

Table of Contents

Information for PanDA Pilot developers.....	1
How to fork the pilot GitHub repository and work on local files.....	1
How to add upstream.....	1
How to update pilot.....	1
How to create a pull request.....	2
How to sync my private fork with the original repository.....	2
Cut-and-paste example (Pilot 1).....	2
Cut-and-paste example (Pilot 2).....	2
How to verify code before a pull request (Pilot 2).....	2
Cut-and-paste example of how to run test suites.....	3
How to download certain pilot version using tag.....	3
How to test pilot code.....	3
Testing on the grid.....	3
How to send test jobs.....	4

Information for PanDA Pilot developers

This wiki will explain how to get started with pilot development.

How to fork the pilot GitHub repository and work on local files

All pilot code should be kept in GitHub. A new developer can start by forking the PanDA Pilot repository, i.e. go to the [Pilot 1](#) or [Pilot 2](#) repository web pages and click on the 'fork' link on the top right and follow the instructions (you need to have a GitHub account).

Download the repository into your local development area, cd into it and checkout the main-dev (Pilot 1) or next (Pilot 2) branch:

```
git clone https://github.com/your_github_username/pilot.git
cd pilot
git checkout main-dev
```

or

```
git clone https://github.com/your_github_username/pilot2.git
cd pilot2
git checkout next
```

How to add upstream

(Replace "pilot" with "pilot2" for Pilot 2)

```
git remote add upstream git@github.com:PanDAWMS/pilot.git
git remote set-url --push upstream git@xxxxx:PanDAWMS/pilot.git
```

```
#git remote -v
origin  ssh://git@github.com/your_name/pilot.git (fetch)
origin  ssh://git@github.com/yourname/pilot.git (push)
upstream      git@github.com/PanDAWMS/pilot.git (fetch)
upstream      git@xxxxx:PanDAWMS/pilot.git (push)
```

How to update pilot

Edit the files you want to work on, commit every change (frequently), i.e.

```
git commit -a -m 'some comment..'
```

and when you are done with all changes, test the code before creating the pull request (instructions will be written at some point). Please make detailed comments for documentation so other people can understand what the update is about (in the pull request).

You will need to synchronize with upstream (i.e. the main repo) frequently, so you need to checkout upstream then rebase on it or merge with it.

You must not push to upstream. You should only push to your own forked repo then you can create pull request.

```
git push [-u] origin branch_name
```

How to create a pull request

After you have committed and tested your changes in your private GitHub fork, go to the main GitHub page in your private fork and click on 'New pull request'. Select the 'compare across forks' option (base fork should be PandaWMS/pilot[2] and head fork should point to your repository), write some comments and then click on 'Create pull request'.

This will send a message to the repository manager (currently Paul) who will review and approve the changes. In Pilot 2, the manager will issue the test suite (which runs pep8, flake8 and unit tests) before any changes will be approved.

NEVER COMMIT TO ANY MASTER BRANCH.

How to sync my private fork with the original repository

Ok so you have been developing in your own fork for a while, but in the meantime the original repository will probably have changed a bit. Sync your private fork with the original repository by following these instructions[?]. For pilot development, replace *master* with *main-dev* (Pilot 1) or *next* (Pilot 2) in the instructions.

Note: when you want to continue making local commits to your own fork, you need to set the following:

```
git push --set-upstream origin main-dev
```

which will push the current branch and set the remote as upstream (or you will get an error message if you just do a 'git push').

