

ctools - Bug #547

make check GException::par_file_open_error on Scientific Linux 6.3

10/11/2012 12:21 PM - Deil Christoph

Status:	Closed	Start date:	10/11/2012
Priority:	Normal	Due date:	
Assigned To:	Knödlseeder Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>I installed the latest releases gammalib 00-06-02 and ctools 00-05-01 at work and got GException::par_file_open_error when running make check:</p>			
<pre>Making check in test make[1]: Entering directory `/lfs/l1/hess/users/deil/scratch/software/cta/gcc/ctools-00-05-01/test' make check-TESTS make[2]: Entering directory `/lfs/l1/hess/users/deil/scratch/software/cta/gcc/ctools-00-05-01/test' ***** * Test ctools * ***** Test ctobssim: terminate called after throwing an instance of 'GException::par_file_open_error' what(): *** ERROR in GPars::read(std::string): Unable to open parameter file 'pfiles/ctobssim.par'. Could not get a lock on the file. ./test_ctools.sh: line 73: 16555 Aborted (core dumped) \$ctobssim infile="data/crab.xml" outfile="events.fits" caldb="irf" irf="kb_E_50h_v3" ra=83.63 dec=22.01 rad=10.0 tmin=0.0 tmax=1800.0 emin=0.1 emax=100.0 . events.fits file is not valid FAIL: test_ctools.sh ***** * Test cscripts * ***** Test cspull: Traceback (most recent call last): File "/home/hfm/deil/scratch/software/cta/gcc/ctools-00-05-01/test/./scripts/cspull.py", line 354, in <module> app = cspull(sys.argv) File "/home/hfm/deil/scratch/software/cta/gcc/ctools-00-05-01/test/./scripts/cspull.py", line 53, in __init__ file = self.parfile() File "/home/hfm/deil/scratch/software/cta/gcc/ctools-00-05-01/test/./scripts/cspull.py", line 113, in parfile pars.save(parfile) File "/home/hfm/deil/scratch/software/cta/gcc/install/lib64/python2.6/site-packages/gammalib/app.py", line 468, in save return _app.GPars_save(self, *args) RuntimeError: *** ERROR in GPars::write(std::string): Unable to open parameter file 'pfiles/cspull.par'. Could not get a lock on the file. Exception TypeError: "in method 'GApplication_logTerse', argument 1 of type 'GApplication const ***' in <bound method cspull.__del__ of <__main__.cspull; >> ignored . pull.dat file is not valid FAIL: test_cscripts.sh ***** * ctools Python interface testing * ***** Test executable analysis: Traceback (most recent call last): File "./test_python.py", line 201, in <module> pipeline_v1() File "./test_python.py", line 62, in pipeline_v1 sim = ctobssim() File "/lfs/l1/hess/users/deil/scratch/software/cta/gcc/ctools-00-05-01/pyext/ctools.py", line 175, in __init__ this = _ctools.new_ctobssim(*args) RuntimeError: *** ERROR in GPars::read(std::string): Unable to open parameter file 'pfiles/ctobssim.par'. Could not get a lock on the file. FAIL: test_python.py ===== 3 of 3 tests failed</pre>			

Please report to jurgen.knodlseder@irap.omp.eu

```
=====
make[2]: *** [check-TESTS] Error 1
make[2]: Leaving directory `/lfs/l1/hess/users/deil/scratch/software/cta/gcc/ctools-00-05-01/test'
make[1]: *** [check-am] Error 2
make[1]: Leaving directory `/lfs/l1/hess/users/deil/scratch/software/cta/gcc/ctools-00-05-01/test'
make: *** [check-recursive] Error 1
```

- Why are the par files not found (but are found on my Mac using the same installation procedure)?
- Even if the par file is not found, why do I get "core dumped" from test_ctools.sh ?

