

# Python Bindings Version 2

Presented by: Cole Bollig

Written by: Todd L Miller

2024-07-11

**CHTC**

**HTC**Condor  
Software Suite

**PATH**

# Talk Outline

- Why a New Version?
- Why Tell You Now?
- What Should You Do?
- What's Changed?

# Why a New Version?

- The version 1 bindings were written using Boost.Python, which hasn't been updated for a while (at least two years) and has required increasingly exotic hacks to actually use.
  - We wanted to avoid suddenly being unable to provide new builds of the bindings, and to provide a smoother transition to users of the bindings.
- This gives us a good opportunity for improvements
  - some of which we'd avoided because they were breaking changes
  - clean up memory management (separate ClassAds and ExprTrees)
  - remove unfixable code (xquery()) and the Transaction object)
  - more maintainable/understandable/extensible code
  - **Python minor version independence**

# Minor Version Independence

- Currently, when you install HTCondor, you get a version of the Python bindings that only work with a specific minor version of Python, e.g., 3.6 on CentOS 7 or 3.12 on Ubuntu 24.
  - This is a pain if your preferred version of Python is different.
- If you HTCondor from pip, you can choose from bindings that work with Python 3.8, 3.9, ..., up to 3.12.
  - CHTC still needs to do a release for each Python version, so you might not be able to use the version you like.
- The version 2 bindings are built so that they work with any version of Python 3 (more recent than 3.2).

# Why Tell You Now? (part 1)

- The new bindings will be available in HTCSS 24
  - (The old bindings will still be there.)
- To give you a chance to test your code before we switch over to the new bindings, hopefully in HTCSS 25 (August 2025).
  - (The old bindings may still be there.)
- To give us a chance to react to what you find testing your code, particularly with respect to API tweaks.

# Why Tell You Now? (part 2)

- The new API is very similar to the old one; we expect that most code using documented and un-deprecated version 1 APIs to mostly just work when run with version 2.
- We'll fix bugs in the new implementation in the 24.0 stable releases.
- We'll make changes to the new API in the 24.x feature series.
  - Those changes will be driven by your feedback.
  - Those changes will not be delayed by deprecation.
  - The new documentation, although it should be more comprehensive and more precise, is not yet considered normative. We don't expect to have to change the new parts of the version 2 API, but we reserve the right to.

# What You Should Do

1. Make the following change...

```
import htcondor2 as htcondor
import classad2 as classad
```

2. ... and then test! Check for `eval()/simplify()` special case.
3. Consult the [migration guide](https://htcondor.readthedocs.io/en/main/apis/python-bindings/api/version2/migration-guide.html).  
<https://htcondor.readthedocs.io/en/main/apis/python-bindings/api/version2/migration-guide.html>
4. And then tell us what happened! Please!  
(send mail to `htcondor-admin@cs.wisc.edu`)
5. If you had problems, please upgrade and test again.

# Separated ClassAds and ExprTrees

- An ExprTree object no longer records which ClassAd object it came from, if any. Therefore:
- If you call `eval()` or `simplify()` and don't specify a context ad, the default in version 2 is no context at all, not the ExprTree's original ad. This may cause you to get different results!
- Check for `eval()` or `simplify()` being called in your code without any arguments!



# Job Submission

- No more transaction objects.
- No more submitting raw ClassAds.
- The `Submit` object no longer tries to translate from Python objects into strings on your behalf.
  - (The submit language, as distinct from the ClassAd language, is untyped.)
- The `Submit` object now supports all valid submit files, and the `Schedd.submit()` method now respects queue statements in the `Submit` object unless you specify otherwise.
- `Submit.issueCredentials()` duplicates *condor\_submit's* credential handling.

# Job Submission Example

```
import htcondor2

submit = htcondor2.Submit("""
    executable = my_prog
    arguments = data_file $(SCENARIO) --seed $(SEED)
    transfer_input_files = data_file $(SCENARIO)
    transfer_output_files = $(SCENARIO).result
    request_cpus = 1
    request_memory = 4096
    out = $(SCENARIO).out
    err = $(SCENARIO).err
    log = my_prog.log

    queue 1 SCENARIO matching file scenarios/*
""")
submit["SEED"] = "0xDEADBEEF"

schedd = htcondor2.Schedd()
result = schedd.submit(submit)
```

# Removed xquery().

- The implementation of xquery() blocked the schedd until the Python bindings had read all the ads, and this caused a lot of problems.
- We have some ideas about how to decrease the memory usage and increase the speed of the new query method and will consider developing them if this removal causes people more grief than inconvenience.

# Exceptions

- We've retained `HTCondorException` and `ClassAdException`, but none of the subclasses.
- The intent is to allow you to catch module-specific exceptions without accidentally catching any exceptions those modules don't intentionally raise.
- Let us know how this works out for you.
- The documentation of which exceptions are raised from which methods is not complete.

# Questions? Comments?

htcondor-admin@cs.wisc.edu