HTCondor’s Python API – The Python Bindings

Jason Patton
Center for High Throughput Computing
Tutorial – Load this now!

› Head over to the HTCondor documentation
   https://htcondor.readthedocs.io
› Application Programming Interfaces
   > Python Bindings
   > HTCondor Python Bindings Tutorials
   > Launch Binder
HTCondor Clients in 2013

Command Line Clients
- Fully Featured!
- Requires fork/exec and process handling
- Outputs in multiple formats

SOAP Clients
- Features! (Some)
- Caveats with respect to scalability, security.

Something Missing In The Middle
Command Line Clients
Fully Featured!
Requires fork/exec and process handling
Outputs in multiple formats

SOAP Clients
Some Features! (Some)
Caveats with respect to scalability, security.

RESTful clients, workflow managers, monitoring tools
Features! (Some)
Caveats with respect to scalability, security.
**Design Philosophy**

**ClassAds**: Everything based on ClassAds; make these the “core” of the bindings.

**Pythonic**: Semantics and APIs should feel natural to a Python programmer.

- Use iterators, exceptions, guards.
- ClassAds behave as much like a dict as reasonable.

**Backward compatible**: APIs are here to stay for as long as possible.

- When we absolutely must, use standard Python DeprecationWarning techniques.

- Yes, this means that we keep even design warts for far longer than we'd like!

**Native code**: Call same HTCondor library code as CLI; identical in performance.

**Complete**: If you can do it with the command line tools, you should be able to do it with Python.
Since *pythonic* is in our design philosophy, the education tools should use the tools favored by the Python community:

- **Sphinx**-based documentation. Hosted on ReadTheDocs; looks / feels / smells like Python documentation.

- **Jupyter**-based tutorials. Use Binder.org service to spawn a Docker container with a private HTCondor instance (or use Docker locally). Interact via your browser.
You can help!

The contents of the tutorials and documentation are kept on GitHub:

› https://github.com/htcondor/htcondor-python-bindings-tutorials

› JupyterLab & Binder integration developed by Josh Karpel.

› Find a bug? Spot some missing content?
  • Send a pull request; Travis-CI will test and update the static content once merged.
Installing Python Bindings

› On Linux
  • Included in the RPM and DEB packages for the system Python(s)
  • Available via PyPI on Linux:
    • `pip install htcondor`
  • Available via Anaconda (via conda-forge channel) on Linux and macOS:
    • `conda install -c conda-forge python-htcondor`

› On Windows
  • Included in the Windows MSI
    • 9.0.x LTS Releases and 9.8.1- Feature Releases: Python 2.7 and Python 3.6
    • 9.9.0+ Feature Releases: Python 3.9

› On MacOS
  • Included in the HTCondor tarball for the system Python
  • Available via Anaconda, see above
Python API – Big Changes

› No major changes since last year
› Upcoming deprecations
  • User-controllable Schedd transactions
    with schedd.transaction() as txn: ...
    not deprecated yet but will be soon (HTCondor 10.1?), check the release notes and docs closely!
  • Use schedd.submit(my_submit_obj) now!
› Keep track of changes in the release notes
› Head over to the HTCondor documentation
   https://htcondor.readthedocs.io

› Application Programming Interfaces
  > Python Bindings
  > HTCondor Python Bindings Tutorials
  > Launch Binder
TIME FOR TUTORIAL DEMO
Questions?

› Python bindings API developers and users alike are active on the HTCondor Users mailing list
Follow us on Twitter! https://twitter.com/HTCondor

This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATH Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.