

**NAME**

**qsub** – submit a batch job to Grid Engine.  
**qsh** – submit an interactive X-windows session to Grid Engine.  
**qlogin** – submit an interactive login session to Grid Engine.  
**qrsh** – submit an interactive rsh session to Grid Engine.  
**qalter** – modify a pending or running batch job of Grid Engine.  
**qresub** – submit a copy of an existing Grid Engine job.

**SYNTAX**

**qsub** [ options ] [ command | -- [ command\_args ] ]  
**qsh** [ options ] [ -- xterm\_args ]  
**qlogin** [ options ]  
**qrsh** [ options ] [ command [ command\_args ] ]  
**qalter** [ options ] wc\_job\_range\_list [ -- [ command\_args ] ]  
**qalter** [ options ] -u user\_list | -uall [ -- [ command\_args ] ]  
**qresub** [ options ] job\_id\_list

**DESCRIPTION**

*Qsub* submits batch jobs to the Grid Engine queuing system. Grid Engine supports single- and multiple-node jobs. **Command** can be a path to a binary or a script (see **-b** below) which contains the commands to be run by the job using a shell (for example, *sh*(1) or *csch*(1)). Arguments to the command are given as **command\_args** to *qsub*. If **command** is handled as a script then it is possible to embed flags in the script. If the first two characters of a script line either match '#\$' or are equal to the prefix string defined with the **-C** option described below, the line is parsed for embedded command flags.

*Qsh* submits an interactive X-windows session to Grid Engine. An *xterm*(1) is brought up from the executing machine with the display directed either to the X-server indicated by the DISPLAY environment variable or as specified with the *-display qsh* option. Interactive jobs are not spooled if no resource is available to execute them. They are either dispatched to a suitable machine for execution immediately or the user submitting the job is notified by *qsh* that appropriate resources to execute the job are not available. **xterm\_args** are passed to the *xterm*(1) executable. Note, however, that the *-e* and *-ls* xterm options do not work with *qsh*.

*Qlogin* is similar to *qsh* in that it submits an interactive job to the queuing system. It does not open an *xterm*(1) window on the X display, but uses the current terminal for user I/O. Usually, *qlogin* establishes a *telnet*(1) connection with the remote host, using standard client- and server-side commands. These commands can be configured with the **qlogin\_daemon** (server-side, Grid Engine *telnetd* if not set, otherwise something like /usr/sbin/in.telnetd) and **qlogin\_command** (client-side, Grid Engine *telnet* if not set, otherwise something like /usr/bin/telnet) parameters in the global and local configuration settings of *ge\_conf*(5). The client side command is automatically parameterized with the remote host name and port number to which to connect, resulting in an invocation like

```
/usr/bin/telnet my_exec_host 2442
```

for example. *Qlogin* is invoked exactly like *qsh* and its jobs can only run on INTERACTIVE queues. *Qlogin* jobs can only be used if the *ge\_execd*(8) is running under the root account.

*Qrsh* is similar to *qlogin* in that it submits an interactive job to the queuing system. It uses the current terminal for user I/O. Usually, *qrsh* establishes a *rsh*(1) connection with the remote host. If no command is given to *qrsh*, an *rlogin*(1) session is established. The server-side commands used can be configured with the **rsh\_daemon** and **rlogin\_daemon** parameters in the global and local configuration settings of *ge\_conf*(5). A Grid Engine *rshd* or *rlogind* is used if the parameters are not set. If the parameters are set, they should be set to something like /usr/sbin/in.rshd or /usr/sbin/in.rlogind. On the client-side, the **rsh\_command** and **rlogin\_command** parameters can be set in the global and local configuration settings

of *ge\_conf*(5). If they are not set, special Grid Engine *rsh*(1) and *rlogin*(1) binaries delivered with Grid Engine are used. Use the cluster configuration parameters to integrate mechanisms like *ssh* or the *rsh*(1) and *rlogin*(1) facilities supplied with the operating system.

*Qrsh* jobs can only run in INTERACTIVE queues unless the option **-now no** is used (see below). They can also only be run, if the *ge\_execd*(8) is running under the root account.

*Qrsh* provides an additional useful feature for integrating with interactive tools providing a specific command shell. If the environment variable **QSSH\_WRAPPER** is set when *qrsh* is invoked, the command interpreter pointed to by **QSSH\_WRAPPER** will be executed to run *qrsh* commands instead of the users login shell or any shell specified in the *qrsh* command-line. The options **-cwd**, **-v**, **-V**, and **-display** only apply to batch jobs.

