

Deleting a request from a CXbatch queue

To delete a request from a CXbatch queue, use the `qdel` command, which has the following syntax:

```
qdel [option] request_id[@host]
```

where *request_id* is the request ID of the job you want to signal, and *host* is the machine from which the request was submitted. *option* can be either of the following:

- `-k` Send a kill signal (SIGKILL) to kill a job.
- `-sig` Send a specified signal to a job, where *sig* can be the number or name of any one of the following signals:
 - 1 QUIT Quit execution.
 - 2 INT Interrupt execution.
 - 6 ABRT Abort execution.
 - 9 KILL Kill execution. (Same as `-k` option.)
 - 15 TERM Terminate execution.
 - 30 USR1 User-defined signal number 1.
 - 31 USR2 User-defined signal number 2.

For more information on signals, refer to the `signal(3c)` man page. The signal you send to your program when deleting it from a CXbatch queue depends on the program's signal handling. Generally, the `-k` option is sufficient.

CONVEX CXbatch for Users Quick Reference

Order No. DSW-188
Second Edition
January 1992

Copyright ©1992 CONVEX Computer Corporation
All rights reserved.

CONVEX and the CONVEX logo "C" are registered trademarks of CONVEX Computer Corporation.

QUICK REFERENCE

CONVEX CXbatch for Users

Order No. DSW-188

Second Edition
January 1992

Printed in the United States of America

Document No. 710-009430-001



CONVEX COMPUTER CORPORATION

Submitting requests to a CXbatch queue

To submit job requests to a CXbatch queue, use the `qsub` command, which has the following syntax:

```
qsub [option...]
```

where *option* can be one or more options described in the sections that follow.

-a *time*

Run request after a stated time. *time* can include day, month, and year, as well as time zone. If any spaces are imbedded, *time* must be enclosed by quotes. For example:

```
% qsub -a "Jan 1, 2001 12:00 -EDT"
```

-h Put request on hold at time of submission.

-i Import your current directory.

-mb Send mail when your request begins executing.

-me Send mail when your request ends execution.

-mu *username*

Send mail for request to specific user indicated by *username*.

-ni Do not import current directory.

-p *priority*

Specify intra-queue priority, where *priority* is an integer between 0 (zero) and 63. Zero is the lowest priority, and 63 is the highest.

-q *queuename* [*@hostname*]

Specify queue to which a request is submitted, where *queuename* is the name of the requested queue, and *hostname* is the name of the host on which the queue resides. *hostname* is required if the host is not localhost.

-r *request_name*

Assign a name to request, where *request_name* is truncated after 15 characters.

-s *shell_name*

Specify a shell to interpret commands, where *shell_name* is the path to the shell.

-t *process_id*

Signal a process when request is finished, where *process_id* is an integer representing the PID.

-x Export values of all environment variables.

-y Append accounting information to stdout file.

Directing output and error messages

-e [*hostname:*] *filename*

Redirect standard output error to *filename*, where *filename* is a complete path name. *hostname* is the name of a remote system, if *filename* is to reside there.

-eo Send error output to standard output file.

-ke Leave standard error output file on the executing machine.

-ko Leave standard output file on executing machine.

-o [*hostname:*] *filename*

Redirect standard output. to *filename*, where *filename* is the complete path name. *hostname* is the name of a remote system, if *filename* is to reside there.

Setting per-process limits

When using the `qsub` command, you can set per-process limits by using the options in this section. With the following options, *size_limit* is the hard limit for the particular per-process limit, and *warning_limit* is the point at which you are warned that you are approaching your hard limit.

-ld *size_limit* [, *warning_limit*]

Establish per-process data segment limit.

-lf *size_limit* [, *warning_limit*]

Establish per-process permanent file limit.

-lm *size_limit* [, *warning_limit*]

Establish per-process memory limit.

-ln *size_limit* [, *warning_limit*]

Establish per-process nice value limit.

-ls *size_limit* [, *warning_limit*]

Establish per-process stack segment limit.

-lt *size_limit* [, *warning_limit*]

Establish per-process CPU time limit.

-lv *size_limit* [, *warning_limit*]

Establish per-process temporary file limit.

-lw *size_limit* [, *warning_limit*]

Establish per-process working set limit.

Support of per-process limits must be set by your system manager. To view per-process limits that are supported on your system, enter at the shell prompt:

```
% qmgr show limits_supported
```

Checkpointing submissions

-c Indicate request can be checkpointed.

-cp *number* {minutes|hours|days|weeks}

Checkpoint request at intervals of the specified period, where *number* is a positive integer representing the number of units of the period. One unit of measure (minutes, hours, days, or weeks) must be specified. Do not include a space between the number and the unit.

-nc Indicate request cannot be checkpointed.

Displaying CXbatch queue information

The `qstat` command displays information about the status of CXbatch queues and has the following syntax:

```
qstat [option] [queuename[@hostname]...]
```

where *queuename* is the name of the queue that is being queried, *hostname* is the name of the remote host if the queue is not local, and *option* is any one of the options that follow:

-a Show all requests.

-l Show requests in long format.

-m Show requests in medium-length format.

-u *username*

Show only those requests belonging to the user specified by *username*.

-x Show queue header in extended format.