GammaLib - Change request #35

Rework exceptions

02/20/2012 05:39 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	01/18/2016
Priority:	High	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2.0.0		

Description

The actual GammaLib code has many exceptions, basically one per exception. This makes things difficult to manage. As the exceptions all have their associated error message, the exceptions can be reduced to some generic ones.

Remove any exceptions for the CTA exceptions class that are redundant with GammaLib GExceptions. CTA exceptions should only contain exceptions that are proper to the CTA classes and methods.

Related issues:

Related to ctools - Bug # 1007: ctskymap and ctbin give unhelpful error messa...	**In Progress**	**11/29/2013**

History

#1 - 12/21/2012 12:44 AM - Knödlseder Jürgen

- Target version set to 00-08-00

#2 - 02/19/2013 04:06 PM - Knödlseder Jürgen

- Subject changed from Rework CTA exceptions to Rework exceptions

- Description updated

- Status changed from New to In Progress

#3 - 02/19/2013 04:13 PM - Knödlseder Jürgen

Here a list of exceptions that are considered as generic:

Exception	Usage
feature_not_implemented	Is thrown when a feature is not implemented
invalid_argument	One of the function arguments is not valid
file_not_found	File not found
out_of_range	Index or value out of range

#4 - 12/10/2013 12:16 AM - Knödlseder Jürgen

- Target version deleted (00-08-00)

#5 - 01/11/2016 10:16 PM - Knödlseder Jürgen

- Start date deleted (02/20/2012)

- Release set to gammalib-1.1.0

#6 - 01/18/2016 03:24 PM - Knödlseder Jürgen

- Target version set to 1.1.0

- Start date set to 01/18/2016

#7 - 02/12/2016 12:43 AM - Knödlseder Jürgen

- % Done changed from 0 to 10

Removed the exceptions in the COMPTEL module.

#8 - 06/21/2016 05:23 PM - Knödlseder Jürgen

- Target version deleted (1.1.0)

#9 - 07/08/2019 11:00 AM - Knödlseder Jürgen

- Priority changed from Normal to High

- Target version set to 1.7.0

#10 - 07/09/2020 03:18 PM - Knödlseder Jürgen

- Assigned To set to Knödlseder Jürgen

- Target version changed from 1.7.0 to 2.0.0

Feature moved to next release.

#11 - 11/17/2020 01:54 PM - Knödlseder Jürgen

- Tracker changed from Feature to Action

#12 - 11/17/2020 01:54 PM - Knödlseder Jürgen

- Tracker changed from Action to Change request

#13 - 05/21/2021 04:06 AM - Knödlseder Jürgen

- % Done changed from 10 to 20

Replaced specific by generic exceptions in the following modules:

- MWL
- LAT
- CTA

All specific exceptions have been removed from the instrument modules.

#14 - 05/21/2021 06:39 PM - Knödlseder Jürgen

- % Done changed from 20 to 50

Replaced specific by generic exceptions in the following modules:

- app
- fits
- linalg
- model
- obs
- sky

#15 - 05/28/2021 02:51 AM - Knödlseder Jürgen

- Status changed from In Progress to Pull request

- % Done changed from 50 to 90

All specific exceptions were removed and replaced by generic exceptions. A number of help functions were implemented to reduce code duplications. The unit test scripts were also adapted to test now the generic instead of the specific exceptions.

#16 - 05/28/2021 07:44 AM - Knödlseder Jürgen

Testing the code integration, on Debian and and Free BSD the following unit test error occurs:

```
*** ERROR in GPythonTestSuite::test: <type 'exceptions.RuntimeError'> *** ERROR in GFits::saveto(GFilename&, bool&): Invalid value. Attempted to overwrite FITS file "test_python_skymap_hpx_v2.fits". Please set clobber flag to true.
```

Not sure what makes these systems different. Debian has Python 2.6, Free BSD has Python 2.7. Debian has gcc 4.4.5, Free BSD has gcc 4.2.1.

