GammaLib - Change request #617

Follow PEP8 in Python code

12/05/2012 08:51 PM - Deil Christoph

Status:	Closed	**Start date:**	12/05/2012
Priority:	Low	**Due date:**	
Assigned To:	Deil Christoph	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

There's a style guide for Python code:
http://www.python.org/dev/peps/pep-0008/

I think gammalib / ctools, actually every Python project, should follow it.
The only real issue with gammalib / ctools Python code is that it mixes spaces and tabs, which has cost me quite some time in the past, chasing cryptic and hard to understand errors.
PEP8 says all Python code should use 4 spaces for indentation:

```
$ pep8 ./test/test_GNumerics.py
./test/test_GNumerics.py:29:1: W191 indentation contains tabs
./test/test_GNumerics.py:30:1: W191 indentation contains tabs
```

It's easy to check with Jenkins be
http://pypi.python.org/pypi/pep8
https://wiki.jenkins-ci.org/display/JENKINS/Violations

and it's also easy to clean up:
https://github.com/hhatto/autopep8

Jürgen, if you agree I can fix all PEP8 and make a pull request tomorrow.

History

#1 - 12/05/2012 09:13 PM - Knödlseder Jürgen

- % Done changed from 0 to 10

In principle, I always use a 4 space Tab for writing the Python code, but there can be mixes with the old code.

I was not aware of the Python style guide, thanks for pointing me to the reference. I guess we should add this to the Coding Conventions. We can should add a chapter "2.2 Python rules" to the document.

Can you go ahead and transform the code and also add a chapter about Python coding rules?

I created feature #618 that asks for inclusion of the PEP8 rules in Jenkins.

Do you know if there are automatic tools to check C++ coding rules?

#2 - 12/05/2012 09:22 PM - Deil Christoph

Jürgen Knödlseder wrote:

> In principle, I always use a 4 space Tab for writing the Python code, but there can be mixes with the old code.
>
> I was not aware of the Python style guide, thanks for pointing me to the reference. I guess we should add this to the Coding Conventions. We can should add a chapter "2.2 Python rules" to the document.
>
> Can you go ahead and transform the code and also add a chapter about Python coding rules?

Will do tomorrow.
Merging it into your several major branches might not be trivial, I'll simply branch devel and let's see if everything works well.

> I created feature #618 that asks for inclusion of the PEP8 rules in Jenkins.

Thanks

> Do you know if there are automatic tools to check C++ coding rules?

There must be, I added this to #559

#3 - 12/05/2012 09:25 PM - Knödlseder Jürgen

Christoph Deil wrote:

> Jürgen Knödlseder wrote:
>
> > In principle, I always use a 4 space Tab for writing the Python code, but there can be mixes with the old code.
> >
> > I was not aware of the Python style guide, thanks for pointing me to the reference. I guess we should add this to the Coding Conventions. We can should add a chapter "2.2 Python rules" to the document.
> >
> > Can you go ahead and transform the code and also add a chapter about Python coding rules?
>
> Will do tomorrow.
> Merging it into your several major branches might not be trivial, I'll simply branch devel and let's see if everything works well.

Right, the idea was that you should always base new code on devel, and also merge back into devel. Actually, hess and comptel are pretty

synchronized, comptel will soon disappear as I'm about to finish a first COMPTEL interface (I'm looking forward to see the first Crab spectrum from 750 keV to 20 TeV obtained with GammaLib - maybe still tonight smile.png

> I created feature #618 that asks for inclusion of the PEP8 rules in Jenkins.

Thanks

> Do you know if there are automatic tools to check C++ coding rules?

There must be, I added this to #559

Great.

#4 - 12/06/2012 09:37 AM - Deil Christoph

Which Python versions should gammalib support?
My suggestion would be Python 2.6, 2.7, 3.2 and 3.3.

I can put this info in the coding guidelines and the install instructions.
Jenkins only runs 2.7 and 3.2 at the moment, could it be changed to run all supported versions?
One architecture / compiler should be enough here.

#5 - 12/06/2012 12:43 PM - Knödlseder Jürgen

Python 2.6 is pretty late, I still have systems running Python 2.4! I would simply say Python 2 and 3, although I'm not sure that we have to go before 2.4 ...

For the moment I'm only testing 2.73 and 3.23 (see https://cta-jenkins.irap.omp.eu/view/CTA%20Python/job/gammalib-build-python/), but this list can be extended (just have to compile the other Python versions on the test machines).

The reason why I use several architectures / compilers is related to swig. The Python interface is built with swig, which implies compilation of wrapper code. I want to make sure that this works nice with different compilers and architectures.

By the way: your new check seems to fail on Python 3 (see https://cta-jenkins.irap.omp.eu/view/CTA%20Python/job/gammalib-check-python/):

basic_string::_S_construct null not valid

I opened bug #624 and assigned it to you (pull from devel before working on the code as I did some editing of the test_GSky.py file).

#6 - 12/06/2012 06:33 PM - Deil Christoph

Pull request: pep8 branch
https://cta-redmine.irap.omp.eu/projects/gammalib/repository/show?rev=pep8

Note that this branch has to be rebased on devel before it can be merged.
So far I haven't managed to rebase (because of conflicts with the COMPTEL changes), but I'll keep trying ...

I ran these commands:

```
autopep8 --ignore=E501 --in-place pyext/check_config.py pyext/setup.py dev/replace_license.py
autopep8 --ignore=E501 --in-place --recursive inst
autopep8 --ignore=E501 --in-place --recursive test
```

Then I fixed these three things by hand which autopep8 didn't fix:

```
$ pep8 . --ignore=E501 --exclude=pyext
./inst/lat/test/binned_integration.py:108:21: E127 continuation line over-indented for visual indent
./test/test_GSky.py:93:16: E221 multiple spaces before operator
./test/test_GSky.py:107:16: E221 multiple spaces before operator
```

There was an IndentationError in test/test_file_function.py. Should this test be run?

Finally I added a short Section to the CDC document on Python versions and style.

#7 - 12/07/2012 11:23 PM - Knödlseder Jürgen

- Status changed from Feedback to Closed

- % Done changed from 10 to 100

Christoph Deil wrote:

> Pull request: pep8 branch
> https://cta-redmine.irap.omp.eu/projects/gammalib/repository/show?rev=pep8
>
> Note that this branch has to be rebased on devel before it can be merged.
> So far I haven't managed to rebase (because of conflicts with the COMPTEL changes), but I'll keep trying ...

Rebased on devel and merged into devel.

> I ran these commands:
> [...]
>
> Then I fixed these three things by hand which autopep8 didn't fix:
> [...]
>
> There was an IndentationError in test/test_file_function.py. Should this test be run?

It's not part of the unit test, but I used this to test the file function code. All files shipped with GammaLib should work, so it's okay that you corrected this.

> Finally I added a short Section to the CDC document on Python versions and style.

Great. Thanks a lot.

So I propose to close that one as the addition of pep8 to Jenkins is in feature #618.