_coords &e) {
        test_try_success();
    }
    catch (std::exception &e) {
        test_try_failure(e);
    }
```

The code aborts in line 4, which should lead to a exception in purpose.

The corresponding code throwing the exception can be found in GWcs.cpp:

```
void GWcs::coordsys(const std::string& coordsys)
{
    // Convert argument to upper case
    std::string ucoordsys = toupper(coordsys);

    // Set coordinate system
    if (ucoordsys == "EQU" || ucoordsys == "CEL" || ucoordsys == "C")
        m_coordsys = 0;
    else if (ucoordsys == "GAL" || ucoordsys == "G")
        m_coordsys = 1;
    else if (ucoordsys == "ECL" || ucoordsys == "E")
        m_coordsys = 2;
```

```
    else if (ucoordsys == "HEL" || ucoordsys == "H")
        m_coordsys = 3;
    else if (ucoordsys == "SGL" || ucoordsys == "S")
        m_coordsys = 4;
    else
        throw GException::wcs_bad_coords(G_COORDSYS_SET, coordsys);

    // Return
    return;
}
```

#3 - 09/17/2012 06:35 PM - Knödlseder Jürgen

After some investigations, I found that the problem arises when the exception is thrown by the GWcs class.

The following code works:

```
int main(void)
{
    try {
        throw GException::wcs_bad_coords("thrower", "Higgs");
        std::cout << "Should never be here." << std::endl;
    }
    catch (GException::wcs_bad_coords &e) {
        std::cout << "Catched with success." << std::endl;
    }
    catch (std::exception &e) {
        std::cout << "Catched something else." << std::endl;
    }
    catch (...) {
        std::cout << "What is going on?" << std::endl;
    }
    return 0;
}
```

The following code also works:

```
int main(void)
{
    try {
        thrower();
        std::cout << "Should never be here." << std::endl;
    }
    catch (GException::wcs_bad_coords &e) {
        std::cout << "Catched with success." << std::endl;
    }
    catch (std::exception &e) {
        std::cout << "Catched something else." << std::endl;
    }
    catch (...) {
        std::cout << "What is going on?" << std::endl;
    }
    return 0;
}
```

where

```
void thrower(void)
{
    std::cout << "Throw now." << std::endl;
    throw GException::wcs_bad_coords("thrower", "Higgs");
    std::cout << "Has thrown." << std::endl;
}
```

is defined in a shared library.

However, the following code does not work:

```
int main(void)
{
    try {
        GWcsHPX hpx;
        hpx.coordsys("Higgs");
        std::cout << "Should never be here." << std::endl;
    }
    catch (GException::wcs_bad_coords &e) {
        std::cout << "Catched with success." << std::endl;
    }
    catch (std::exception &e) {
        std::cout << "Catched something else." << std::endl;
    }
    catch (...) {
        std::cout << "What is going on?" << std::endl;
    }
    return 0;
}
```

#4 - 09/17/2012 09:07 PM - Knödlseder Jürgen

Finally, I found a case where a GammaLib exception is catched correctly. The following test program produces a successful catch:

```
#include <iostream>
#include "GammaLib.hpp"
int main(void)
{
    std::cout << "START" << std::endl;
    try {
        GCsv csv;
        std::string result = csv(10,-1);
        std::cout << "Should never be here." << std::endl;
    }
    catch (GException::out_of_range &e) {
        std::cout << "Catched with success." << std::endl;
    }
    catch (std::exception &e) {
        std::cout << "Catched something else." << std::endl;
    }
    catch (...) {
        std::cout << "What is going on?" << std::endl;
    }
    std::cout << "END" << std::endl;
    return 0;
}
```

The executable produces the following output:

```
START
ctor
Catched with success.
dtor
END
```

The interesting feature here is that the method that is throwing the exception (GCsv access operator) is located in the same module (support) as the exception handler base class. The exception that is thrown is located in the linalg module.

#5 - 09/17/2012 09:12 PM - Knödlseder Jürgen

Trying another example using the GMatrix class, analogous to the GCsv class, also worked. The same GException::out_of_range exception is thrown. The main difference here is that the GMatrix class is not implemented in the same module as the exception handler.

#6 - 09/17/2012 09:34 PM - Knödlseder Jürgen

The following code