I tried two compilers (in separate build / installs of course), both give the same problem:

- gcc (GCC) 4.4.6 20120305 (Red Hat 4.4.6-4)
- icc (ICC) 13.0.0 20120731

I don't see this problem on my Mac so in case it matters, here's the Linux version info:

Here's

```
cat /etc/system-release
Scientific Linux release 6.3 (Carbon)
cat /proc/version
Linux version 2.6.32-279.9.1.el6.x86_64 (mockbuild@sl6.fnal.gov) (gcc version 4.4.6 20120305 (Red Hat 4.4.6-4) (GCC) ) #1 SMP
Tue Sep 25 14:55:44 CDT 2012
```

Related issues:

Related to GammaLib - Feature # 550: Add configure test to checks if file loc...

Closed

10/11/2012

History

#1 - 10/11/2012 12:35 PM - Deil Christoph

After running make install I get this:

```
lfs1> ctbin
terminate called after throwing an instance of 'GException::par_file_open_error'
  what():  *** ERROR in GPars::read(std::string): Unable to open parameter file '/home/hfm/deil/scratch/software/cta/gcc/install//syspfiles/ctbin.par'.
  Could not get a lock on the file.
Aborted (core dumped)
lfs1> ls -lh /home/hfm/deil/scratch/software/cta/gcc/install//syspfiles/ctbin.par
-rw-r--r-- 1 deil hfm 2.8K Oct 11 12:30 /home/hfm/deil/scratch/software/cta/gcc/install//syspfiles/ctbin.par
lfs1> which ctbin
~/scratch/software/cta/gcc/install/bin/ctbin
lfs1> gdb ctbin
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-56.el6)
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /lfs/l1/hess/users/deil/scratch/software/cta/gcc/install/bin/ctbin...done.
(gdb) run
Starting program: /lfs/l1/hess/users/deil/scratch/software/cta/gcc/install/bin/ctbin
[Thread debugging using libthread_db enabled]
terminate called after throwing an instance of 'GException::par_file_open_error'
  what():  *** ERROR in GPars::read(std::string): Unable to open parameter file '/home/hfm/deil/scratch/software/cta/gcc/install//syspfiles/ctbin.par'.
  Could not get a lock on the file.
```

Program received signal SIGABRT, Aborted.

0x00007ffff62108a5 in raise () from /lib64/libc.so.6

Missing separate debuginfos, use: debuginfo-install cfitsio-3.240-3.el6.x86_64 glibc-2.12-1.80.el6_3.5.x86_64 libgcc-4.4.6-4.el6.x86_64

libgomp-4.4.6-4.el6.x86_64 libstdc++-4.4.6-4.el6.x86_64 ncurses-libs-5.7-3.20090208.el6.x86_64 readline-6.0-4.el6.x86_64

(gdb) bt

```
#0 0x00007ffff62108a5 in raise () from /lib64/libc.so.6
#1 0x00007ffff6212085 in abort () from /lib64/libc.so.6
#2 0x00007ffff6ef3a5d in __gnu_cxx::__verbose_terminate_handler() () from /usr/lib64/libstdc++.so.6
#3 0x00007ffff6ef1be6 in ?? () from /usr/lib64/libstdc++.so.6
#4 0x00007ffff6ef1c13 in std::terminate() () from /usr/lib64/libstdc++.so.6
#5 0x00007ffff6ef1d0e in __cxa_throw () from /usr/lib64/libstdc++.so.6
#6 0x00007ffff7ae19ad in GParams::read (this=0x7ffff6fdae0, filename="/home/hfm/deil/scratch/software/cta/gcc/install//sysfiles/ctbin.par") at
  GParams.cpp:899
#7 0x00007ffff7ae2cad in GParams::load (this=0x7ffff6fdae0, filename="ctbin.par", args=std::vector of length 1, capacity 1 = {...}) at GParams.cpp:440
#8 0x00007ffff7adafb2 in GApplication::GApplication (this=0x7ffff6fda78, name=<value optimized out>, version=<value optimized out>, argc=1,
  argv=<value optimized out>) at GApplication.cpp:117
#9 0x0000000000404050 in ctbin::ctbin (this=0x7ffff6fda70, argc=1, argv=0x7ffff6fdd78) at ctbin.cpp:95
#10 0x000000000040392b in main (argc=<value optimized out>, argv=<value optimized out>) at main.cpp:50
```

I don't know why ctbin can't get a lock on the file. The file exists and I can read and write it just fine.
Does the backtrace help figuring out why a core dump occurs?