The relevant Python code before the change was

```
    # Save HEALPix skymap twice. The second saving should fail.
    try:
        pixels.save(file2, True)
        pixels.save(file2)
    except RuntimeError:
        pass
    else:
        raise RuntimeError('*** TEST ERROR: FITS file overwritten!')
```

and after the change it is

```
    # Save HEALPix skymap twice. The second saving should fail.
    try:
        pixels.save(file2, True)
        pixels.save(file2)
    except ValueError:
        pass
    else:
        raise RuntimeError('*** TEST ERROR: FITS file overwritten!')
```

This explains why this error did not occur before.

#17 - 05/28/2021 07:49 AM - Knödlseder Jürgen

I'm wondering whether this has to do this the RTLD_GLOBAL flag mentioned here:
https://stackoverflow.com/questions/7121631/uncatchable-c-exceptions-shared-libs-arm-linux-gnueabi-g.

#18 - 05/28/2021 07:59 AM - Knödlseder Jürgen

Here is on Debian the compilation and link step for the sky module:

```
gcc -pthread -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototypes -fPIC -I../include -I../inst/mwl/include -I../inst/cta/include
 -I../inst/lat/include -I../inst/com/include -I../inst/spi/include -I/usr/include/python2.6 -c gammalib/sky_wrap.cpp -o
 build/temp.linux-x86_64-2.6/gammalib/sky_wrap.o -fopenmp
g++ -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions build/temp.linux-x86_64-2.6/gammalib/sky_wrap.o -L../src/.libs -Wl,-R../src/.libs -lgamma
 -lcfitsio -o build/lib.linux-x86_64-2.6/gammalib/_sky.so -fopenmp
```

here for Free BSD

```
cc -fno-strict-aliasing -O2 -pipe -fno-strict-aliasing -DNDEBUG -O2 -pipe -fno-strict-aliasing -fPIC -I../include -I/usr/local/include -I../inst/mwl/include
 -I../inst/cta/include -I../inst/lat/include -I../inst/com/include -I../inst/spi/include -I/usr/local/include/python2.7 -c gammalib/sky_wrap.cpp -o
 build/temp.freebsd-9.0-RELEASE-amd64-2.7/gammalib/sky_wrap.o -fopenmp
c++ -shared -pthread build/temp.freebsd-9.0-RELEASE-amd64-2.7/gammalib/sky_wrap.o -L../src/.libs -L/usr/local/lib -R../src/.libs -R/usr/local/lib
 -lgamma -lcfitsio -lreadline -lncurses -o build/lib.freebsd-9.0-RELEASE-amd64-2.7/gammalib/_sky.so -fopenmp
```

and here the equivalent on CentOS 6:

```
gcc -pthread -fno-strict-aliasing -DNDEBUG -O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector
 --param=ssp-buffer-size=4 -m64 -mtune=generic -D_GNU_SOURCE -fPIC -fwrapv -fPIC -I../include -I/usr/include/cfitsio -I../inst/mwl/include
 -I../inst/cta/include -I../inst/lat/include -I../inst/com/include -I../inst/spi/include -I/usr/include/python2.6 -c gammalib/sky_wrap.cpp -o
 build/temp.linux-x86_64-2.6/gammalib/sky_wrap.o -fopenmp
g++ -pthread -shared build/temp.linux-x86_64-2.6/gammalib/sky_wrap.o -L../src/.libs -L/usr/lib64 -Wl,-R../src/.libs -lgamma -lcfitsio -lreadline -lncurses
 -lpython2.6 -o build/lib.linux-x86_64-2.6/gammalib/_sky.so -fopenmp
```

on CentOS 7:

```
gcc -pthread -fno-strict-aliasing -O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4
 -grecord-gcc-switches -m64 -mtune=generic -D_GNU_SOURCE -fPIC -fwrapv -DNDEBUG -O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2
 -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -D_GNU_SOURCE -fPIC -fwrapv -fPIC
 -I../include -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include -I../inst/com/include -I../inst/spi/include -I/usr/include/python2.7 -c
 gammalib/sky_wrap.cpp -o build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o -fopenmp
g++ -pthread -shared -Wl,-z,relro build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o -L../src/.libs -L/usr/lib64 -Wl,-R../src/.libs -lgamma -lcfitsio
 -lreadline -lncurses -lpython2.7 -o build/lib.linux-x86_64-2.7/gammalib/_sky.so -fopenmp
```