```
#include <iostream>
#include "GammaLib.hpp"

int main(void)
{
    //
    std::cout << "START" << std::endl;
    try {
        GSkymap sky("Higgs", "Boson", 10, "found", 1);
        std::cout << "Should never be here." << std::endl;
    }
    catch (GException::wcs_invalid &e) {
        std::cout << "GException::wcs_invalid catched with success." << std::endl;
    }
    catch (std::exception &e) {
        std::cout << "Catched something else." << std::endl;
    }
    catch (...) {
        std::cout << "What is going on?" << std::endl;
    }
    std::cout << "END" << std::endl;

    //
    std::cout << "START" << std::endl;
    try {
        GWcsHPX hpx(37, "Higgs", "Boson");
        std::cout << "Should never be here." << std::endl;
    }
    catch (GException::wcs_hpx_bad_nside &e) {
        std::cout << "GException::wcs_hpx_bad_nside catched with success." << std::endl;
    }
    catch (std::exception &e) {
        std::cout << "Catched something else." << std::endl;
    }
    catch (...) {
        std::cout << "What is going on?" << std::endl;
    }
    std::cout << "END" << std::endl;

    //
    std::cout << "START" << std::endl;
    try {
        GWcsHPX hpx;
        hpx.coordsys("Higgs");
        std::cout << "Should never be here." << std::endl;
    }
    catch (GException::wcs_bad_coords &e) {
        std::cout << "Catched with success." << std::endl;
    }
    catch (std::exception &e) {
        std::cout << "Catched something else." << std::endl;
    }
    catch (...) {
        std::cout << "What is going on?" << std::endl;
    }
    std::cout << "END" << std::endl;

    return 0;
}
```

produces the following output

```
START
ctor
GException::wcs_invalid catched with success.
dtor
END
START
ctor
GException::wcs_hpx_bad_nside catched with success.
dtor
END
START
Going to throw exception.
ctor
Entered exception.
0x1009015a0
GWcs::coordsys(std::string)
Invalid WCS coordinate system (Higgs). Should be one of EQU/CEL/GAL/ECL/HEL/SGL.
Catched with success.
dtor
END
```

This contradicts what I have reported above. Something has changed ... we're closing in ...

#7 - 09/17/2012 10:31 PM - Knödlseder Jürgen

- % Done changed from 20 to 50

Okay, I found the issue. It was the line

export MACOSX_DEPLOYMENT_TARGET

in src/Makefile.am (a similar line was present in test/Makefile.am, but this apparently did not affect the code).

The export of the MACOSX_DEPLOYMENT_TARGET environment variable was introduced when handling universal binaries, and different SDKs. Some sections have been added to configure.ac, and these sections have been inspired from the configure.in script that is shipped with Python.

From http://developer.apple.com/library/mac/#technotes/tn2064/_index.html it follows that the MACOSX_DEPLOYMENT_TARGET environment variable influences the linker. In particular, from Mac OS X 10.2 weak linking is implemented, and I'm wondering whether this is not causing the problem here. Following http://developer.apple.com/library/mac/#technotes/tn2064/_index.html, if no MACOSX_DEPLOYMENT_TARGET environment variable is specified a value of 10.1 is assumed, hence no weak linking will be used.

I cannot be affirmative that weak linking is indeed the root cause, but removing the MACOSX_DEPLOYMENT_TARGET environment variable export helps.

I changed the configure.ac script so that MACOSX_DEPLOYMENT_TARGET is only exported if the --enable-universalsdk option is passed. I have now to check whether this causes any problems using the standard Apple compiler.

#8 - 09/18/2012 10:41 AM - Knödlseder Jürgen

- Status changed from In Progress to Feedback

- % Done changed from 50 to 100

I did some more checking, including also the compilation of ctools using the MacPorts gcc compiler.

This needed some additional minor adjustments. In particular, static linking during unit testing was now removed as this posed problems with the MacPorts gcc compiler. It also seems not needed actually by the Apple gcc compiler (I don't know why this is so as some time ago I introduced the static linking as the Apple gcc unit test failed; maybe some other changes in the meanwhile corrected the problems?).

I also profited from this change and removed the double compilation of the Python bindings for unit testing. This speeds up the test execution substantially.

The new GammaLib version has now been released. For the moment I still keep this bug on Feedback to see whether further problems arise in the next days during testing.

#9 - 12/01/2012 09:50 PM - Knödlseder Jürgen

- Status changed from Feedback to Closed