Cut-and-paste example (Pilot 1)

Starting with a fresh checkout of code from the private repository belonging to USERNAME.

```
git clone https://github.com/USERNAME/pilot.git
cd pilot
git checkout main-dev
git remote -v
git remote add upstream https://github.com/PanDAWMS/pilot.git
git fetch upstream
git merge upstream/main-dev
```

Cut-and-paste example (Pilot 2)

```
git clone https://github.com/USERNAME/pilot2.git
cd pilot2
git checkout next
git remote -v
git remote add upstream https://github.com/PanDAWMS/pilot2.git
git fetch upstream
git merge upstream/next
```

How to verify code before a pull request (Pilot 2)

For Pilot 2 we enforce PEP8 and Flake8 coding standards, A pull request will trigger an automated test which also runs unit tests. It can be practical to do these tests locally before making any pull request as it can take several iterations to get things right. For running these test suites, it is also practical to use virtualenv (available on lxplus - but some lxplus nodes have problems, in case of strange errors, just switch to another

node) since you might not have root privileges to install additional software.

Cut-and-paste example of how to run test suites

```
cd [somewhere]
mkdir test
virtualenv test
cd test
source bin/activate
pip install 'unittest2'
pip install "flake8${FLAKE8_VERSION}"
pip install 'pep8-naming'
pip install 'flake8-blind-except'

cp -r [someplace]/pilot2 .
cd pilot2
flake8 --config .flake8 pilot.py pilot/
pycodestyle --ignore E501 pilot.py pilot/
unit2 -v
```

How to download certain pilot version using tag

First of all, see the list of tags in [GitHub](#). If you want to clone the 2.2.1 tag, do

```
git clone -b '2.2.1' --single-branch --depth 1 https://github.com/PanDAWMS/pilot2.git cd pilot2
```

How to test pilot code

Testing on the grid

1. Login to a condor-submit host.
2. Create a submit.jdl script. Use the following as a start and update for the relevant queue.

```
# Condor-G glidein pilot for panda
Dir                = *** path to batch system stdout/err *** / *** site name *** /
Dir2               = *** path to testing directory where this idl is located *** / *** site name ***
notify_user       = *** your email ***
grid_resource      = gt5 gk04.swt2.uta.edu/jobmanager-pbs *** <- an example ***
+MATCH_APF_QUEUE  = " *** site name *** "
executable         = $(Dir2)/pilot2-wrapper.sh
arguments          = -w generic -a /scratch -j ptest -q *** queue name *** -r *** resource name ***
transfer_input_files = $(Dir2)/pilot2-wrapper.sh
globusrs1         = (jobtype=single)(queue=paul_test_q) *** <- an example ***
+RACF_Group       = "dq2test"
universe           = grid
output             = $(Dir)/pilot-*** site name ***-0001.output
error              = $(Dir)/pilot-*** site name ***-0001.error
log                = $(Dir)/pilot-*** site name ***-0001.log
stream_output      = False
stream_error       = False
notification       = Error
transfer_executable = True
should_transfer_files = YES
when_to_transfer_output = ON_EXIT_OR_EVICT
periodic_hold      = GlobusResourceUnavailableTime != UNDEFINED && (CurrentTime-GlobusResourceUnava
periodic_remove    = (JobStatus == 5 && (CurrentTime - EnteredCurrentStatus) > 3600) || (JobStatus =
+Nonessential      = True
copy_to_spool      = false
x509userproxy      = *** path to your user proxy ***
queue              1
```

Tip: for any other queue, have a look at the idl script located in the batch stdout directory which is reachable from any job page in the PanDA monitor. I.e. follow a stdout link in the job information section, remove the job stdout filename (e.g. 6870628.2.out) from the URL at the top of your browser to get the contents of the stdout directory, then look at the bottom of the long list and you will find the submit.jdl script. Modify your idl accordingly.

3. Download a local copy of the Pilot 2 wrapper, e.g. from [here](#), and store it in the same directory as the submit.jdl script.

4. Submit with

```
condor_submit submit.jdl
```

and follow the progress in the output directory defined in the jdl.

More (but older) examples can be found in the RunPilot section.

How to send test jobs

(To be written; for now, ask Paul for a test job script).

-- PaulNilsson - 2016-09-07

This topic: PanDA > DeveloperInfo

Topic revision: r13 - 2019-11-15 - PaulNilsson



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)