qsub

Submit PBS job.

Synopsis

```
qsub [-a date_time] [-A account_string] [-b secs] [-c checkpoint_options]
[-C directive_prefix] [-d path] [-D path] [-e path] [-f] [-F] [-h]
[-I ] [-j join ] [-k keep ] [-l resource_list ]
[-m mail_options] [-M user_list] [-n] [-N name] [-o path]
[-p priority] [-P user[:group]] [-q destination] [-r c] [-S path_to_shell(s)]
[-t array_request] [-u user_list]
[-v variable_list] [-V] [-W additional_attributes] [-x] [-X] [-z] [script]
```

Description

To create a job is to submit an executable script to a batch server. The batch server will be the default server unless the **-q** option is specified. The command parses a script prior to the actual script execution; it does not execute a script itself. All script-writing rules remain in effect, including the "**#!**" at the head of the file (see discussion of PBS_DEFAULT under Environment variables). Typically, the script is a shell script which will be executed by a command shell such as sh or csh.

Options on the qsub command allow the specification of attributes which affect the behavior of the job.

The qsub command will pass certain environment variables in the Variable_List attribute of the job. These variables will be available to the job. The value for the following variables will be taken from the environment of the qsub command: HOME, LANG, LOGNAME, PATH, MAIL, SHELL, and TZ. These values will be assigned to a new name which is the current name prefixed with the string "PBS_O_". For example, the job will have access to an environment variable named PBS_O_HOME which have the value of the variable HOME in the qsub command environment.

In addition to the above, the following environment variables will be available to the batch job:

Variable	Description
PBS_O_HOST	The name of the host upon which the qsub command is running.
PBS_SERVER	The hostname of the pbs_server which qsub submits the job to.
PBS_O_QUEUE	The name of the original queue to which the job was submitted.

Variable	Description
PBS_O_WORKDIR	The absolute path of the current working directory of the qsub command.
PBS_ARRAYID	Each member of a job array is assigned a unique identifier (see **-t** option).
PBS_ENVIRONMENT	Set to PBS_BATCH to indicate the job is a batch job, or to PBS_INTERACTIVE to indicate the job is a PBS interactive job (see **-I** option).
PBS_JOBID	The job identifier assigned to the job by the batch system. It can be used in the stdout and stderr paths. TORQUE replaces $PBS_JOBID with the job's jobid (for example, #PBS -o /tmp/$PBS_JOBID.output).
PBS_JOBNAME	The job name supplied by the user.
PBS_NODEFILE	The name of the file contain the list of nodes assigned to the job (for parallel and cluster systems).
PBS_QUEUE	The name of the queue from which the job is executed.

Options

Option	Name	Description
-a	date_time	Declares the time after which the job is eligible for execution.
		The date_time argument is in the form:
		`[[[[CC]YY]MM]DD]hhmm[.SS]`
		where *CC* is the first two digits of the year (the century), *YY* is the second two digits of the year, *MM* is the two digits for the month, *DD* is the day of the month, *hh* is the hour, *mm* is the minute, and the optional *SS* is the seconds.
		If the month (MM) is not specified, it will default to the current month if the specified day (DD) is in the future. Otherwise, the month will be set to next month. Likewise, if the day (DD) is not specified, it will default to today if the time (hhmm) is in the future. Otherwise, the day will be set to tomorrow.
		For example, if you submit a job at 11:15 am with a time of -a 1110, the job will be eligible to run at 11:10 am tomorrow.
-A	account_string	Defines the account string associated with the job. The account_string is an undefined string of characters and is interpreted by the server which executes the job. See section 2.7.1 of the PBS ERS.
-b	seconds	Defines the maximum number of seconds qsub will block attempting to contact pbs_server. If pbs_server is down, or for a variety of communication failures, qsub will continually retry

