

Scripting and Other Clever Things

Brian Bockelman

European HTCondor Site Admins Workshop

Topics for Today

- This presentation is not an overview of the command line utilities for sysadmins.
- Instead, it covers how to utilize them in your day to day work, with an emphasis on the “new and unknown” features that could make your life easier.
- I’ll cover both the command line tools and python scripts.

Condor CLI

Before we get into details, a reminder: condor command line utilities have immaculate man pages. Use them!

```
CRABClient - bbockelm@hcc-briantest:~ - ssh - 80x24
just-man-pages/condor_status(1)                just-man-pages/condor_status(1)

Name
  condor_status Display status of the HTCondor pool

Synopsis
  condor_status [ -debug ] [ help options ] [ query options ] [ display options ] [ custom options ] [ name ... ]

Description
  condor_status is a versatile tool that may be used to monitor and query the HTCondor pool. The condor_status tool can be used to query resource information, submitter information, checkpoint server information, and daemon master information. The specific query sent and the resulting information display is controlled by the query options supplied. Queries and display formats can also be customized.

  The options that may be supplied to condor_status belong to five groups:

    * Help options provide information about the condor_status tool.

    * Query options control the content and presentation of status
```

Basics - Projections and Filtering

- **Filtering:** Given a constraint expression, return only the ClassAds that match.
 - *Example:* Send me the list of job ads where **Owner=?="bbockelm"**.
 - The filter is expressed with the “-const” option.
- **Projection:** A list of attributes to return by the query.
 - *Example:* Send the job IDs, the Owner, and the number of requested CPUs for all jobs.
 - The projection is expressed with the “-af” option.
- Whenever we query remote daemon in HTCCondor, we do some combination of filtering and projection.

Basics - Querying Remote Daemons

- The `condor_collector` serves as a central information service - each daemon registers a `ClassAd` so you can contact it remotely.
- The “-pool” command line specifies what collector to query.
- For `condor_status`, you can specify a daemon type to show (such as “-schedd”).
- For `condor_q`, you can specify a remote schedd to query with “-name”.
- This is useful not only for querying, but for running admin scripts from a central host without having to SSH into each worker.

Querying a Remote Pool

bbockelm — bbockelm@hcc-briantest:~ — ssh — 117x28

```
[bbockelm@hcc-briantest ~]$ condor_status -pool vocms097.cern.ch -schedd -wide
```

Name	Machine	TotalRunningJobs	TotalIdleJobs	TotalHeldJobs
cmsgwms-submit1.fnal.gov	cmsgwms-submit1.fnal.gov	1402	620	0
cmsgwms-submit2.fnal.gov	cmsgwms-submit2.fnal.gov	1051	1110	0
cmssrv113.fnal.gov	cmssrv113.fnal.gov	124	0	0
cmssrv217.fnal.gov	cmssrv217.fnal.gov	0	0	0
cmssrv218.fnal.gov	cmssrv218.fnal.gov	0	0	0
cmssrv219.fnal.gov	cmssrv219.fnal.gov	0	0	0
cmssrv95.fnal.gov	cmssrv95.fnal.gov	5	0	0
hcc-crabserver.unl.edu	hcc-crabserver.unl.edu	29	4	13
crab3test-1@submit-5.t2.ucsd.edu	submit-5.t2.ucsd.edu	73	1503	139
crab3test-4@vocms0109.cern.ch	vocms0109.cern.ch	50	1858	0
crab3test-5@vocms0114.cern.ch	vocms0114.cern.ch	21	1	27
crab3test-7@vocms0115.cern.ch	vocms0115.cern.ch	0	0	0
vocms0224.cern.ch	vocms0224.cern.ch	0	0	0
vocms0230.cern.ch	vocms0230.cern.ch	0	0	0
vocms0231.cern.ch	vocms0231.cern.ch	0	0	0
vocms039.cern.ch	vocms039.cern.ch	0	0	0
crab3test-6@vocms059.cern.ch	vocms059.cern.ch	1387	3462	0
crab3test-2@vocms095.cern.ch	vocms095.cern.ch	47	2835	0
crab3test-3@vocms096.cern.ch	vocms096.cern.ch	3557	2672	1
vocms83.cern.ch	vocms83.cern.ch	142	11	3146
	TotalRunningJobs		TotalIdleJobs	TotalHeldJobs
	Total	7888	14076	3326

```
[bbockelm@hcc-briantest ~]$
```

