

# Torque

## I see exit codes from Torque and showq, but what do they mean.

### Explanation:

An exit code of zero always means the job completed successfully, as far as Torque can tell.

There are only two types of exit codes returned by Torque. Torque, itself, has it's own set of exit codes, which are returned when a job did not complete and exit on it's own, which are in the range of -1 through -14. As of version 6.1.2 (check the documentation for your version), these are as follows:

- 0: Job execution successful
- 1: Job execution failed, before files, no retry
- 2: Job execution failed, after files, no retry
- 3: Job execution failed, do retry
- 4: Job aborted on MOM initialization
- 5: Job aborted on MOM init, chkpt, no migrate
- 6: Job aborted on MOM init, chkpt, ok migrate
- 7: Job restart failed
- 8: Exec() of user command failed
- 9: Could not create/open stdout stderr files
- 10 Job exceeded a memory limit
- 11 Job exceeded a walltime limit
- 12 Job exceeded a CPU time limit
- 13 Could not create the job's cgroups
- 14 Prologue failed\_EXEC

When a job script was successfully started and completes without Torque intervention, the completion code returned is exactly as it is was received by Torque's Mom. This will also be the exit code from the last command executed in the job shell, with the exception of un-caught signals (see below). Most shells will mask the job's exit code to 7 bits, although a few (ksh, for one) mas to 8 bits. This means with an 7-bit mask, if a job does an "exit(258)", or "exit 130", the exit code will show up as "2". Likewise, if a job should execute "exit(-1)", the exit code will be

# Torque

127. For this reason, jobs should be aware of their use of exit codes, and it's best practice to keep those in the range of 0-127.

There is one other possible set of exit code values. When a shell detects a command exited due to a signal, the return value is the signal value added to 128 (or 256, for shells with 8-bit masks). To see a list of all signals for a given Linux distribution, run "kill -l" (the letter l), or for "ksh", "kill -L". So an exit value of 137, for example, would indicate a job was killed with the "-9" signal. The ksh shell is different, as it sets the low-order bit of the upper byte, effectively adding "256" to the signal value. So a return code of 265 from a job running ksh would indicate the job was killed with SIGKILL (9).

The above discussion is referring to jobs where the job, itself, ran and exited. If Moab was unable to start the job, or if the job was canceled before it started, then a "showq" would show something entirely different. For a canceled job, as an example, "showq" will show a CCODE value of "R CNCLD".

Unique solution ID: #1212  
Author: Rob Greenbank  
Last update: 2018-08-22 19:41