on Fedora 17:

```
gcc -pthread -fno-strict-aliasing -O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector --param=ssp-buffer-size=4 -m64
 -mtune=generic -D_GNU_SOURCE -fPIC -fwrapv -DNDEBUG -O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector
 --param=ssp-buffer-size=4 -m64 -mtune=generic -D_GNU_SOURCE -fPIC -fwrapv -fPIC -I../include -I/usr/include/cfitsio -I../inst/mwl/include
 -I../inst/cta/include -I../inst/lat/include -I../inst/com/include -I../inst/spi/include -I/usr/include/python2.7 -c gammalib/sky_wrap.cpp -o
 build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o -fopenmp
g++ -pthread -shared -Wl,-z,relro build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o -L../src/.libs -L. -Wl,-R../src/.libs -lgamma -lcfitsio -lreadline
 -lncurses -lpython2.7 -o build/lib.linux-x86_64-2.7/gammalib/_sky.so -fopenmp
```

on Mandriva:

```
gcc -pthread -fno-strict-aliasing -O2 -g -frecord-gcc-switches -Wstrict-aliasing=2 -pipe -Wformat -Werror=format-security
 -Wp,-D_FORTIFY_SOURCE=2 -fstack-protector --param=ssp-buffer-size=4 -fPIC -DNDEBUG -O2 -g -frecord-gcc-switches -Wstrict-aliasing=2 -pipe
 -Wformat -Werror=format-security -Wp,-D_FORTIFY_SOURCE=2 -fstack-protector --param=ssp-buffer-size=4 -fPIC -g -fPIC -I../include
 -I/usr/include/cfitsio -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include -I../inst/com/include -I../inst/spi/include -I/usr/include/python2.7 -c
 gammalib/sky_wrap.cpp -o build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o -fopenmp
g++ -pthread -shared -Wl,--as-needed -Wl,--no-undefined -Wl,-z,relro -Wl,-O1 -Wl,--build-id -Wl,--enable-new-dtags
 build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o -L../src/.libs -L/usr/lib64 -Wl,-R../src/.libs -lgamma -lcfitsio -lreadline -lncurses -lpython2.7 -o
 build/lib.linux-x86_64-2.7/gammalib/_sky.so -fopenmp
```

on OpenSolaris:

```
/usr/lib/python2.6/pycc -DNDEBUG -KPIC -I../include -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include -I../inst/com/include -I../inst/spi/include
 -I/usr/include/python2.6 -c gammalib/sky_wrap.cpp -o build/temp.solaris-2.11-i86pc-2.6/gammalib/sky_wrap.o
gcc: unrecognized option `-KPIC'
/usr/lib/python2.6/pyCC -G build/temp.solaris-2.11-i86pc-2.6/gammalib/sky_wrap.o -L../src/.libs -L/usr/lib -R../src/.libs -lgamma -lcfitsio -lpython2.6 -o
 build/lib.solaris-2.11-i86pc-2.6/gammalib/_sky.so
```

on OpenSuse:

```
gcc -pthread -fno-strict-aliasing -g -O2 -DNDEBUG -fmessage-length=0 -O2 -Wall -D_FORTIFY_SOURCE=2 -fstack-protector -funwind-tables
 -fasynchronous-unwind-tables -g -fPIC -I../include -I/usr/include/libcfitsio0 -I../inst/mwl/include -I../inst/cta/include -I../inst/lat/include
 -I../inst/com/include -I../inst/spi/include -I/usr/include/python2.7 -c gammalib/sky_wrap.cpp -o build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o
 -fopenmp
g++ -pthread -shared build/temp.linux-x86_64-2.7/gammalib/sky_wrap.o -L../src/.libs -L/usr/lib64 -Wl,-R../src/.libs -lgamma -lcfitsio -lreadline -lncurses
 -lpython2.7 -o build/lib.linux-x86_64-2.7/gammalib/_sky.so -fopenmp
```