*Qalter* can be used to change the attributes of pending jobs. For array jobs with a mix of running and pending tasks (see the **-t** option below), modification with *qalter* only affects the pending tasks. *Qalter* can change most of the characteristics of a job (see the corresponding statements in the **OPTIONS** section below), including those which were defined as embedded flags in the script file (see above). Some submit options, such as the job script, cannot be changed with *qalter*.

*Qresub* allows the user to create jobs as copies of existing pending or running jobs. The copied jobs will have exactly the same attributes as the ones from which they were copied, except with a new job ID and with a cleared hold state. The only modification to the copied jobs supported by *qresub* is assignment of a new hold state with the **-h** option. This option can be used to first copy a job and then change its attributes via *qalter*.

Only a manager can use *qresub* on jobs submitted by another user. Regular users can only use *qresub* on their own jobs.

For *qsub*, *qsh*, *qrsh*, and *qlogin* the administrator and the user may define default request files (see *ge\_request*(5)) which can contain any of the options described below. If an option in a default request file is understood by *qsub* and *qlogin* but not by *qsh* the option is silently ignored if *qsh* is invoked. Thus you can maintain shared default request files for both *qsub* and *qsh*.

A cluster wide default request file may be placed under *\$GE\_ROOT/\$GE\_CELL/common/ge\_request*. User private default request files are processed under the locations *\$HOME/.ge\_request* and *\$cwd/.ge\_request*. The working directory local default request file has the highest precedence, then the home directory located file and then the cluster global file. The option arguments, the embedded script flags and the options in the default request files are processed in the following order:

- left to right in the script line,
- left to right in the default request files,
- from top to bottom of the script file (*qsub* only),
- from top to bottom of default request files,
- from left to right of the command line.

In other words, the command line can be used to override the embedded flags and the default request settings. The embedded flags, however, will override the default settings.

**Note**, that the **-clear** option can be used to discard any previous settings at any time in a default request file, in the embedded script flags, or in a command-line option. It is, however, not available with *qalter*.

The options described below can be requested either hard or soft. By default, all requests are considered hard until the **-soft** option (see below) is encountered. The hard/soft status remains in effect until its counterpart is encountered again. If all the hard requests for a job cannot be met, the job will not be scheduled. Jobs which cannot be run at the present time remain spooled.

## OPTIONS

### **-@ optionfile**

Forces *qsub*, *qrsh*, *qsh*, or *qlogin* to use the options contained in **optionfile**. The indicated file may contain all valid options. Comment lines must start with a **"#"** sign.

**-a date\_time**

Available for *qsub* and *qalter* only.

Defines or redefines the time and date at which a job is eligible for execution. **Date\_time** conforms to [[CC]]YY]MMDDhhmm[.SS], for the details, please see **Date\_time** in: *sge\_types*(1).

If this option is used with *qsub* or if a corresponding value is specified in *qmon* then a parameter named **a** and the value in the format CCYYMMDDhhmm.SS will be passed to the defined JSV instances (see **-jsv** option below or find more information concerning JSV in *jsv*(1))

**-ac variable[=value],...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Adds the given name/value pair(s) to the job's context. **Value** may be omitted. Grid Engine appends the given argument to the list of context variables for the job. Multiple **-ac**, **-dc**, and **-sc** options may be given. The order is important here.

The outcome of the evaluation of all **-ac**, **-dc**, and **-sc** options or corresponding values in *qmon* is passed to defined JSV instances as parameter with the name **ac**. (see **-jsv** option below or find more information concerning JSV in *jsv*(1)) *QALTER* allows changing this option even while the job executes.

**-ar ar\_id**

Available for *qsub*, *qalter*, *qrsh*, *qsh*, or *qlogin* only.

Assigns the submitted job to be a part of an existing Advance Reservation. The complete list of existing Advance Reservations can be obtained using the *qrstat*(1) command.

Note that the **-ar** option adds implicitly the **-w e** option if not otherwise requested.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **ar**. (see **-jsv** option below or find more information concerning JSV in *jsv*(1))

**-A account\_string**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Identifies the account to which the resource consumption of the job should be charged. The **account\_string** should conform to the **name** definition in *M sge\_types* 1 . In the absence of this parameter Grid Engine will place the default account string "ge" in the accounting record of the job.