Option	Name	Description
		connecting to pbs_server for job submission. This value overrides the CLIENTRETRY parameter in `torque.cfg`. This is a non-portable TORQUE extension. Portability-minded users can use the PBS_CLIENTRETRY environmental variable. A negative value is interpreted as infinity. The default is 0.
-c	checkpoint_options	Defines the options that will apply to the job. If the job executes upon a host which does not support checkpoint, these options will be ignored. Valid checkpoint options are: • **none** – No checkpointing is to be performed. • **enabled** – Specify that checkpointing is allowed but must be explicitly invoked by either the **qhold** or **qchkpt** commands. • **shutdown** – Specify that checkpointing is to be done on a job at pbs_mom shutdown. • **periodic** – Specify that periodic checkpointing is enabled. The default interval is 10 minutes and can be changed by the $checkpoint_interval option in the MOM config file or by specifying an interval when the job is submitted • **interval=minutes** – Checkpointing is to be performed at an interval of minutes, which is the integer number of minutes of wall time used by the job. This value must be greater than zero. • **depth=number** – Specify a number (depth) of checkpoint images to be kept in the checkpoint directory. • **dir=path** – Specify a checkpoint directory (default is `/var/spool/torque/checkpoint`).
-C	directive_prefix	Defines the prefix that declares a directive to the qsub command within the script file. (See the paragraph on script directives under Extended description.) If the –c option is presented with a directive_prefix argument that is the null string, qsub will not scan the script file for directives.
-d	path	Defines the working directory path to be used for the job. If the –d option is not specified, the default working directory is the home directory. This option sets the environment variable PBS_O_INITDIR.
-D	path	Defines the root directory to be used for the job. This option sets the environment variable PBS_O_ROOTDIR.
-e	path	Defines the path to be used for the standard error stream of the batch job. The path argument is of the form: `[hostname:]path_name` where *hostname* is the name of a host to which the file will be returned, and *path_name* is the path name on that host in the syntax recognized by POSIX. The argument will be interpreted as

Option	Name	Description
		follows: • **path_name** – where *path_name* is not an absolute path name, then the qsub command will expand the path name relative to the current working directory of the command. The command will supply the name of the host upon which it is executing for the hostname component. • **hostname:path_name** – where *path_name* is not an absolute path name, then the qsub command will not expand the path name relative to the current working directory of the command. On delivery of the standard error, the path name will be expanded relative to the users home directory on the hostname system. • **path_name** – where *path_name* specifies an absolute path name, then the qsub will supply the name of the host on which it is executing for the hostname. • **hostname:path_name** – where *path_name* specifies an absolute path name, the path will be used as specified. If the –e option is not specified, the default file name for the standard error stream will be used. The default name has the following form: • **job_name.esequence_number** – where *job_name* is the name of the job (see the **-n** name option) and *sequence_number* is the job number assigned when the job is submitted.
-f	---	Job is made fault tolerant. Jobs running on multiple nodes are periodically polled by mother superior. If one of the nodes fails to report, the job is canceled by mother superior and a failure is reported. If a job is fault tolerant, it will not be canceled based on failed polling (no matter how many nodes fail to report). This may be desirable if transient network failures are causing large jobs not to complete, where ignoring one failed polling attempt can be corrected at the next polling attempt. ⓘ If TORQUE is compiled with PBS_NO_POSIX_VIOLATION (there is no config option for this), you have to use –W `fault_tolerant=true` to mark the job as fault tolerant.
-F	---	Specfies the arguments that will be passed to the job script when the script is launched. The accepted syntax is: `qsub –F "myarg1 myarg2 myarg3=myarg3value" myscript2.sh` ⓘ Quotation marks are required. qsub will fail with an error message if the argument following –F is not a quoted value. The pbs_mom server will pass the quoted value as arguments to the job script when it launches the script.
-h	---	Specifies that a user hold be applied to the job at submission time.

Option	Name	Description
-I	---	Declares that the job is to be run "interactively". The job will be queued and scheduled as any PBS batch job, but when executed, the standard input, output, and error streams of the job are connected through qsub to the terminal session in which qsub is running. Interactive jobs are forced to not rerunable. See Extended description for additional information of interactive jobs.
-j	join	Declares if the standard error stream of the job will be merged with the standard output stream of the job. An option argument value of **oe** directs that the two streams will be merged, intermixed, as standard output. An option argument value of **eo** directs that the two streams will be merged, intermixed, as standard error. If the join argument is **n** or the option is not specified, the two streams will be two separate files.