#19 - 05/28/2021 08:40 AM - Knödlseder Jürgen

Debugging the code on Debian, it appears that in the block

```
  try {
    ((GSkyMap const *)arg1)->save((GFilename const &)*arg2);
  }
  catch (const GException::out_of_range& e) {
    SWIG_exception(SWIG_IndexError, e.what());
  }
  catch (const GException::invalid_value& e) {
    SWIG_exception(SWIG_ValueError, e.what());
  }
  catch (const GException::invalid_argument& e) {
    SWIG_exception(SWIG_ValueError, e.what());
  }
  catch (const GException::invalid_return_value& e) {
    SWIG_exception(SWIG_ValueError, e.what());
  }
  catch (const GException::fits_error& e) {
    SWIG_exception(SWIG_IOError, e.what());
  }
  catch (const GException::file_error& e) {
    SWIG_exception(SWIG_IOError, e.what());
  }
  catch (const GException::runtime_error& e) {
    SWIG_exception(SWIG_RuntimeError, e.what());
  }
  catch (const GException& e) {
    SWIG_exception(SWIG_RuntimeError, e.what());
```

```
  }
  catch (const std::exception& e) {
    SWIG_exception(SWIG_RuntimeError, e.what());
  }
  catch (...) {
    SWIG_exception(SWIG_RuntimeError, "unknown exception");
  }
```

the exception that is catch is std::exception&, hence the GException class instances are ignored.

#20 - 05/28/2021 08:43 AM - Knödlseder Jürgen

The page https://stackoverflow.com/questions/6324299/problem-throwing-and-catching-custom-exceptions-in-c hints that the exception may not be exported properly.

One possibility is that the gcc compilers are too old: https://github.com/aalexand/sharedlib_typeinfo.

And here some code that manipulates dynamic load flags: http://doxygen.lsst.codes/stack/doxygen/x_11_0/lsstimport_8py_source.html. I tried setting the flag to 258 before loading the module but this did not change the behaviour.

#21 - 05/28/2021 09:21 AM - Knödlseder Jürgen

Inspecting the symbols of the Python module it seems that the GammaLib exception methods are all there:

```
$ nm --demangle pyext/gammalib/_sky.so | grep Exception
00000000002f6460 d DW.ref._ZTI10GException
00000000002f6468 d DW.ref._ZTIN10GException10file_errorE
00000000002f6470 d DW.ref._ZTIN10GException10fits_errorE
00000000002f6478 d DW.ref._ZTIN10GException12out_of_rangeE
00000000002f6480 d DW.ref._ZTIN10GException13invalid_valueE
00000000002f6488 d DW.ref._ZTIN10GException13runtime_errorE
00000000002f6490 d DW.ref._ZTIN10GException16invalid_argumentE
00000000002f6498 d DW.ref._ZTIN10GException20invalid_return_valueE
                 U GException::out_of_range::out_of_range(std::string const&, std::string const&, int const&, int const&, std::string const&)
00000000000affc0 W GException::out_of_range::~out_of_range()
00000000000aff00 W GException::out_of_range::~out_of_range()
                 U GException::invalid_argument::invalid_argument(std::string const&, std::string const&)
00000000000b0080 W GException::invalid_argument::~invalid_argument()
00000000000af1d0 W GException::invalid_argument::~invalid_argument()
                 U GExceptionHandler::what() const
00000000002ef330 V typeinfo for GException
                 U typeinfo for GExceptionHandler
00000000002ef2f0 V typeinfo for GException::file_error
00000000002ef2d0 V typeinfo for GException::fits_error
00000000002ee9c0 V typeinfo for GException::out_of_range
00000000002ef290 V typeinfo for GException::invalid_value
00000000002ef310 V typeinfo for GException::runtime_error
00000000002eea10 V typeinfo for GException::invalid_argument
00000000002ef2b0 V typeinfo for GException::invalid_return_value
00000000000d300e V typeinfo name for GException
00000000000d2fd0 V typeinfo name for GException::file_error
00000000000d2fb0 V typeinfo name for GException::fits_error
00000000000d2800 V typeinfo name for GException::out_of_range
00000000000d2f50 V typeinfo name for GException::invalid_value
00000000000d2ff0 V typeinfo name for GException::runtime_error
00000000000d2820 V typeinfo name for GException::invalid_argument
00000000000d2f80 V typeinfo name for GException::invalid_return_value
```

```
                 U vtable for GExceptionHandler
00000000002ee9e0 V vtable for GException::out_of_range
00000000002eea40 V vtable for GException::invalid_argument
```