*Qalter* allows changing this option even while the job executes.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **A**. (see **-jsv** option below or find more information concerning JSV in *jsv*(1))

**-binding [ binding\_instance ] binding\_strategy**

A job can request a specific processor core binding (processor affinity) with this parameter. This request is neither a hard nor a soft request, it is a hint for the execution host to do this if possible. Please note that the requested binding strategy is not used for resource selection within Grid Engine. As a result an execution host might be selected where Grid Engine does not even know the hardware topology and therefore is not able to apply the requested binding.

To enforce Grid Engine to select hardware on which the binding can be applied please use the **-l** switch in combination with the complex attribute **m\_topology**.

**binding\_instance** is an optional parameter. It might either be **env**, **pe** or **set** depending on which instance should accomplish the job to core binding. If the value for **binding\_instance** is not specified then **set** will be used.

**env** means that the environment variable **SGE\_BINDING** will be exported to the job environment of the job. This variable contains the selected operating system internal processor numbers. They might be more than selected cores in presence of SMT or CMT because each core could be represented by multiple processor identifiers. The processor numbers are space separated.

**pe** means that the information about the selected cores appears in the fourth column of the **pe\_hostfile**. Here the logical core and socket numbers are printed (they start at 0 and have no holes) in colon separated pairs (i.e. 0,0:1,0 which means core 0 on socket 0 and core 0 on socket 1). For more information about the **\$pe\_hostfile** check *ge\_pe*(5)

**set** (default if nothing else is specified). The binding strategy is applied by Grid Engine. How this is achieved depends on the underlying hardware architecture of the execution host where the submitted job will be started.

On Solaris 10 hosts a processor set will be created where the job can exclusively run in. Because of operating system limitations at least one core must remain unbound. This resource could of course be used by an unbound job.

On Linux hosts a processor affinity mask will be set to restrict the job to run exclusively on the selected cores. The operating system allows other unbound processes to use these cores. Please note that on Linux the binding requires a Linux kernel version of 2.6.16 or greater. It might be even possible to use a kernel with lower version number but in that case additional kernel patches have to be applied. The **loadcheck** tool in the utilbin directory can be used to check if the hosts capabilities. You can also use the **-sep** in combination with **-cb** of *qconf*(5) command to identify if Grid Engine is able to recognize the hardware topology.

Possible values for **binding\_strategy** are as follows:

```
linear:<amount>[:<socket>,<core>]
striding:<amount>:<n>[:<socket>,<core>]
explicit:[<socket>,<core>;...]<socket>,<core>
```

For the binding strategy linear and striding there is an optional socket and core pair attached. These denotes the mandatory starting point for the first core to bind on.

**linear** means that Grid Engine tries to bind the job on **amount** successive cores. If **socket** and **core** is omitted then Grid Engine first allocates successive cores on the first empty socket found. Empty means that there are no jobs bound to the socket by Grid Engine. If this is not possible or is not sufficient Grid Engine tries to find (further) cores on the socket with the most unbound cores and so on. If the amount of allocated cores is lower than requested cores, no binding is done for the job. If **socket** and **core** is specified then Grid Engine tries to find amount of empty cores beginning with this starting point. If this is not possible then binding is not done.

**striding** means that Grid Engine tries to find cores with a certain offset. It will select **amount** of empty cores with a offset of **n - 1** cores in between. Start point for the search algorithm is socket 0 core 0. As soon as **amount** cores are found they will be used to do the job binding. If there are not enough empty cores or if correct offset cannot be achieved then there will be no binding done.

**explicit** binds the specified sockets and cores that are mentioned in the provided socket/core list. Each socket/core pair has to be specified only once. If a socket/core pair is already in use by a different job the whole binding request will be ignored.

**Qalter** allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then these values will be passed to defined JSV instances as parameters with the names **binding\_strategy**, **binding\_type**, **binding\_amount**, **binding\_step**, **binding\_socket**, **binding\_core**, **binding\_exp\_n**, **binding\_exp\_socket**<id>, **binding\_exp\_core**<id>.

Please note that the length of the socket/core value list of the explicit binding is reported as **binding\_exp\_n**. <id> will be replaced by the position of the socket/core pair within the explicit list (0 <= id < **binding\_exp\_n**). The first socket/core pair of the explicit binding will be reported with the parameter names **binding\_exp\_socket0** and **binding\_exp\_core0**.

Values that do not apply for the specified binding will not be reported to JSV. E.g. **binding\_step** will only be reported for the striding binding and all **binding\_exp\_\*** values will be passed to JSV if explicit binding was specified. (see *-jsv* option below or find more information concerning JSV in *jsv(1)*)

#### **-b y[es]|n[o]**

Available for *qsub*, *qrsh* only. *Qalter* does not allow changing this option. This option cannot be embedded in the script file itself.