-k	keep	Defines which (if either) of standard output or standard error will be retained on the execution host. If set for a stream, this option overrides the path name for that stream. If not set, neither stream is retained on the execution host.
		
		 The argument is either the single letter "e" or "o", or the letters "e" and "o" combined in either order. Or the argument is the letter "n".
		
		 • **e** – The standard error stream is to retained on the execution host. The stream will be placed in the home directory of the user under whose user id the job executed. The file name will be the default file name given by:
		
		 `job_name.esequence`
		
		 where *job_name* is the name specified for the job, and *sequence* is the sequence number component of the job identifier.
		
		 • **o** – The standard output stream is to retained on the execution host. The stream will be placed in the home directory of the user under whose user id the job executed. The file name will be the default file name given by:
		
		 `job_name.osequence`
		
		 where *job_name* is the name specified for the job, and *sequence* is the sequence number component of the job identifier.
		
		 • **eo** – Both the standard output and standard error streams will be retained.
		
		 • **oe** – Both the standard output and standard error streams will be retained.
		
		 • **n** – Neither stream is retained.

-l	resource_list	Defines the resources that are required by the job and establishes a limit to the amount of resource that can be consumed. If not set for a generally available resource, such as CPU time, the limit is infinite. The resource_list argument is of the form: `resource_name[=[value]][,resource_name[=[value]],...]`

Option	Name	Description
		ⓘ In this situation, you should request the more inclusive resource first. For example, a request for procs should come before a gres request. For information on specifying multiple types of resources for allocation, see "Multi-Req Support" under "General Job Policies" in the Moab Workload Manager documentation.
-m	mail_options	Defines the set of conditions under which the execution server will send a mail message about the job. The mail_options argument is a string which consists of either the single character "n", or one or more of the characters "a", "b", and "e". If the character "n" is specified, no normal mail is sent. Mail for job cancels and other events outside of normal job processing are still sent. For the letters "a", "b", and "e": • **a** – Mail is sent when the job is aborted by the batch system. • **b** – Mail is sent when the job begins execution. • **e** – Mail is sent when the job terminates. If the –m option is not specified, mail will be sent if the job is aborted.
-M	user_list	Declares the list of users to whom mail is sent by the execution server when it sends mail about the job. The user_list argument is of the form: `user[@host][,user[@host],...]` If unset, the list defaults to the submitting user at the qsub host, i.e. the job owner.
-n	node-exclusive	Allows a user to specify an exclusive-node access/allocation request for the job. This affects only cpusets and compatible schedulers (see Linux cpuset support).
-N	name	Declares a name for the job. The name specified may be an unlimited number of characters in length. It must consist of printable, non white space characters with the first character alphabetic. If the –N option is not specified, the job name will be the base name of the job script file specified on the command line. If no script file name was specified and the script was read from the standard input, then the job name will be set to STDIN.
-o	path	Defines the path to be used for the standard output stream of the batch job. The path argument is of the form: `[hostname:]path_name` where *hostname* is the name of a host to which the file will be

Option	Name	Description
		returned, and *path_name* is the path name on that host in the syntax recognized by POSIX. The argument will be interpreted as follows: • **path_name** – where *path_name* is not an absolute path name, then the qsub command will expand the path name relative to the current working directory of the command. The command will supply the name of the host upon which it is executing for the hostname component. • **hostname:path_name** – where *path_name* is not an absolute path name, then the qsub command will not expand the path name relative to the current working directory of the command. On delivery of the standard output, the path name will be expanded relative to the users home directory on the hostname system. • **path_name** – where *path_name* specifies an absolute path name, then the qsub will supply the name of the host on which it is executing for the hostname. • **hostname:path_name**where *path_name* specifies an absolute path name, the path will be used as specified. If the –o option is not specified, the default file name for the standard output stream will be used. The default name has the following form: • **job_name.osequence_number** – where *job_name* is the name of the job (see the **-n** name option) and *sequence_number* is the job number assigned when the job is submitted.
-p	priority	Defines the priority of the job. The priority argument must be a integer between -1024 and +1023 inclusive. The default is no priority which is equivalent to a priority of zero.
-P	user[:group]	Allows a root user to submit a job as another user. TORQUE treats proxy jobs as though the jobs were submitted by the supplied username. This feature is available in TORQUE 2.4.7 and later, however, TORQUE 2.4.7 does not have the ability to supply the `[:group]` option; it is available in TORQUE 2.4.8 and later.
-q	destination	Defines the destination of the job. The destination names a queue, a server, or a queue at a server. The qsub command will submit the script to the server defined by the destination argument. If the destination is a routing queue, the job may be routed by the server to a new destination. If the -q option is not specified, the qsub command will submit the script to the default server. (See Environment variables and the PBS ERS section 2.7.4, "Default Server".) If the -q option is specified, it is in one of the following three forms: • queue • @server • queue@server