Querying remote schedd

bbockelm — bbockelm@hcc-briantest:~ — ssh — 117x27

```
[bbockelm@hcc-briantest ~]$ condor_q -pool vocms097.cern.ch -name 'crab3test-4@vocms0109.cern.ch' | head -n 25
```

```
-- Schedd: crab3test-4@vocms0109.cern.ch : <128.142.143.118:4080?noUDP&sock=30738_b297_25>
```

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
153713.0	cms1120	10/7 09:09	57+19:54:02	R	10	0.1	dag_bootstrap_star
153714.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153715.0	cms1120	10/7 09:09	0+00:00:00	I	20	0.0	gWMS-CMSRunAnalsi
153716.0	cms1120	10/7 09:09	0+00:00:00	I	20	0.0	gWMS-CMSRunAnalsi
153717.0	cms1120	10/7 09:09	0+00:00:00	I	20	0.0	gWMS-CMSRunAnalsi
153718.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153719.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153720.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153721.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153722.0	cms1120	10/7 09:09	0+00:00:00	I	20	0.0	gWMS-CMSRunAnalsi
153723.0	cms1120	10/7 09:09	0+00:00:00	I	20	0.0	gWMS-CMSRunAnalsi
153724.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153725.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153726.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153727.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153728.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153729.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153730.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153731.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153732.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi
153733.0	cms1120	10/7 09:09	0+00:00:00	I	10	0.0	gWMS-CMSRunAnalsi

```
[bbockelm@hcc-briantest ~]$ █
```

Basics -

Commands that we've skipped so far

- **condor_qedit**: Edit job(s) attributes in the queue.
- **condor_fetchlog**: Fetch log from a remote daemon.
- **condor_drain**: Set a machine in drain mode (no longer accepts new jobs).
- **condor_ping**: Test authorization against a remote daemon.
- **condor_sos**: Prefix before another condor command to run on the “high priority” socket. Useful if the machine is busy/unresponsive.
 - `condor_sos condor_rm gthain`
- **condor_tail**: Tail stdout/err of remote job (use when `condor_ssh_to_job` is banned).
- **condor_who**: Print “who”-like output

