

Managing and monitoring PBS jobs

Here are some of the most useful commands for managing and monitoring jobs launched with PBS and running on Casper or Cheyenne nodes.

Most of these commands will only modify or query data from jobs that are active on the same system. That is, run each command on Cheyenne if you want to interact with a job on Cheyenne.

Run any of these commands followed by **-h** to get help, as in **qhist -h**.

Page contents

- [qdel](#)
- [qhist](#)
- [qstat](#)

qdel

Run **qdel** with the job ID to kill a pending or running job.

```
qdel jobID
```

Kill all of your own pending or running jobs. (Be sure to use *backticks* as shown.)

```
qdel `qselect -u $USER`
```

qhist

Run **qhist** for information on finished jobs.

```
qhist -u $USER
```

Your output will include jobs that finished on the current day unless you specify the number (N) of days to include.

```
qhist -u $USER -d N
```

Your output will be similar to this, with Mem(GB) and CPU(%) indicating approximate total memory usage **per job** and average CPU usage **per core per job**:

Job ID	User	Queue	Nodes	NCPUs	Finish	RMem(GB)	Mem(GB)	CPU(%)	Elap(h)
2426690	stormyk	regular	1	1	05-1527	-	0.3	75.0	0.09
2426693	stormyk	regular	1	1	05-1527	-	0.1	90.0	0.09
2426541	stormyk	regular	1	1	05-1523	-	0.1	83.0	0.03
2426542	stormyk	regular	1	1	05-1524	-	0.1	70.0	0.04
2426683	stormyk	regular	1	1	05-1523	-	0.1	0.0	0.02
2426444	stormyk	regular	1	1	05-1522	-	0.1	19.0	0.02
2426435	stormyk	regular	1	1	05-1522	-	0.1	13.0	0.02

The following variation will generate a list of jobs that finished with non-zero exit codes to help you identify jobs that failed.

```
qhist -u $USER -r x0
```

qstat

Run this to see the status of all of your own unfinished jobs.

```
qstat -u $USER
```

Your output will be similar to what is shown just below. Most column headings are self-explanatory – NDS for nodes, TSK for tasks, and so on.

In the status (S) column, most jobs are either queued (Q) or running (R). Sometimes jobs are held (H), which might mean they are dependent on the completion of another job. If you have a job that is held and is not dependent on another job, CISL recommends killing and resubmitting the job.

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Memory	Req'd Time	Req'd S	Elap Time
657237.chadmin	apatelsm	economy	ens603	46100	60	216	--	02:30	R	01:24
657238.chadmin	apatelsm	regular	ens605	--	1	36	--	00:05	H	--
657466.chadmin	apatelsm	economy	ens701	5189	60	216	--	02:30	R	00:46
657467.chadmin	apatelsm	regular	ens703	--	1	36	--	00:10	H	--

Following are examples of qstat with some other commonly used options and arguments.

Get a long-form summary of the status of an unfinished job. (Use this only sparingly; it places a high load on PBS.)

```
qstat -f jobID
```

Get a single-line summary of the status of an unfinished or recently completed job (within 72 hours).

```
qstat -x jobID
```

Get information about unfinished jobs in a specified queue.

```
qstat queue_name
```

See job activity by queue (e.g., pending, running) in terms of numbers of jobs.

```
qstat -Q
```

Display information for all of your pending, running, and finished jobs.

```
qstat -x -u $USER
```

Query jobs running on one system by specifying **@cheyenne** or **@casper** from either system as shown here. (Only these options are supported when running **qstat** in this cross-server mode: **-x**, **-u**, **-w**, **-n**, **-s**)

```
qstat -w -u $USER @cheyenne
```