Option	Name	Description
		If the destination argument names a queue and does not name a server, the job will be submitted to the named queue at the default server.
		If the destination argument names a server and does not name a queue, the job will be submitted to the default queue at the named server.
		If the destination argument names both a queue and a server, the job will be submitted to the named queue at the named server.
-r	y/n	Declares whether the job is rerunable (see the **qrerun** command). The option argument is a single character, either y or n.
		If the argument is "y", the job is rerunable. If the argument is "n", the job is not rerunable. The default value is y, rerunable.
-S	path_list	Declares the path to the desires shell for this job.
		`qsub script.sh -S /bin/tcsh`
		The option argument path_list is in the form:
		`path[@host][,path[@host],...]`
		Only one path may be specified for any host named. Only one path may be specified without the corresponding host name. The path selected will be the one with the host name that matched the name of the execution host. If no matching host is found, then the path specified without a host will be selected, if present.
		If the -S option is not specified, the option argument is the null string, or no entry from the path_list is selected, the execution will use the users login shell on the execution host.
-t	array_request	Specifies the task ids of a job array. Single task arrays are allowed.
		The array_request argument is an integer id or a range of integers. Multiple ids or id ranges can be combined in a comma delimted list. Examples: `-t 1-100` or `-t 1,10,50-100`
		An optional **slot limit** can be specified to limit the amount of jobs that can run concurrently in the job array. The default value is unlimited. The slot limit must be the last thing specified in the array_request and is delimited from the array by a percent sign (**%**).
| | | ```
qsub script.sh -t 0-299%5
``` |
		This sets the slot limit to 5. Only 5 jobs from this array can run at the same time.
		You can use **qalter** to modify slot limits on an array. The server parameter **max_slot_limit** can be used to set a global slot limit policy. When using slot limits in TORQUE with Moab or Maui, you should also set the moab_array_compatible server parameter to `TRUE`.
-u	user_list	Defines the user name under which the job is to run on the execution system.

Option	Name	Description
		The user_list argument is of the form: `user[@host][,user[@host],...]` Only one user name may be given per specified host. Only one of the user specifications may be supplied without the corresponding host specification. That user name will used for execution on any host not named in the argument list. If unset, the user list defaults to the user who is running qsub.
-v	variable_list	Expands the list of environment variables that are exported to the job. In addition to the variables described in the "Description" section above, variable_list names environment variables from the qsub command environment which are made available to the job when it executes. The variable_list is a comma separated list of strings of the form `variable` or `variable=value`. These variables and their values are passed to the job.
-V	---	Declares that all environment variables in the qsub commands environment are to be exported to the batch job.

-W	additional_attributes	The –W option allows for the specification of additional job attributes. The general syntax of –W is in the form:
		`-W attr_name=attr_value[,attr_name=attr_value...]`
		
		ⓘ If white space occurs anywhere within the option argument string or the equal sign, "**=**", occurs within an attribute_value string, then the string must be enclosed with either single or double quote marks.
		
		PBS currently supports the following attributes within the –W option:
		• **depend=dependency_list** – Defines the dependency between this and other jobs. The dependency_list is in the form:
		`type[:argument[:argument...]][,type:argument...]`
		The argument is either a numeric count or a PBS job id according to type. If argument is a count, it must be greater than 0. If it is a job id and not fully specified in the form `seq_number.server.name`, it will be expanded according to the default server rules which apply to job IDs on most commands. If argument is null (the preceding colon need not be specified), the dependency of the corresponding type is cleared (unset). For more information, see depend=dependency_list valid dependencies.
		• **group_list=g_list** – Defines the group name under which the job is to run on the execution system. The g_list argument is of the form:
		`group[@host][,group[@host],...]`
		Only one group name may be given per specified host. Only

Option	Name	Description
		one of the group specifications may be supplied without the corresponding host specification. That group name will used for execution on any host not named in the argument list. If not set, the group_list defaults to the primary group of the user under which the job will be run.
		• **interactive=true** – If the interactive attribute is specified, the job is an interactive job. The **-I** option is a alternative method of specifying this attribute.
		• **stagein=file_list**