Note however that the vtable does not exist for GException::invalid_value. For comparison, here is the result on CentOS 6. Things look pretty much identical there:

```
$ nm --demangle pyext/gammalib/_sky.so | grep Exception
00000000002f6660 d DW.ref._ZTI10GException
00000000002f6668 d DW.ref._ZTIN10GException10file_errorE
00000000002f6670 d DW.ref._ZTIN10GException10fits_errorE
00000000002f6678 d DW.ref._ZTIN10GException12out_of_rangeE
00000000002f6680 d DW.ref._ZTIN10GException13invalid_valueE
00000000002f6688 d DW.ref._ZTIN10GException13runtime_errorE
00000000002f6690 d DW.ref._ZTIN10GException16invalid_argumentE
00000000002f6698 d DW.ref._ZTIN10GException20invalid_return_valueE
                 U GException::out_of_range::out_of_range(std::string const&, std::string const&, int const&, int const&, std::string const&)
00000000000ade80 W GException::out_of_range::~out_of_range()
00000000000add00 W GException::out_of_range::~out_of_range()
00000000000add00 W GException::out_of_range::~out_of_range()
                 U GException::invalid_argument::invalid_argument(std::string const&, std::string const&)
00000000000adf40 W GException::invalid_argument::~invalid_argument()
00000000000addc0 W GException::invalid_argument::~invalid_argument()
00000000000addc0 W GException::invalid_argument::~invalid_argument()
                 U GExceptionHandler::what() const
00000000002ef430 V typeinfo for GException
                 U typeinfo for GExceptionHandler
00000000002ef3f0 V typeinfo for GException::file_error
00000000002ef3d0 V typeinfo for GException::fits_error
00000000002eeab0 V typeinfo for GException::out_of_range
00000000002ef390 V typeinfo for GException::invalid_value
00000000002ef410 V typeinfo for GException::runtime_error
00000000002eeb10 V typeinfo for GException::invalid_argument
00000000002ef3b0 V typeinfo for GException::invalid_return_value
00000000000d0e0e V typeinfo name for GException
00000000000d0dd0 V typeinfo name for GException::file_error
00000000000d0db0 V typeinfo name for GException::fits_error
00000000000d05f0 V typeinfo name for GException::out_of_range
00000000000d0d50 V typeinfo name for GException::invalid_value
00000000000d0df0 V typeinfo name for GException::runtime_error
00000000000d0620 V typeinfo name for GException::invalid_argument
00000000000d0d80 V typeinfo name for GException::invalid_return_value
                 U vtable for GExceptionHandler
00000000002eeae0 V vtable for GException::out_of_range
00000000002eeb40 V vtable for GException::invalid_argument
```

#22 - 05/28/2021 09:41 AM - Knödlseder Jürgen

Adding the code

```
   catch (const std::exception& e) {
printf("std::exception\n");
     const GException::invalid_argument* ptr1 = dynamic_cast<const GException::invalid_argument*>(&e);
     const GException::invalid_value*    ptr2 = dynamic_cast<const GException::invalid_value*>(&e);
printf("%p, %p\n", ptr1, ptr2);
```

results in

```
std::exception
(nil), (nil)
```

which means that the issue is that virtual tables are not exported as they should.