Example - condor_who

```
condor-build — root@red-d9n1:~ — ssh — 96x21

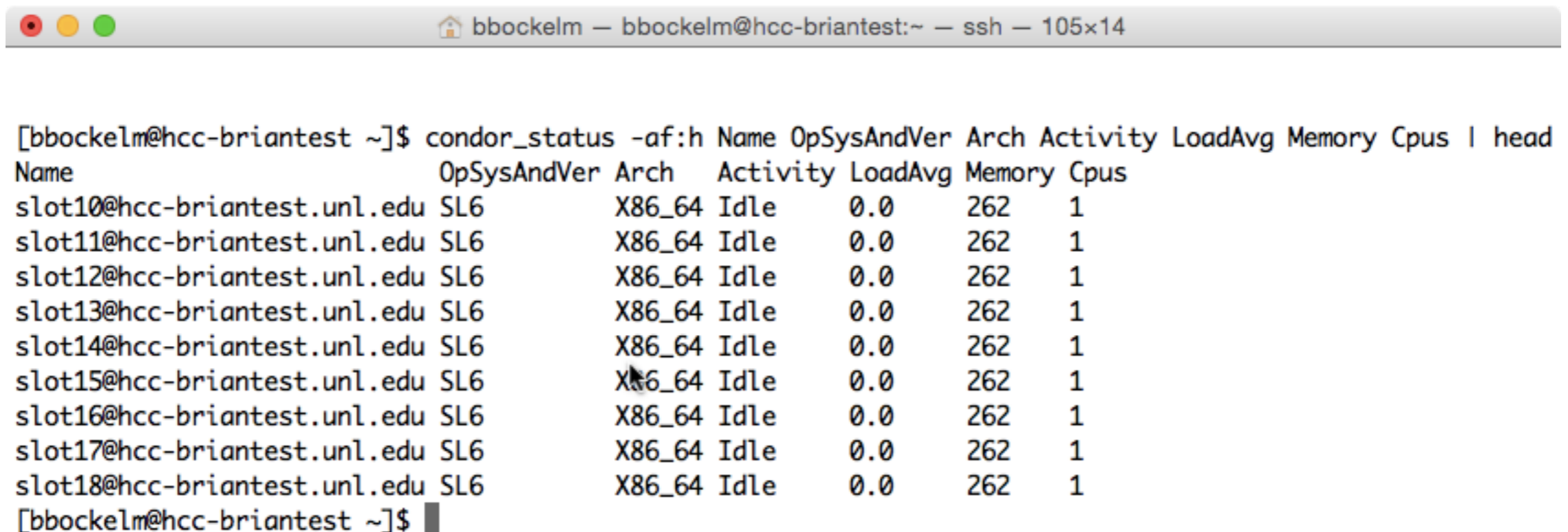
[root@red-d9n1 ~]# condor_who

OWNER                CLIENT                SLOT JOB                RUNTIME    PID    PROGRAM
cmsprod088@unl.edu   red.unl.edu          1_10 4180969.0            0+01:36:12 9572   /var/lib/condor/execut
uscmsPool2576@unl.edu red-gw2.unl.edu      1_7  3856773.0            0+00:06:05 18906  /var/lib/condor/execut
cmsprod088@unl.edu   red.unl.edu          1_13 4181377.0            0+00:06:00 21234  /var/lib/condor/execut
cmsprod088@unl.edu   red.unl.edu          1_8  4179962.0            0+05:38:33 21579  /var/lib/condor/execut
uscmsPool2576@unl.edu red.unl.edu          1_2  4178723.0            0+18:30:27
glow@unl.edu         red.unl.edu          1_15 4177548.0            1+06:18:47
glow@unl.edu         red.unl.edu          1_6  4177247.0            1+07:31:51
osg@unl.edu          red-gw2.unl.edu      1_9  3852922.0            1+12:07:58
osg@unl.edu          red.unl.edu          1_4  4176607.0            1+13:59:04
osg@unl.edu          red-gw2.unl.edu      1_5  3852566.0            1+14:22:04
osg@unl.edu          red-gw2.unl.edu      1_14 3852201.0            1+14:38:05
osg@unl.edu          red-gw1.unl.edu      1_3  3969215.0            1+15:55:46
osg@unl.edu          red.unl.edu          1_11 4176174.0            1+17:11:49
osg@unl.edu          red-gw1.unl.edu      1_1  3968911.0            1+17:14:47
osg@unl.edu          red-gw1.unl.edu      1_12 3968816.0            1+17:59:50

[root@red-d9n1 ~]# █
```

Tip #1: Use “-af” to create custom output tables

- The “-af” option allows you to form



```
[bbockelm@hcc-briantest ~]$ condor_status -af:h Name OpSysAndVer Arch Activity LoadAvg Memory Cpus | head
Name OpSysAndVer Arch Activity LoadAvg Memory Cpus
slot10@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot11@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot12@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot13@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot14@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot15@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot16@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot17@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
slot18@hcc-briantest.unl.edu SL6 X86_64 Idle 0.0 262 1
[bbockelm@hcc-briantest ~]$
```

Tip #2: Don't use Tip #1 for common queries

- HTCondor allows you to specify very detailed formatting options for condor_q and condor_status using the **print format** file.
 - See <https://htcondor-wiki.cs.wisc.edu/index.cgi/wiki?p=ExperimentalCustomPrintFormats> for documentation.
- These allow extremely high level of customization, especially with respect to formatting and headers. On the downside, they take effort to design and implement - this is not for ad-hoc queries.