		• **stageout=file_list** – Specifies which files are staged (copied) in before job start or staged out after the job completes execution. On completion of the job, all staged-in and staged-out files are removed from the execution system. The file_list is in the form:
		`local_file@hostname:remote_file[,...]`
		regardless of the direction of the copy. The name local_file is the name of the file on the system where the job executed. It may be an absolute path or relative to the home directory of the user. The name remote_file is the destination name on the host specified by hostname. The name may be absolute or relative to the users home directory on the destination host. The use of wildcards in the file name is not recommended. The file names map to a remote copy program (rcp) call on the execution system in the follow manner:
		○ For stagein: `rcp hostname:remote_file local_file`
		○ For stageout: `rcp local_file hostname:remote_file`
		Data staging examples:
		`-W stagein=/tmp/input.txt@headnode:/home/user/input.txt`
		`-W stageout=/tmp/output.txt@headnode:/home/user/output.txt`
		If TORQUE has been compiled with wordexp support, then variables can be used in the specified paths. Currently only `$PBS_JOBID`, `$HOME`, and `$TMPDIR` are supported for stagein.
		• **umask=XXX** – Sets umask used to create stdout and stderr spool files in pbs_mom spool directory. Values starting with 0 are treated as octal values, otherwise the value is treated as a decimal umask value.
-X	---	Enables X11 forwarding. The DISPLAY environment variable must be set.
-z	---	Directs that the qsub command is not to write the job identifier assigned to the job to the commands standard output.

depend=dependency_list valid dependencies

Dependency	Description
synccount:count	This job is the first in a set of jobs to be executed at the same time. Count is the number of additional jobs in the set.
syncwith:jobid	This job is an additional member of a set of jobs to be executed at the same time. In the above and following dependency types, jobid is the job identifier of the first job in the set.
after:jobid[:jobid...]	This job may be scheduled for execution at any point after jobs jobid have started execution.
afterok:jobid[:jobid...]	This job may be scheduled for execution only after jobs jobid have terminated with no errors. See the csh warning under Extended description.
afternotok:jobid[:jobid...]	This job may be scheduled for execution only after jobs jobid have terminated with errors. See the csh warning under Extended description.
afterany:jobid[:jobid...]	This job may be scheduled for execution after jobs jobid have terminated, with or without errors.
on:count	This job may be scheduled for execution after count dependencies on other jobs have been satisfied. This form is used in conjunction with one of the "before" forms (see below).
before:jobid[:jobid...]	When this job has begun execution, then jobs jobid... may begin.
beforeok:jobid[:jobid...]	If this job terminates execution without errors, then jobs jobid... may begin. See the csh warning under Extended description.
beforenotok:jobid[:jobid...]	If this job terminates execution with errors, then jobs jobid... may begin. See the csh warning under Extended description.
beforeany:jobid[:jobid...]	When this job terminates execution, jobs jobid... may begin. If any of the before forms are used, the jobs referenced by jobid must have been submitted with a dependency type of on. If any of the before forms are used, the jobs referenced by jobid must have the same owner as the job being submitted. Otherwise, the dependency is ignored.

ℹ Array dependencies make a job depend on an array or part of an array. If no count is given,

Dependency	Description
then the entire array is assumed. For examples, see Dependency examples.	
afterstartarray:arrayid[count]	After this many jobs have started from arrayid, this job may start.
afterokarray:arrayid[count]	This job may be scheduled for execution only after jobs in arrayid have terminated with no errors.
afternotokarray:arrayid[count]	This job may be scheduled for execution only after jobs in arrayid have terminated with errors.
afteranyarray:arrayid[count]	This job may be scheduled for execution after jobs in arrayid have terminated, with or without errors.
beforestartarray:arrayid[count]	Before this many jobs have started from arrayid, this job may start.
beforeokarray:arrayid[count]	If this job terminates execution without errors, then jobs in arrayid may begin.
beforenotokarray:arrayid[count]	If this job terminates execution with errors, then jobs in arrayid may begin.
beforeanyarray:arrayid[count]	When this job terminates execution, jobs in arrayid may begin. If any of the before forms are used, the jobs referenced by arrayid must have been submitted with a dependency type of on. If any of the before forms are used, the jobs referenced by arrayid must have the same owner as the job being submitted. Otherwise, the dependency is ignored.

ℹ Error processing of the existence, state, or condition of he job on which the newly submitted job is a deferred service, i.e. the check is performed after the job is queued. If an error is detected, the new job will be deleted by the server. Mail will be sent to the job submitter stating the error.

Dependency examples