#2 - 10/11/2012 02:00 PM - Knödseder Jürgen

This is indeed strange. I also have a Scientific Linux 6 installed in our virtual test box, and there it works perfectly. The detailed information of our system is:

```
SL 6.2
Python 2.6.5
gcc 4.4.4
kernel 2.6.32-71.el6.x86_64
cfitsio 3.240
```

The locking of the parfiles is a very specific action, that I introduced to allow parallel processing of ctools. Otherwise it could happen that two instances of an executable try to write at the same time to the same file. I had this problem when I was running many jobs in parallel.

I think that the file locking operations are not 100%, so I was indeed looking for cases where they do not work. However, it is weird that this happens on SL which is a quite standard system, and the same type of system that I test here.

I have to think a little more about this. Eventually, I'll send you a modified GParams.cpp file so that we get some more diagnostics.

I think that the core dump comes from the fact that an exception was thrown, so I'm not really surprised about this.

#3 - 10/11/2012 02:10 PM - Deil Christoph

Jürgen Knödseder wrote:

I think that the file locking operations are not 100%, so I was indeed looking for cases where they do not work. However, it is weird that this happens on SL which is a quite standard system, and the same type of system that I test here.

We updated to SL 6.3 last week.
But it could also be our file system setup or ...

I think that the core dump comes from the fact that an exception was thrown, so I'm not really surprised about this.

Sorry, I thin we've discussed this before, but I don't remember.
As far as I know every Linux problem should always have a top-level catch and never core dump, except maybe on developer machines?

#4 - 10/11/2012 03:09 PM - Deil Christoph

Jürgen Knödseder wrote:

The locking of the parfiles is a very specific action, that I introduced to allow parallel processing of ctools. Otherwise it could happen that two instances of an executable try to write at the same time to the same file. I had this problem when I was running many jobs in parallel.

I think that the file locking operations are not 100%, so I was indeed looking for cases where they do not work. However, it is weird that this happens on SL which is a quite standard system, and the same type of system that I test here.

Our sys admin commented that the disk I was using is an old Lustre file system, which does not support file locking.
This is the same system that I plan to use in the future running 100+ parallel ctools jobs.

The ftools / chandra / Fermi software must have a different mechanism for avoiding problems of multiple jobs trying to access the same pfile.
I don't remember ever seing this locking problem, I did have a problem about /dev/tty when running parallel fcopy, there the solution was to set this in the submitted scripts:
export HEADASNOQUERY=1

#5 - 10/11/2012 03:43 PM - Knödseder Jürgen

- *Status changed from New to In Progress*
- *Assigned To set to Knödseder Jürgen*
- *% Done changed from 0 to 10*

In fact, the Fermi software does not avoid the problem. What I do to circumvent the problem is setting PFILES to a local working directory, hence each job has its own par file. Using centralized parameter files, I get the same concurrent file access problem as with GammaLib without file locking.

Note that you need a large number of jobs to run into this problem, as the file access is very short. I think it is at least of the order of a hundred jobs, maybe even more ...

I added the feature request #550 to check for file locking in the configuration step, and only enable file locking if the system also supports it.

#6 - 10/11/2012 10:05 PM - Knödseder Jürgen

- *Status changed from In Progress to Feedback*
- *% Done changed from 10 to 100*

The check for file locking success has been disabled, so the problem should be fixed now.

#7 - 02/07/2014 08:17 AM - Knödseder Jürgen

- *Status changed from Feedback to Closed*