Print format - custom status

```
CRABClient - bbockelm@hcc-briantest:~ - ssh - 101x26
[bbockelm@hcc-briantest ~]$ cat ~/.condor/status.cpf

# status.cpf
# produce the standard output of condor_status
SELECT
  Name          AS Name      WIDTH -18
  OpSysAndVer  AS OpSys      WIDTH -10
  State         AS State     WIDTH -9
  Activity     AS Activity   WIDTH -8
  LoadAvg      AS LoadAv          PRINTAS LOAD_AVG
  Memory       AS Mem          PRINTF "%4d"
  Cpus         AS CPU
  EnteredCurrentActivity AS " ActvtyTime\n" NOPREFIX PRINTAS ACTIVITY_TIME
SUMMARY NONE

[bbockelm@hcc-briantest ~]$ condor_status -pr ~/.condor/status.cpf | head -n 9
Name                OpSys      State      Activity LoadAv Mem CPU  ActvtyTime

slot10@hcc-briantest.unl.edu SL6        Unclaimed Idle      0.000  262 1   3+12:17:45
slot11@hcc-briantest.unl.edu SL6        Unclaimed Idle      0.000  262 1   3+12:17:46
slot12@hcc-briantest.unl.edu SL6        Unclaimed Idle      0.000  262 1   3+12:17:47
slot13@hcc-briantest.unl.edu SL6        Unclaimed Idle      0.000  262 1   3+12:17:48
slot14@hcc-briantest.unl.edu SL6        Unclaimed Idle      0.000  262 1   3+12:17:49
slot15@hcc-briantest.unl.edu SL6        Unclaimed Idle      0.000  262 1   3+12:17:50
slot16@hcc-briantest.unl.edu SL6        Unclaimed Idle      0.000  262 1   3+12:17:43
[bbockelm@hcc-briantest ~]$
```

Print format - Partitionable Slots

```
CRABClient - bbockelm@hcc-briantest:~ - ssh - 109x26
[bbockelm@hcc-briantest ~]$ cat ~/.condor/status_pslot.cpf
# status summary output for partitionable slots (8.1.6 or later for some fields)
SELECT
  Machine           AS Machine   WIDTH -32 TRUNCATE
  strcat(OpSysAndVer,"_x",substr(Arch,-2)) AS Platform WIDTH -8
  split(condorversion)[1] AS Condor
  TotalCpus         AS Cpus      PRINTF %4d
  Cpus              AS Free      PRINTF %4d
  TotalMemory/1024.0 AS " Mem(Gb)" PRINTF %8.2f
  ifThenElse(max(childmemory) is null, 0, max(childmemory)) AS MaxMem   WIDTH 6 TRUNCATE
  Memory*100.0/TotalMemory AS FreeMem%  PRINTF %8.1f
  NumDynamicSlots    AS Slots     WIDTH 5 TRUNCATE
  TotalLoadAvg*1.0/TotalCpus AS CpuUtil  PRINTF %7.2f
  RecentJobStarts/20.0 AS Jobs/Min PRINTF %8.2f
WHERE PartitionableSlot
SUMMARY NONE
[bbockelm@hcc-briantest ~]$ condor_status -pr ~/.condor/status_pslot.cpf -pool red-condor.unl.edu | head -n 8
Machine           Platform Condor Cpus Free Mem(Gb) MaxMem FreeMem% Slots CpuUtil Jobs/Min
red-d11n1.unl.edu  SL6_x64  8.3.1   16   1   37.54   2500    2.4    15   0.92   0.00
red-d11n10.unl.edu SL6_x64  8.3.1   16   1   37.54   2500    2.4    15   0.83   0.00
red-d11n11.unl.edu SL6_x64  8.3.1   16   1   37.54   2500    2.4    15   1.00   0.00
red-d11n12.unl.edu SL6_x64  8.3.1   16   1   37.54   2500    2.4    15   0.97   0.05
red-d11n13.unl.edu SL6_x64  8.3.1   16   1   37.54   2500    2.4    15   0.95   0.10
red-d11n14.unl.edu SL6_x64  8.3.1   16   1   37.54   2500    2.4    15   0.94   0.05
red-d11n15.unl.edu SL6_x64  8.3.1   16  16   37.54    0    100.0    0   0.00   0.00
[bbockelm@hcc-briantest ~]$
```

