# GammaLib - Feature #550

## Add configure test to checks if file locking is supported

10/11/2012 03:41 PM - Knödlseder Jürgen

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 10/11/2012 |
| **Priority:** | Normal | **Due date:** | |
| **Assigned To:** | Knödlseder Jürgen | **% Done:** | 100% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |

**Description**

On some file systems, file locking may not be supported, hence parameter file access will run into a problem.

A check should be added into the configure script to test if file locking is supported. Only if support is found, file locking should be used in the GPars class. Otherwise, files will not be locked.

This new feature should solve Bug #547.

**Related issues:**

| | | |
|---|---|---|
| Related to ctools - Bug # 547: make check GException::par_file_open_error on ... | **Closed** | **10/11/2012** |

### History

#### #1 - 10/11/2012 04:16 PM - Deil Christoph

If I understand correctly, the FTOOLS, CIAO, the Fermi ScienceTools don't lock parameter files?
This requires users to set the PFILES environment variable when running multiple processes for the same tool, see e.g.
http://polywww.in2p3.fr/activites/physique/glast/workbook/pages/Announcements/Warning_multipleBatchJobs_onSlacPublic.html

You want to add file locking to the ctools for users that have file systems that support this (auto-discovered by configure) to avoid problems for users running batch jobs without setting PFILES explicitly?

I'm not sure doing this differently than the existing ftools is a good idea, specifically I am thinking about the case where configure is run on a disk that supports file locking, but then ctools try to access parameter files on another disk that doesn't support file locking. This would actually be the case for some users on our cluster that build software in $HOME, but then do all data analysis (including setting PFILES to their working directory as you do) on a Lustre file system that doesn't support file locking.

Maybe not implementing any file locking and clearly describing the PFILES environment variable solution in the documentation is a good enough solution?

#### #2 - 10/11/2012 04:21 PM - Knödlseder Jürgen

Christoph Deil wrote:

> If I understand correctly, the FTOOLS, CIAO, the Fermi ScienceTools don't lock parameter files?
> This requires users to set the PFILES environment variable when running multiple processes for the same tool, see e.g.
> http://polywww.in2p3.fr/activites/physique/glast/workbook/pages/Announcements/Warning_multipleBatchJobs_onSlacPublic.html

Right. I wanted that GammaLib is a little more clever :-)

> You want to add file locking to the ctools for users that have file systems that support this (auto-discovered by configure) to avoid problems for users running batch jobs without setting PFILES explicitly?

That was my initial idea.

> I'm not sure doing this differently than the existing ftools is a good idea, specifically I am thinking about the case where configure is run on a disk that supports file locking, but then ctools try to access parameter files on another disk that doesn't support file locking. This would actually be the case for some users on our cluster that build software in $HOME, but then do all data analysis (including setting PFILES to their working directory as you do) on a Lustre file system that doesn't support file locking.

I agree, configure will not be able to detect this case, so this is not a viable solution.

> Maybe not implementing any file locking and clearly describing the PFILES environment variable solution in the documentation is a good enough solution?

What I would propose is to just ignore if file locking does not work. I'm explicitly throwing an exception if file locking is not working, but I could simple go one without any locking. This would be a little better than the existing setup, but still would require some fiddling with the PFILES environment variable on systems that do no support file locking. Alternatively, a GammaLib specific locking algorithm could be developed, which should then work in all cases. I though indeed about this possibility already, yet as I had not encountered any file locking problems so far, I did not pursue this idea.

#### #3 - 10/11/2012 04:26 PM - Deil Christoph

Jürgen Knödlseder wrote:

> Christoph Deil wrote:
>
> > Maybe not implementing any file locking and clearly describing the PFILES environment variable solution in the documentation is a good enough solution?
>
> What I would propose is to just ignore if file locking does not work.

Sounds good to me, since the file locking is already implemented.

#### #4 - 10/11/2012 04:50 PM - Knödlseder Jürgen

*- Status changed from New to Feedback*

*- % Done changed from 0 to 100*

I changed the code to disable the file locking check. Commit commit:daaba9b9 in hess branch or commit:f2d7016b in devel branch.

Can you check if this solves your problem?

#### #5 - 10/12/2012 01:35 PM - Deil Christoph

Jürgen Knödlseder wrote:

> I changed the code to disable the file locking check. Commit commit:daaba9b9 in hess branch or commit:f2d7016b in devel branch.
>
> Can you check if this solves your problem?

Problem solved. Thanks!

#### #6 - 10/12/2012 04:29 PM - Knödlseder Jürgen

*- Status changed from Feedback to Closed*