

GammaLib - Feature #614

Pull request: Add .gitignore to gammalib

12/05/2012 12:03 PM - Deil Christoph

Status:	Closed	Start date:	12/05/2012
Priority:	Normal	Due date:	
Assigned To:	Deil Christoph	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
I use `git status` a lot, so I added a .gitignore file for all the generated files that should be ignored by git.			
https://cta-redmine.irap.omp.eu/projects/gammalib/repository/revisions/57b1a8418ae0fa184097452146f2f2da9d825aff			
It doesn't matter here, but for the coming years, in my opinion this is not a very nice way to do code review, which I think is essential before merging into main branches where it will be part of the repo forever, compared to e.g. how nice it is to work on github. I want my own repo (actually I just made my own private repo on github).			
Jürgen, should such pull requests be assigned to you, or me, or noone?			

History

#1 - 12/05/2012 12:38 PM - Deil Christoph

Pull request: Add .gitignore file to ctools:

<https://cta-redmine.irap.omp.eu/projects/ctools/repository/revisions/f2690d66fbb0416696cbfdf53a8da4e60c1c3326>

#2 - 12/05/2012 12:40 PM - Knödlseider Jürgen

Great, champagne for your first commit !

Thanks a lot for the file. In fact, I also use git status a lot and found the many files listed files annoying; but as I'm not yet a git expert, I was not aware of the possibility to add a .gitignore file.

Concerning the merging, I guess we have to learn what the best way of working is. The release and master branches are actually protected from pushing, only the integration manager can deal with them. The devel branch is fully open, but could also be protected if we want to have a more strict code review policy. For the moment, I was thinking of leaving things as open as possible to encourage people to contribute.

#3 - 12/05/2012 12:40 PM - Knödlseider Jürgen

Christoph Deil wrote:

Pull request: Add .gitignore file to ctools:

<https://cta-redmine.irap.omp.eu/projects/ctools/repository/revisions/f2690d66fbb0416696cbfdf53a8da4e60c1c3326>

Could you put this one in the ctools project?

#4 - 12/05/2012 12:46 PM - Knödlseider Jürgen

Just merged your commit into devel, comptel and hess branches.

In the feature branch philosophy, who should remove the feature branch once the merging is done? The integration manager?

#5 - 12/05/2012 12:54 PM - Deil Christoph

Jürgen Knödlseider wrote:

Concerning the merging, I guess we have to learn what the best way of working is. The release and master branches are actually protected from pushing, only the integration manager can deal with them. The devel branch is fully open, but could also be protected if we want to have a more strict code review policy. For the moment, I was thinking of leaving things as open as possible to encourage people to contribute.

I don't think it's a good idea to have people like me and other new developers push to devel. Once a commit is in devel it's public and we really shouldn't remove it. Also I like to have a new branch for every little feature, I don't want to work in one hess branch, and I also don't want a central repo with hundreds of branches from all the developers.

I love the github workflow with forks and code review via pull requests. Check this out:
<http://docs.astropy.org/en/latest/#developer-documentation>

I can see the problem that there are no free tools as nice as github you could install at IRAP and you don't want to issues / code review / project history be split between github or bitbucket and IRAP, so at the moment I don't have concrete suggestion, but just the general comment that I think doing more on github and fully open source would be great.

In the end this needs to be discussed on a CTA developer mailing list, there was a thread once, but I don't think anything was decided or concrete plans were made.

#6 - 12/05/2012 12:59 PM - Deil Christoph

Jürgen Knödlseider wrote:

Just merged your commit into devel, comptel and hess branches.

In the feature branch philosophy, who should remove the feature branch once the merging is done? The integration manager?

I only know the open source method where every developer forks and has their own public repo. I can't imagine that for CTA with 10 -- 100 developers one public repo with 100 or 1000 branches makes sense. Concretely, both approaches have a problem:

- if the developer must delete his branch, many won't do it.
- developers might not want to have their branches deleted, because they want to link to them in discussions or keep them as starting points for further work and branches.

#7 - 12/05/2012 01:08 PM - Deil Christoph

Jürgen Knödlseider wrote:

Great, champagne for your first commit !

Hey, you stole my first commit :-)

<https://cta-redmine.irap.omp.eu/projects/gammlib/repository/revisions/a5b7c48c67d41785e292a790a69e1afb0ed7441b>

#8 - 12/05/2012 01:12 PM - Deil Christoph

Jürgen Knödlseider wrote:

Christoph Deil wrote:

Pull request: Add .gitignore file to ctools:

<https://cta-redmine.irap.omp.eu/projects/ctools/repository/revisions/f2690d66fbb0416696cbfdf53a8da4e60c1c3326>

Could you put this one in the ctools project?

I see you merged it already into ctools devel, so I assume there's no need for me to open a ctools "pull request" issue.

