

## AutoPyfactory (a.k.a APF)

### Current version:

Current version used in production at BNL factories is 2.4.1, installed from RACF repository:

[http://dev.racf.bnl.gov/yum/grid/testing/rhel/6Workstation/x86\\_64/](http://dev.racf.bnl.gov/yum/grid/testing/rhel/6Workstation/x86_64/)

It is also in the OSG development repo:

<https://koji-hub.batlab.org/koji/packageinfo?packageID=374>

built as a single RPM.

This version installs the configuration files under the documentation directory, with filename ending with -example:

```
ll /usr/share/doc/autopyfactory-2.4.1/*example
-rw-r--r-- 1 root root 6184 Oct 8 2014 /usr/share/doc/autopyfactory-2.4.1/autopyfactory.conf-example
-rw-r--r-- 1 root root 1082 Oct 8 2014 /usr/share/doc/autopyfactory-2.4.1/mappings.conf-example
-rw-r--r-- 1 root root 732 Oct 8 2014 /usr/share/doc/autopyfactory-2.4.1/monitor.conf-example
-rw-r--r-- 1 root root 5625 Oct 8 2014 /usr/share/doc/autopyfactory-2.4.1/proxy.conf-example
-rw-r--r-- 1 root root 47105 Oct 8 2014 /usr/share/doc/autopyfactory-2.4.1/queues.conf-example
```

Same for the logrotate config files and sysconfig config files.

### Next steps:

Not everything equally important. And not necessarily everything to be included in the next release.

- Split the RPM into multiple packages. Still to be decided how many, and the content of each one of them.
- Install the configuration files directly into /etc/autopyfactory directory. Users don't feel comfortable with the idea of copying them from the doc directory.
- Re-factorize the way the configuration directory queues.d/ is being used. It should be the UNIX standard behavior, which is not right now.
- Make the code reading the configuration another plugin category.
- Re-factorize code to allow more than one internal queues to cluster and work together.
- Introduce the HTCondor python bindings.
- Improve the documentation.

From now on, we commit to release backward compatible versions. When not possible, the name of the package will change (i.e. autopyfactory3).

### Wrappers and glideins:

Wrappers are not exactly part of AutoPyFactory, but they come together, as they are the executable being submitted to the sites via condor-g.

For ATLAS there are currently 2 wrappers: one used in OSG and one for the rest of the world. The latter one is a single file written in bash. The OSG one is composed by a first part in bash and a series of plugins and core code in python. The OSG one is currently under re-factoring, with the ultimate goal of merging both wrappers.

In this plugin-based architecture, the intention is to have another plugin being a glidein. There is no stand-alone glidein code provided by HTCondor team. Several people have created their own, usually based on what BOSCO does. We also started a new one, written in python. And the HTCondor team is working on a new stand-alone one based on the master daemon. No ETA. Our current plan is to have a look to the existing ad-hoc ones and see what they have in common and try to write a more generic one.

**Customers:**

Currently being used by ATLAS and AMS, both VOs based on PanDA.  
Our next potential customers are:

- GLOW VO. For that we need a glidein that can be submit by the factory.
- ICECUBE VO. Plan is to start by reusing their glidein.