ctools - Bug #1149

running in 'for'-loop ctselect fills up the memory very quickly

02/20/2014 10:59 AM - Mayer Michael

Status:	Closed	**Start date:**	02/20/2014
Priority:	Immediate	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	00-08-00		

Description

It seems that in one of the latest revisions (I couldn't track down which one), something has been introduced that prevents to run several ctselect jobs in a for loop:
The following code crashes very quickly:

```
obs = GObservations()
cta_obs = GCTAObservation()
cta_obs.load_unbinned("crab_evt_list.fits")
obs.append(cta_obs)

for i in range(10000):
    print i
    select = ctselect(obs)
    select["ra"].real(83.633)
    select["dec"].real(22.01)
    select["rad"].real(2.5)
    select["tmin"].real(0.0)
    select["tmax"].real(0.0)
    select["emin"].real(1.0)
    select["emax"].real(10.0)
    select.run()
```

The resulting error message is *RuntimeError: *** ERROR in GFits::open(std::string&): Unable to open FITS file "ctselectaXxrCI" (status=105)*. The problem occurred also on different platforms (After loop 124 on my mac and after loop 508 on Scientific Linux, I guess this depends on the memory size).
I tracked down the problem in the code and it seems that in line 1035 of GFits.cpp FHANDLE(m_fitsfile) returns a null pointer at some point. So a new fits file can not be initialised by __ffinit() and status=105 is returned. According to the fits error status this is due to a problem with the operating system. So I guess the memory just gets filled very quickly.

This piece of code would have worked for me one month ago.

P.S. The example code is just to illustrate the problem. My problem occurs as well when having a larger observation container (size=100). Then it already happens after ~4 loops...

History

#1 - 02/20/2014 11:53 AM - Mayer Michael

I am a bit clueless about the origin of this problem. It might as well correspond to a gammalib issue rather than ctools.

#2 - 02/20/2014 12:04 PM - Knödlseder Jürgen

- Assigned To set to Knödlseder Jürgen

This look like the ctselect destructor does not close the FITS file. I'll try to reproduce this and keep you posted ...

#3 - 02/20/2014 12:11 PM - Knödlseder Jürgen

Stange, on my Mac this runs without any problem, and monitoring with top does not reveal any memory eater.

What branch are you using?

#4 - 02/20/2014 12:15 PM - Knödlseder Jürgen

I get another interesting problem:

```
1278
Traceback (most recent call last):
  File "./test.py", line 15, in <module>
  File "/usr/local/gamma/lib/python2.6/site-packages/ctools.py", line 422, in __init__
RuntimeError: *** ERROR in GApplicationPars::read(std::string&): Unable to open parameter file "/Users/jurgen/pfiles/ctselect.par".
```

This is repeatable on my machine.

#5 - 02/20/2014 12:35 PM - Knödlseder Jürgen

By the way, your problem is related to temporary files that are created by ctselect, see http://www.gnu.org/software/libc/manual/html_node/Temporary-Files.html for some information. I recognized that the files are eventually not closed properly in ctselect, hence this may explain the problem. I'll do some check here and let you know. However, as I cannot reproduce the problem I'll send you some patch to check this on your side.

#6 - 02/20/2014 12:40 PM - Knödlseder Jürgen

You may put the following code in ctselect.cpp from line 631 on:

```
    // Get temporary file name
    //std::string tmpname = std::tmpnam(NULL);
    char tpl[]  = "ctselectXXXXXX";
    int  fileid = mkstemp(tpl);
    std::string tmpname(tpl);

    // Save FITS file to temporary file
    file.saveto(tmpname, true);

    // Load observation from temporary file
    obs->load_unbinned(tmpname);

    // Close temporary file
    close(fileid);

    // Remove temporary file
    std::remove(tmpname.c_str());
```

(in fact, the thing that changed w/r to the old code is the usage of mkstemp instead of std::tmpnam, but I did not recognized that the new function opens in fact the file).

#7 - 02/20/2014 01:18 PM - Knödlseder Jürgen

- Target version set to 00-07-02

- % Done changed from 0 to 50

I finally could reproduce the problem by removing the parameter file locker from GApplicationPars::read. I guess the problem is the same in both cases, i.e. the maximum number of open files is exceeded. Continue to track this down ...

```
1278
Traceback (most recent call last):
  File "./test.py", line 23, in <module>
  File "/usr/local/gamma/lib/python2.6/site-packages/ctools.py", line 437, in run
RuntimeError: *** ERROR in GFits::open(std::string&): Unable to open FITS file "ctselectwNd8Td" (status=105)
```

#8 - 02/20/2014 01:27 PM - Mayer Michael

Thanks for your quick reply and your efforts.

> What branch are you using?

I am on devel

> You may put the following code in ctselect.cpp from line 631 on:

my mac compiler did not know the close-function. On linux, I still have the same problem after introducing the close-command.

I also got the problem in GApplicationPars::read() at some points during my tests. So you are right, it must be about to many opened files at the same time.

I will try with checking out a new, clean version of gammalib and ctools to ensure that I don't use some obsolete library files.

> Continue to track this down ...

thanks a lot

#9 - 02/20/2014 01:33 PM - Knödlseder Jürgen

Indeed, I recognized that this does not fix the problem, just moved it twice as much iterations later on my system. I probably should not use mkstemp which opens a file ... I'll keep you posted.

#10 - 02/20/2014 01:53 PM - Mayer Michael

Mayer Michael wrote:

> I will try with checking out a new, clean version of gammalib and ctools to ensure that I don't use some obsolete library files.

just checked with a new version - the problem remains.
However, using std::tmpnam instead of mkstemp, the problem disappears all of a sudden.

Should we change back to std::tmpnam or is it considered unsafe compared to mkstemp?

#11 - 02/20/2014 02:08 PM - Knödlseder Jürgen

- % Done changed from 50 to 70

In fact, I use the logic in two places, and did only correct the second one wink.png

Now it seems to work.

I changed from std::tmpnam to mkstemp because the compiler issues a warning about a security hole, and recommended to use mkstemp. This was the motivation to make this change.

I also recognized that the second mkstemp step can be avoided by adding a load method to GCTAObservation that takes a GFits object as argument. So I made also this change to GammaLib. By the way, in the long-run we can drop the load_binned and load_unbinned methods, as I also implemented a load method that automatically recognizes the file type.

#12 - 02/20/2014 02:40 PM - Knödlseder Jürgen

- Status changed from New to Feedback

- % Done changed from 70 to 100

Should now be solved. Please checkout devel and verify ...

#13 - 02/20/2014 03:20 PM - Mayer Michael

- Status changed from Feedback to Resolved

Excellent. This works just fine.
The new GCTAObservation::load-function is also great to unify (and simplify) the interfaces.
Thanks for reacting so quickly and resolving this problem.

#14 - 07/19/2014 02:15 AM - Knödlseder Jürgen

- Target version changed from 00-07-02 to 00-08-00

#15 - 07/20/2014 11:17 PM - Knödlseder Jürgen

- Status changed from Resolved to Closed