Print format - Defaults

- Hate the default condor_status? Think you can do better? You can make the print format file default for condor_status or condor_q!
 - If you do this, let the htcondor-users mailing list know - I'm curious how this feature is used "in the wild"

```
CRABClient - bbockelm@hcc-briantest:~ - ssh - 109x26
[bbockelm@hcc-briantest ~]$ condor_config_val STATUS_DEFAULT_STARTD_PRINT_FORMAT_FILE
/home/cse496/bbockelm/.condor/status.cpf
[bbockelm@hcc-briantest ~]$ condor_status | head -n 20
```

Name	OpSys	State	Activity	LoadAv	Mem	CPU	ActvtyTime
slot10@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:45
slot11@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:46
slot12@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:47
slot13@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:48
slot14@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:49
slot15@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:50
slot16@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:43
slot17@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:44
slot18@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:45
slot19@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:46
slot1@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:16
slot20@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:47
slot21@hcc-briantest.unl.edu	SL6	Unclaimed	Idle	0.000	262	1	3+12:37:48

Tip #3 - Never use “-l” in scripts

- The “-l” option to `condor_q` and `condor_status` prints all attributes for the returned ads. Never use this in scripts! Why?
 - This causes a heavy load on the remote host; there may be dozens of attributes per slot/job, even though you only want a few. If you’re scripting, use “-format” to return only the attributes you need, correctly formatted.
- “-l” is a handy tool for debugging when you don’t know where to start!

Tip #5 - Safely writing shell scripts

- Recall that the user has complete control over all attribute values in jobs *except* Owner. Thus, for the output of `condor_q`, you are accepting unsanitized input from remote users. In particular, string attributes are tricky.
- The only safe sanitization is through “-l” or “-xml”, assuming you have a compliant ClassAd or XML parser, respectively.
 - I know of no ClassAd / XML parser for shell scripts.
 - It’s tempting to write your own ClassAd parser using `sed + awk`; don’t, you can’t get the quoting rules right.
- Therefore, if you write shell scripts involving `condor_q`, make sure all outputs are formatted as integers, not strings. Consider dropping privileges to non-root if you take action based on jobs in the schedd.
- `condor_status` has less issues, but it’s still worth being careful.
- **ALTERNATELY**: Write admin scripts using the python bindings!

Tip #6: Ask the experts

- Scripts are great for ad-hoc or infrequently-done activities:
 - “Increase the requested disk size by 100MB for a single user.”
- Or for activities that correlate across several hosts:
 - “One-time increase in priority for all users running CMSSW jobs.”
- **However**, if it’s a frequent task, HTCondor may have a built-in way to do this; ask the experts!
 - Alternately, if you have a clever script, share it with the experts; we may want to someday include the functionality by default.

Part 2: Python Bindings

Python Bindings

- Introduced in the 7.9 series, the python bindings give access to the C++ libraries in a scripting language.
 - The bindings try to keep a balance between “python style” of APIs and “HTCondor style”.
 - Most of the CLI functionality is found within the bindings; you don’t have to fork processes, managing their lifetime, check exit codes, etc.
 - If things fail, exceptions are thrown!
 - This gives safe ways to quote and unquote ClassAd strings.
- Let’s switch to the python bindings tutorial from 2014: <http://research.cs.wisc.edu/htcondor/HTCondorWeek2014/presentations/TheisenT-Python.pdf>