#23 - 05/28/2021 10:03 AM - Knödlseder Jürgen

Checking the symbols in GammaLib on Debian also looks reasonable (and similar to CentOS 6):

```
$ nm --demangle src/.libs/libgamma.so | grep Exception
000000000083da18 d DW.ref._ZTIN10GException13invalid_valueE
000000000083d978 d DW.ref._ZTIN10GException13runtime_errorE
000000000083d970 d DW.ref._ZTIN10GException16invalid_argumentE
000000000083da10 d DW.ref._ZTIN10GException20invalid_return_valueE
0000000000118ff0 t global constructors keyed to GException.cpp
00000000001191a0 T GException::file_error::file_error(std::string const&, std::string const&)
0000000000119270 T GException::file_error::file_error(std::string const&, std::string const&)
000000000011ccb0 W GException::file_error::~file_error()
000000000011ccd0 W GException::file_error::~file_error()
000000000011a110 T GException::fits_error::fits_error(std::string const&, int const&, std::string const&)
000000000011a490 T GException::fits_error::fits_error(std::string const&, int const&, std::string const&)
000000000011ccf0 W GException::fits_error::~fits_error()
000000000011cd10 W GException::fits_error::~fits_error()
00000000001195b0 T GException::test_error::test_error(std::string const&, std::string const&)
0000000000119700 T GException::test_error::test_error(std::string const&, std::string const&)
000000000011cbb0 W GException::test_error::~test_error()
000000000011cbd0 W GException::test_error::~test_error()
000000000011b220 T GException::out_of_range::out_of_range(std::string const&, std::string const&, int const&, int const&, std::string const&)
000000000011a9e0 T GException::out_of_range::out_of_range(std::string const&, std::string const&, int const&, int const&, std::string const&)
000000000011cd70 W GException::out_of_range::~out_of_range()
000000000011cd90 W GException::out_of_range::~out_of_range()
0000000000119850 T GException::test_failure::test_failure(std::string const&, std::string const&)
00000000001199a0 T GException::test_failure::test_failure(std::string const&, std::string const&)
000000000011cbf0 W GException::test_failure::~test_failure()
000000000011cc10 W GException::test_failure::~test_failure()
0000000000119410 T GException::invalid_value::invalid_value(std::string const&, std::string const&)
00000000001194e0 T GException::invalid_value::invalid_value(std::string const&, std::string const&)
000000000011cdf0 W GException::invalid_value::~invalid_value()
000000000011ce10 W GException::invalid_value::~invalid_value()
000000000011c400 T GException::runtime_error::runtime_error(std::string const&, std::string const&)
000000000011c510 T GException::runtime_error::runtime_error(std::string const&, std::string const&)
```

```
000000000011cd30 W GException::runtime_error::~runtime_error()
000000000011cd50 W GException::runtime_error::~runtime_error()
000000000011ba60 T GException::invalid_argument::invalid_argument(std::string const&, std::string const&)
0000000000119d90 T GException::invalid_argument::invalid_argument(std::string const&, std::string const&, std::string const&)
0000000000119340 T GException::invalid_argument::invalid_argument(std::string const&, std::string const&)
0000000000119f50 T GException::invalid_argument::invalid_argument(std::string const&, std::string const&, std::string const&)
000000000011cb90 W GException::invalid_argument::~invalid_argument()
000000000011cb70 W GException::invalid_argument::~invalid_argument()
00000000002b4100 W GException::invalid_return_value::invalid_return_value(GException::invalid_return_value const&)
000000000011c620 T GException::invalid_return_value::invalid_return_value(std::string const&, std::string const&)
000000000011c730 T GException::invalid_return_value::invalid_return_value(std::string const&, std::string const&)
000000000011cdb0 W GException::invalid_return_value::~invalid_return_value()
000000000011cdd0 W GException::invalid_return_value::~invalid_return_value()
0000000000119af0 T GException::test_nested_try_error::test_nested_try_error(std::string const&, std::string const&)
0000000000119c40 T GException::test_nested_try_error::test_nested_try_error(std::string const&, std::string const&)
