

Add api-sanity-checker on Jenkins for ctools

11/02/2012 06:51 PM - Deil Christoph

Status:	New	Start date:	11/02/2012
Priority:	Normal	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

We already run the api-sanity-checker for gammalib (see issue #579) and it's discovered (admittedly extremely few given the size of the library) problems:
https://cta-jenkins.irap.omp.eu/job/gammalib-sanity/API_Sanity_Checker_report/

Although this is much less common, the ctools can also be used as a C++ library (e.g. one could wrap or subclass a ctool GApplication in another C++ or python tool).
Running it through the the api-sanity-checker can be useful to avoid any program crashes or hangs.

How to run the api-sanity-checker on the ctools API

It's not trivial to generate the ctools tests without generating also the gammalib tests, as far as I can see there's no option to give it the gammalib header include path, I've asked how to do this here:
<https://github.com/lvc/api-sanity-checker/issues/1>

For now a workaround is to add GammaLib.hpp to the headers section in VERSION.xml

```
<version>0.0</version>
<headers>
/home/deil/software/code/ctools/src
/usr/local/gamma/include/gammalib/GammaLib.hpp
</headers>
<libs>/usr/local/gamma/lib</libs>
```

and then to explicitly only generate tests for each ctools header file one by one:

```
api-sanity-checker -lib libctools -d VERSION.xml -header ctskymap.hpp -gen -build -run
api-sanity-checker -lib libctools -d VERSION.xml -header ctobssim.hpp -gen -build -run
api-sanity-checker -lib libctools -d VERSION.xml -header ctlike.hpp -gen -build -run
api-sanity-checker -lib libctools -d VERSION.xml -header ctbin.hpp -gen -build -run
api-sanity-checker -lib libctools -d VERSION.xml -header ctselect.hpp -gen -build -run
```

Segfaults found

ctlike

The catlike copy constructor and operator= segfault because in line 506 of ctlike.cpp the NULL pointer app.m_opt is dereferenced:

```
m_opt = app.m_opt->clone();
```

Here's the test code for the first failure:

```
#include <ctlike/ctlike.hpp>
int main(int argc, char *argv[])
```

```
{
    GObservations obs;
    ctlike app(obs);
    ctlike* obj = new ctlike(app); //target call
    return 0;
}
```

Is there a single use case to ever copy a GApplication?

If not I'd recommend making the copy constructor and operator= empty and private to disable copying of GApplications.

ctselect

ctselect segfaults for this test case:

```
#include <ctselect/ctselect.hpp>
int main(int argc, char *argv[])
{
    GObservations obs;
    ctselect* obj = new ctselect(obs);
    std::basic_string<char> instrument = "instrument";
    GCTAObservation* obs_1 = new GCTAObservation(instrument);
    std::basic_string<char> filename = "testdata/sample.txt";
    obj->select_events(obs_1, filename); //target call
    return 0;
}
```

The crash occurs in line 495 of ctselect.cpp because the NULL pointer pnt is dereferenced:

```
$ cat tests/libctools/0.0/run_fail_list
tests/libctools/0.0/groups/ctselect/classes/ctselect/_ZN8ctselect13select_eventsEP15GCTAObservationRKSs

deil@ubuntu:~/software/code/ctools_sanity$ gdb
tests/libctools/0.0/groups/ctselect/classes/ctselect/_ZN8ctselect13select_eventsEP15GCTAObservationRKSs/test
...
(gdb) run
Starting program:
```

```
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
```

```
Program received signal SIGSEGV, Segmentation fault.
ctselect::select_events (this=0x606ae0, obs=0x608630, filename=...) at ctselect.cpp:495
495      ra = pnt->dir().ra_deg();
```

```
(gdb) print pnt
$1 = (const GCTAPointing *) 0x0
```

History

#1 - 11/07/2012 02:18 PM - Deil Christoph

OK, the api-sanity-checker author has helped me figure out how to find the gammalib header and lib for the ctools tests:

<https://github.com/lvc/api-sanity-checker/issues/1#issuecomment-10147606>

Jürgen, could you add this as a nightly Jenkins ctools-sanity job?

```
$ cat VERSION.xml
<version>0.0</version>
<headers>/home/deil/software/code/ctools/src</headers>
<libs>/usr/local/gamma/lib/libctools.so</libs>
<add_include_paths>/usr/local/gamma/include/gammalib</add_include_paths>
<libs_depend>/usr/local/gamma/lib/libgamma.so.0</libs_depend>

$ api-sanity-checker -lib libctools -d VERSION.xml -gen -build -run
library(ies) analysis: [100.00%]
header(s) analysis: [100.00%]
generating tests: 134/134 [100.00%], success/fail: 129/5
1. see generated test suite in the directory 'tests/libctools/0.0/'
2. for viewing the tests use 'tests/libctools/0.0/view_tests.html'
3. use -build option for building tests
building tests: 129/129 [100.00%], success/fail: 129/0
running tests: 129/129 [100.00%], success/fail: 125/4
creating report ...
see test results in the file:
  test_results/libctools/0.0/test_results.html
```