




# Q&A/Discussion: HTCondor Python Bindings

Jason Patton  
Center for High Throughput Computing

# Tutorials

- › Prior to this session, you should have worked through the tutorials: [Manual](#) → [APIs](#) → [Python Bindings](#) → [Tutorials](#)
- › The “” link provides tutorials inside a JupyterLab environment
- › The tutorials cover:
  - Submitting jobs and DAGs
  - Manipulating ClassAds
  - Querying daemons

# Notable Changes

- › No longer suggest using Schedd transaction method for submitting jobs:

```
schedd = htcondor.Schedd()           # Create a schedd object using default settings.  
with schedd.transaction() as txn:    # txn will now represent the transaction.  
    cluster_id = sub.queue(txn)      # Queues one job in the current transaction
```

- › Instead, use the submit method:

```
schedd = htcondor.Schedd()           # get the Python representation of the scheduler  
submit_result = schedd.submit(sub)   # submit the job  
cluster_id = submit_result.cluster() # get the job's ClusterId
```

# Examples

- › condor\_watch\_q – Uses `htcondor.JobEventLog()` to monitor jobs' statuses without polling the schedd.
- › condor\_adstash, condor\_credmon\_oauth – Use `htcondor.set_subsystem()` and `htcondor.set_ready_state()` to signal status to a condor\_master.
- › box/gdrive/onedrive/gwdata file transfer plugins – Use the classad binding to read/write ClassAds from/to files.
- › HTMap – wraps job submission in a `map()`-like function.