#9 - 12/05/2012 01:55 PM - Knödlseider Jürgen

Christoph Deil wrote:

Jürgen Knödlseider wrote:

Great, champagne for your first commit !

Hey, you stole my first commit :-)

<https://cta-redmine.irap.omp.eu/projects/gammlib/repository/revisions/a5b7c48c67d41785e292a790a69e1afb0ed7441b>

Was there any alternative method to do the merge that keeps your name on the commit?

#10 - 12/05/2012 01:59 PM - Deil Christoph

Jürgen Knödseder wrote:

Christoph Deil wrote:

Jürgen Knödseder wrote:

Great, champagne for your first commit !

Hey, you stole my first commit :-)

<https://cta-redmine.irap.omp.eu/projects/gammalib/repository/revisions/a5b7c48c67d41785e292a790a69e1afb0ed7441b>

Was there any alternative method to do the merge that keeps your name on the commit?

What did you do?

Actually my commit is gone and there's a new one with you as author.

Typically you merge my branch and then there's my commits (one in this case) and then a merge commit from you.

#11 - 12/05/2012 02:02 PM - Deil Christoph

- Status changed from Feedback to Closed

- Assigned To set to Deil Christoph

Christoph Deil wrote:

Jürgen Knödseder wrote:

Christoph Deil wrote:

Jürgen Knödseder wrote:

Great, champagne for your first commit !

Hey, you stole my first commit :-)

<https://cta-redmine.irap.omp.eu/projects/gammalib/repository/revisions/a5b7c48c67d41785e292a790a69e1afb0ed7441b>

Was there any alternative method to do the merge that keeps your name on the commit?

What did you do?

Actually my commit is gone and there's a new one with you as author.

Typically you merge my branch and then there's my commits (one in this case) and then a merge commit from you.

I suggest to not rewrite history in this case, i.e. not try to fix the author for that one commit.

I promise to make more commits this week. :-)

#12 - 12/05/2012 02:05 PM - Deil Christoph

This is a nice visualisation, you can see the commits from developer branches are unchanged when merged into `master` after pull request review:

<https://github.com/astropy/astropy/network>

<https://github.com/astropy/astropy/commits/master>

#13 - 12/05/2012 02:06 PM - Knödlseider Jürgen

Christoph Deil wrote:

I don't think it's a good idea to have people like me and other new developers push to devel. Once a commit is in devel it's public and we really shouldn't remove it. Also I like to have a new branch for every little feature, I don't want to work in one hess branch, and I also don't want a central repo with hundreds of branches from all the developers.

The only point with the many branches is that they can't be put easily in the continuous integration system (we would not to generate a job for each branch, and also the workload on the CI server would increase). So having many feature branches would mean that we don't really control them, and CI is only applied once we merge.

I love the github workflow with forks and code review via pull requests. Check this out:

<http://docs.astropy.org/en/latest/#developer-documentation>

I can see the problem that there are no free tools as nice as github you could install at IRAP and you don't want to issues / code review / project history be split between github or bitbucket and IRAP, so at the moment I don't have concrete suggestion, but just the general comment that I think doing more on github and fully open source would be great.

Ok, let's just learn by doing and see how things go.

In the end this needs to be discussed on a CTA developer mailing list, there was a thread once, but I don't think anything was decided or concrete plans were made.

For CTA, svn was selected as repository, hence here the workflow may be different. I think we should keep things separated for the moment from the CTA mailing list, as we have so far no clear agreement on the development philosophy and respective roles.

#14 - 12/05/2012 02:08 PM - Knödlseider Jürgen

Christoph Deil wrote:

Christoph Deil wrote:

Jürgen Knödlseider wrote:

Christoph Deil wrote:

Jürgen Knödlseider wrote:

Great, champagne for your first commit !

Hey, you stole my first commit :-)

<https://cta-redmine.irap.omp.eu/projects/gammalib/repository/revisions/a5b7c48c67d41785e292a790a69e1afb0ed7441b>

Was there any alternative method to do the merge that keeps your name on the commit?

What did you do?

Actually my commit is gone and there's a new one with you as author.

Typically you merge my branch and then there's my commits (one in this case) and then a merge commit from you.

I used SmartGit for the merge, so maybe this is a feature of this tool (I in fact cherry picked a single commit instead of merging a branch).

I suggest to not rewrite history in this case, i.e. not try to fix the author for that one commit.

I promise to make more commits this week. :-)

Jürgen Knödseder wrote:

Christoph Deil wrote:

I don't think it's a good idea to have people like me and other new developers push to devel. Once a commit is in devel it's public and we really shouldn't remove it. Also I like to have a new branch for every little feature, I don't want to work in one hess branch, and I also don't want a central repo with hundreds of branches from all the developers.