000000000011cc30 W GException::test_nested_try_error::~test_nested_try_error()
000000000011cc50 W GException::test_nested_try_error::~test_nested_try_error()
0000000000119020 T GException::feature_not_implemented::feature_not_implemented(std::string const&, std::string const&)
00000000001190e0 T GException::feature_not_implemented::feature_not_implemented(std::string const&, std::string const&)
000000000011cc70 W GException::feature_not_implemented::~feature_not_implemented()
000000000011cc90 W GException::feature_not_implemented::~feature_not_implemented()
000000000011c930 W GExceptionHandler::~GExceptionHandler()
000000000011c9f0 W GExceptionHandler::~GExceptionHandler()
000000000011cab0 W GExceptionHandler::~GExceptionHandler()
000000000011a810 T GExceptionHandler::what() const
0000000000824a30 V typeinfo for GExceptionHandler
0000000000824c70 V typeinfo for GException::file_error
0000000000824cd0 V typeinfo for GException::fits_error
0000000000824af0 V typeinfo for GException::test_error
0000000000824d90 V typeinfo for GException::out_of_range
0000000000824b50 V typeinfo for GException::test_failure
0000000000824e50 V typeinfo for GException::invalid_value
0000000000824d30 V typeinfo for GException::runtime_error
0000000000824a50 V typeinfo for GException::invalid_argument
0000000000824df0 V typeinfo for GException::invalid_return_value
0000000000824bb0 V typeinfo for GException::test_nested_try_error
0000000000824c10 V typeinfo for GException::feature_not_implemented
0000000000531a20 V typeinfo name for GExceptionHandler
0000000000531b30 V typeinfo name for GException::file_error
0000000000531b50 V typeinfo name for GException::fits_error
0000000000531a70 V typeinfo name for GException::test_error
0000000000531b90 V typeinfo name for GException::out_of_range
0000000000531a90 V typeinfo name for GException::test_failure
0000000000531bf0 V typeinfo name for GException::invalid_value
0000000000531b70 V typeinfo name for GException::runtime_error
0000000000531a40 V typeinfo name for GException::invalid_argument
0000000000531bc0 V typeinfo name for GException::invalid_return_value
0000000000531ac0 V typeinfo name for GException::test_nested_try_error
0000000000531b00 V typeinfo name for GException::feature_not_implemented
0000000000824a00 V vtable for GExceptionHandler
0000000000824c40 V vtable for GException::file_error
0000000000824ca0 V vtable for GException::fits_error
0000000000824ac0 V vtable for GException::test_error
0000000000824d60 V vtable for GException::out_of_range
0000000000824b20 V vtable for GException::test_failure
0000000000824e20 V vtable for GException::invalid_value
0000000000824d00 V vtable for GException::runtime_error
0000000000824a80 V vtable for GException::invalid_argument
0000000000824dc0 V vtable for GException::invalid_return_value
0000000000824b80 V vtable for GException::test_nested_try_error
0000000000824be0 V vtable for GException::feature_not_implemented
```

#24 - 05/28/2021 10:14 AM - Knödlseder Jürgen

I added an equivalent C++ test

```
// Test saving
test_try("Test saving");
try {
    GSkyMap map("GAL", 1, "RING", 1);
    map.save("test_cpp_skymap_hpx_v2.fits", true);
    map.save("test_cpp_skymap_hpx_v2.fits");
    test_try_failure();
}
catch (GException::invalid_value &e) {
    test_try_success();
}
catch (std::exception &e) {
    test_try_failure(e);
}
```

and this test succeded

```
<testcase classname="GSky" name="Test Healpix GSkyMap constructors: Test saving" time="0.000" />
```

Hence the issue seems to be related to the Python interface.

#25 - 05/28/2021 10:23 AM - Knödlseder Jürgen

I found a solution!

By adding

```
import sys

# Get original flags
originalDLFlags = sys.getdlopenflags()

# Set RTLD_GLOBAL | RTLD_NOW
sys.setdlopenflags(258)

# Import modules
from gammalib.app import *
...

# Reset original flags
sys.setdlopenflags(originalDLFlags)
```

to pyext/__init__.py the issue was solved.

#26 - 05/28/2021 02:56 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

- % Done changed from 90 to 100

Code was merged into devel.