```
qsub -W depend=afterok:123.big.iron.com /tmp/script
```

```
qsub -W depend=before:234.hunk1.com:235.hunk1.com
```

```
/tmp/script
```

```
qsub script.sh -W depend=afterokarray:427[]
```

This assumes every job in array 427 has to finish successfully for the dependency to be satisfied.

```
qsub script.sh -W depend=afterokarray:427[][5]
```

This means that 5 of the jobs in array 427 have to successfully finish in order for the dependency to be satisfied.

Operands

The qsub command accepts a script operand that is the path to the script of the job. If the path is relative, it will be expanded relative to the working directory of the qsub command.

If the script operand is not provided or the operand is the single character "**-**", the qsub command reads the script from standard input. When the script is being read from Standard Input, qsub will copy the file to a temporary file. This temporary file is passed to the library interface routine pbs_submit. The temporary file is removed by qsub after pbs_submit returns or upon the receipt of a signal which would cause qsub to terminate.

Standard input

The qsub command reads the script for the job from standard input if the script operand is missing or is the single character "**-**".

Input files

The script file is read by the qsub command. qsub acts upon any directives found in the script.

When the job is created, a copy of the script file is made and that copy cannot be modified.

Standard output

Unless the **-z** option is set, the job identifier assigned to the job will be written to standard output if the job is successfully created.

Standard error

The qsub command will write a diagnostic message to standard error for each error occurrence.

Environment variables

The values of some or all of the variables in the qsub commands environment are exported with the job (see the **-v** and **-v** options).

The environment variable PBS_DEFAULT defines the name of the default server. Typically, it

corresponds to the system name of the host on which the server is running. If PBS_DEFAULT is not set, the default is defined by an administrator established file.

The environment variable PBS_DPREFIX determines the prefix string which identifies directives in the script.

The environment variable PBS_CLIENTRETRY defines the maximum number of seconds qsub will block (see the **-b** option). Despite the name, currently qsub is the only client that supports this option.

torque.cfg

The `torque.cfg` file, located in PBS_SERVER_HOME (`/var/spool/torque` by default) controls the behavior of the qsub command. This file contains a list of parameters and values separated by whitespace.

- **QSUBSLEEP** – takes an integer operand which specifies time to sleep when running qsub command. Used to prevent users from overwhelming the scheduler.
- **SUBMITFILTER** – specifies the path to the submit filter used to pre-process job submission. The default path is libexecdir/qsub_filter, which falls back to /usr/local /sbin/torque_submitfilter for backwards compatibility. This torque.cfg parameter overrides this default.
- **SERVERHOST**
- **QSUBHOST**
- **QSUBSENDUID**
- **XAUTHPATH**
- **CLIENTRETRY**
- **VALIDATEGROUP**
- **DEFAULTCKPT**
- **VALIDATEPATH**
- **RERUNNABLEBYDEFAULT**

For example:

```
QSUBSLEEP 2
```

```
RERUNNABLEBYDEFAULT false
```

Extended description

Script Processing:

A job script may consist of PBS directives, comments and executable statements. A PBS directive provides a way of specifying job attributes in addition to the command line options. For example:

```
:
#PBS -N Job_name
#PBS -l walltime=10:30,mem=320kb
#PBS -m be
```

```
#
step1 arg1 arg2
step2 arg3 arg4
```

The qsub command scans the lines of the script file for directives. An initial line in the script that begins with the characters "**#!**" or the character "**:**" will be ignored and scanning will start with the next line. Scanning will continue until the first executable line, that is a line that is not blank, not a directive line, nor a line whose first non white space character is "**#**". If directives occur on subsequent lines, they will be ignored.

A line in the script file will be processed as a directive to qsub if and only if the string of characters starting with the first non white space character on the line and of the same length as the directive prefix matches the directive prefix.

The remainder of the directive line consists of the options to qsub in the same syntax as they appear on the command line. The option character is to be preceded with the "**-**" character.

If an option is present in both a directive and on the command line, that option and its argument, if any, will be ignored in the directive. The command line takes precedence.

If an option is present in a directive and not on the command line, that option and its argument, if any, will be processed as if it had occurred on the command line.

The directive prefix string will be determined in order of preference from:

- The value of the **-c** option argument if the option is specified on the command line.
- The value of the environment variable PBS_DPREFIX if it is defined.
- The four character string #PBS.

If the **-c** option is found in a directive in the script file, it will be ignored.

User Authorization:

When the user submits a job from a system other than the one on which the PBS Server is running, the name under which the job is to be executed is selected according to the rules listed under the **-u** option. The user submitting the job must be authorized to run the job under the execution user name. This authorization is provided if:

- The host on which qsub is run is trusted by the execution host (see /etc/hosts.equiv).
- The execution user has an .rhosts file naming the submitting user on the submitting host.

C-Shell .logout File:

The following warning applies for users of the c-shell, csh. If the job is executed under the csh and a .logout file exists in the home directory in which the job executes, the exit status of the job is that of the .logout script, not the job script. This may impact any inter-job dependencies. To preserve the job exit status, either remove the .logout file or place the following line as the first line in the .logout file:

set EXITVAL = $status

and the following line as the last executable line in .logout:

exit $EXITVAL

Interactive Jobs:

If the **-I** option is specified on the command line or in a script directive, or if the "interactive" job

attribute declared true via the **-W** option, `-W interactive=true`, either on the command line or in a script directive, the job is an interactive job. The script will be processed for directives, but will not be included with the job. When the job begins execution, all input to the job is from the terminal session in which qsub is running.

When an interactive job is submitted, the qsub command will not terminate when the job is submitted. qsub will remain running until the job terminates, is aborted, or the user interrupts qsub with an SIGINT (the control-C key). If qsub is interrupted prior to job start, it will query if the user wishes to exit. If the user response "yes", qsub exits and the job is aborted.

One the interactive job has started execution, input to and output from the job pass through qsub. Keyboard generated interrupts are passed to the job. Lines entered that begin with the tilde (**~**) character and contain special sequences are escaped by qsub. The recognized escape sequences are:

Sequence	Description
~.	qsub terminates execution. The batch job is also terminated.
~susp	Suspend the qsub program if running under the C shell. "susp" is the suspend character (usually CNTL-Z).
~asusp	Suspend the input half of qsub (terminal to job), but allow output to continue to be displayed. Only works under the C shell. "asusp" is the auxiliary suspend character, usually CNTL-Y.

Exit status

Upon successful processing, the qsub exit status will be a value of zero.

If the qsub command fails, the command exits with a value greater than zero.

Related topics

      qalter(1B)
      qdel(1B)
      qhold(1B)
      qrls(1B)
      qsig(1B)
      qstat(1B)
      pbs_server(8B)

Non-Adaptive Computing topics

      pbs_connect(3B)
      pbs_job_attributes(7B)
      pbs_queue_attributes(7B)
      pbs_resources_irix5(7B)
      pbs_resources_sp2(7B)
      pbs_resources_sunos4(7B)
      pbs_resources_unicos8(7B)

pbs_server_attributes(7B)
qselect(1B)
qmove(1B)
qmsg(1B)
qrerun(1B)