The only point with the many branches is that they can't be put easily in the continuous integration system (we would not to generate a job for each branch, and also the workload on the CI server would increase). So having many feature branches would mean that we don't really control them, and CI is only applied once we merge.

What is needed is a way to run tests for pull requests, but not all branches.

<https://github.com/astropy/astropy/wiki/Continuous%20Integration>

On github most projects do this with travis-ci, you could also use a staging branch to run tests before you merge.

In the end this needs to be discussed on a CTA developer mailing list, there was a thread once, but I don't think anything was decided or concrete plans were made.

For CTA, svn was selected as repository, hence here the workflow may be different. I think we should keep things separated for the moment from the CTA mailing list, as we have so far no clear agreement on the development philosophy and respective roles.

OK, I didn't know that.

Who / when selected SVN?

Was this discussed and announced on a mailing list?

Where is the repo?

(sorry, I haven't been keeping up with CTA software in the last months, but now want to start contributing)

#16 - 12/05/2012 02:19 PM - Deil Christoph

Jürgen Knödlseider wrote:

I used SmartGit for the merge, so maybe this is a feature of this tool (I in fact cherry picked a single commit instead of merging a branch).

I think you should simply merge and get a merge commit.
This is a simple way to see who did the code review.

#17 - 12/05/2012 02:53 PM - Knödlseider Jürgen

Christoph Deil wrote:

Jürgen Knödlseider wrote:

Christoph Deil wrote:

I don't think it's a good idea to have people like me and other new developers push to devel. Once a commit is in devel it's public and we really shouldn't remove it. Also I like to have a new branch for every little feature, I don't want to work in one hess branch, and I also don't want a central repo with hundreds of branches from all the developers.

The only point with the many branches is that they can't be put easily in the continuous integration system (we would not to generate a job for each branch, and also the workload on the CI server would increase). So having many feature branches would mean that we don't really control them, and CI is only applied once we merge.

What is needed is a way to run tests for pull requests, but not all branches.
<https://github.com/astropy/astropy/wiki/Continuous%20Integration>

On github most projects do this with travis-ci, you could also use a staging branch to run tests before you merge.

In the end this needs to be discussed on a CTA developer mailing list, there was a thread once, but I don't think anything was decided or concrete plans were made.

For CTA, svn was selected as repository, hence here the workflow may be different. I think we should keep things separated for the moment from the CTA mailing list, as we have so far no clear agreement on the development philosophy and respective roles.

OK, I didn't know that.
Who / when selected SVN?

Selection done basically by the DATA workpackage upon request of the Project Office. You may check the presentation of Nadine Neyroud at the last CTA meeting in Rome.

Was this discussed and announced on a mailing list?

I guess you find traces on the DM mailing list, but information flow is still very bad in CTA.

Where is the repo?

At CC-Lyon (the IN2P3 data centre in France).

(sorry, I haven't been keeping up with CTA software in the last months, but now want to start contributing)

#18 - 12/05/2012 02:54 PM - Knödlseider Jürgen

Christoph Deil wrote:

Jürgen Knödlseider wrote:

I used SmartGit for the merge, so maybe this is a feature of this tool (I in fact cherry picked a single commit instead of merging a branch).

I think you should simply merge and get a merge commit.
This is a simple way to see who did the code review.

In this case I guess that you first pull in changes from the devel branch to solve possible conflicts in the feature branch?

#19 - 12/05/2012 03:05 PM - Deil Christoph

Jürgen Knödlseider wrote:

Christoph Deil wrote:

Jürgen Knödlseider wrote:

I used SmartGit for the merge, so maybe this is a feature of this tool (I in fact cherry picked a single commit instead of merging a branch).

I think you should simply merge and get a merge commit.
This is a simple way to see who did the code review.

In this case I guess that you first pull in changes from the devel branch to solve possible conflicts in the feature branch?

Yes, the developer typically rebases on master so that the reviewer can always merge without conflicts:
http://docs.astropy.org/en/latest/development/workflow/development_workflow.html#rebasing-on-trunk

#20 - 12/05/2012 03:06 PM - Deil Christoph

If you want to play with the github workflow, make a free account and we'll play a bit here:
<https://github.com/cdeil/gammalib>

(it's a private repo, I pay 7 dollars per month to github. I made it private because I wasn't sure you'd want a public gammalib github repo, although since it's open source, I'd be allowed to make it public I think)

#21 - 12/05/2012 03:56 PM - Knödlseider Jürgen

Christoph Deil wrote:

Yes, the developer typically rebases on master so that the reviewer can always merge without conflicts:
http://docs.astropy.org/en/latest/development/workflow/development_workflow.html#rebasing-on-trunk

Thanks for the hint. I never used this option so far, so I have to look more seriously into it. By the way, in our case we should rebase on devel.

#22 - 12/05/2012 06:26 PM - Deil Christoph

Jürgen Knödlseider wrote:

I used SmartGit for the merge, so maybe this is a feature of this tool (I in fact cherry picked a single commit instead of merging a branch).

I get this error which I've never seen before:

```
ChristophMacbook:gammalib deil$ git fetch
From github.com:cdeil/gammalib
+ a30be6a...a3e90cb devel -> devel (forced update)
+ 72740b5...a3e90cb origin/HEAD -> origin/HEAD (forced update)
+ 6042f79...9bf3872 origin/comptel -> origin/comptel (forced update)
error: Ref refs/remotes/origin/devel is at a3e90cba1bf837edf3bf4a1eedc22b79b635c6c6 but expected
72740b59edc3b1859a31e62758b756b8610bf6df
! 72740b5...a3e90cb origin/devel -> origin/devel (unable to update local ref)
+ f37e185...8624151 origin/hess -> origin/hess (forced update)
```

I can look later today what happened.

#23 - 12/05/2012 09:03 PM - Knödlseider Jürgen

I made some changes to .gitignore since I have additional files in the way, maybe that's what is causing the problems in the comptel branch.

#24 - 12/05/2012 09:12 PM - Deil Christoph

Jürgen Knödlseider wrote:

I made some changes to .gitignore since I have additional files in the way, maybe that's what is causing the problems in the comptel branch.

Did you do git push --force?

#25 - 12/05/2012 09:17 PM - Knödlseider Jürgen

No sure. As mentioned above, I use SmartGit for doing most of my transactions, but I think it does not forced push by default. But I'm not really sure about this.

#26 - 12/05/2012 09:24 PM - Deil Christoph

Jürgen Knödlseider wrote:

No sure. As mentioned above, I use SmartGit for doing most of my transactions, but I think it does not forced push by default. But I'm not really sure about this.

Do you have a SmartGit's log.txt ?

<http://smartgit.3668570.n2.nabble.com/GIT-commands-sent-by-SmartGit-td4113010.html>

My guess is that you rewrote history and force pushed, which should never happen to branches that others have checked out.

#27 - 12/05/2012 09:32 PM - Knödlseider Jürgen

I learned again something: SmartGit has a log file!

Here the push commands from today:

```
1026043431 (2012-12-05 12:44:03,942) [WorkerThread-1] INFO smartgit.core.executable - Executing following command: /usr/local/git/bin/git push --recurse-submodules=check --porcelain origin refs/heads/comptel:refs/heads/comptel
1026066826 (2012-12-05 12:44:27,337) [WorkerThread-1] INFO smartgit.core.executable - Executing following command: /usr/local/git/bin/git push --recurse-submodules=check --porcelain origin refs/heads/hess:refs/heads/hess
1026327556 (2012-12-05 12:48:48,067) [WorkerThread-9] INFO smartgit.core.executable - Executing following command: /usr/local/git/bin/git push --recurse-submodules=check --porcelain origin refs/heads/devel:refs/heads/devel
1026349720 (2012-12-05 12:49:10,231) [WorkerThread-9] INFO smartgit.core.executable - Executing following command: /usr/local/git/bin/git push --recurse-submodules=check --porcelain origin refs/heads/hess:refs/heads/hess
1026582079 (2012-12-05 12:53:02,590) [WorkerThread-1] INFO smartgit.core.executable - Executing following command: /usr/local/git/bin/git push --recurse-submodules=check --porcelain origin refs/heads/comptel:refs/heads/comptel
```

I have no idea what --porcelain is, but I cant' see --force.

#28 - 12/05/2012 09:43 PM - Deil Christoph

Jürgen Knödlseider wrote:

I learned again something: SmartGit has a log file!

Here the push commands from today:

[...]

I have no idea what --porcelain is, but I cant' see --force.

Both --porcelain and --recurse-submodules=check are irrelevant:

\$ git help push

...

--porcelain

Produce machine-readable output. The output status line for each ref will be tab-separated and sent to stdout instead of stderr.

The full symbolic names of the refs will be given.

--recurse-submodules=check|on-demand

Make sure all submodule commits used by the revisions to be pushed are available on a remote-tracking branch. If check is used git will verify that all submodule commits that changed in the revisions to be pushed are available on at least one remote of the submodule. If any commits are missing the push will be aborted and exit with non-zero status. If on-demand is used all submodules that changed in the revisions to be pushed will be pushed. If on-demand was not able to push all necessary revisions it will also be aborted and exit with non-zero status.

...

I just did a git fetch from another machine where I had gammalib cloned yesterday, no problem there.

So apparently the problem only showed up in the clone where I made the original commit.

Maybe something went wrong when you cherry-picked, then, because somehow you managed to get the content from my commit, but my commit is not in the history any more.

Let's forget about it for now, we can always track down the problem should this happen again.