Gives the user the possibility to indicate explicitly whether **command** should be treated as binary or script. If the value of **-b** is 'y', then **command** may be a binary or script. The **command** might not be accessible from the submission host. Nothing except the path of the **command** will be transferred from the submission host to the execution host. Path aliasing will be applied to the path of **command** before **command** will be executed.

If the value of **-b** is 'n' then **command** needs to be a script and it will be handled as script. The script file has to be accessible by the submission host. It will be transferred to the execution host. *qsub/qrsh* will search directive prefixes within script.

*qsub* will implicitly use **-b n** whereas *qrsh* will apply the **-b y** option if nothing else is specified.

The value specified with this option or the corresponding value specified in *qmon* will only be passed to defined JSV instances if the value is *yes*. The name of the parameter will be **b**. The value will be **y** also when then long form **yes** was specified during submission. (see *-jsv* option below or find more information concerning JSV in *jsv(1)*)

Please note that submission of **command** as script (**-b n**) can have a significant performance impact, especially for short running jobs and big job scripts. Script submission adds a number of operations to the submission process: The job script needs to be

- parsed at client side (for special comments)
- transferred from submit client to qmaster
- spooled in qmaster
- transferred to execd at job execution
- spooled in execd
- removed from spooling both in execd and qmaster once the job is done

If job scripts are available on the execution nodes, e.g. via NFS, binary submission can be the better choice.

**-c occasion\_specifier**

Available for *qsub* and *qalter* only.

Defines or redefines whether the job should be checkpointed, and if so, under what circumstances. The specification of the checkpointing occasions with this option overwrites the definitions of the *when* parameter in the checkpointing environment (see *checkpoint(5)*) referenced by the *qsub -ckpt* switch. Possible values for **occasion\_specifier** are

n	no checkpoint is performed.
s	checkpoint when batch server is shut down.
m	checkpoint at minimum CPU interval.
x	checkpoint when job gets suspended.
<interval>	checkpoint in the specified time interval.

The minimum CPU interval is defined in the queue configuration (see *queue\_conf(5)* for details). <interval> has to be specified in the format hh:mm:ss. The maximum of <interval> and the queue's minimum CPU interval is used if <interval> is specified. This is done to ensure that a machine is not overloaded by checkpoints being generated too frequently.

The value specified with this option or the corresponding value specified in *qmon* will be passed to defined JSV instances. The <interval> will be available as parameter with the name **c\_interval**. The character sequence specified will be available as parameter with the name **c\_occasion**. Please note that if you change **c\_occasion** via JSV then the last setting of **c\_interval** will be overwritten and vice versa. (see **-jsv** option below or find more information concerning JSV in *jsv(1)*)

**-ckpt ckpt\_name**

Available for *qsub* and *qalter* only.

Selects the checkpointing environment (see *checkpoint(5)*) to be used for checkpointing the job. Also declares the job to be a checkpointing job.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **ckpt**. (see **-jsv** option below or find more information concerning JSV in *jsv(1)*)

**-clear** Available for *qsub*, *qsh*, *qrsh*, and *qlogin* only.

Causes all elements of the job to be reset to the initial default status prior to applying any modifications (if any) appearing in this specific command.

**-cwd** Available for *qsub*, *qsh*, *qrsh* and *qalter* only.

Execute the job from the current working directory. This switch will activate Grid Engine's path aliasing facility, if the corresponding configuration files are present (see *ge\_aliases(5)*).

In the case of *qalter*, the previous definition of the current working directory will be overwritten if *qalter* is executed from a different directory than the preceding *qsub* or *qalter*.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **cwd**. The value of this parameter will be the absolute path to the current working directory. JSV scripts can remove the path from jobs during the verification process by setting the value of this parameter to an empty string. As a result the job

behaves as if **-cwd** was not specified during job submission. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

### **-C prefix\_string**

Available for *qsub* and *qrsh* with script submission (**-b n**).

*Prefix\_string* defines the prefix that declares a directive in the job's command. The prefix is not a job attribute, but affects the behavior of *qsub* and *qrsh*. If **prefix** is a null string, the command will not be scanned for embedded directives.

The directive prefix consists of two ASCII characters which, when appearing in the first two bytes of a script line, indicate that what follows is an Grid Engine command. The default is "#\$".

The user should be aware that changing the first delimiting character can produce unforeseen side effects. If the script file contains anything other than a "#" character in the first byte position of the line, the shell processor for the job will reject the line and may exit the job prematurely.

If the -C option is present in the script file, it is ignored.

### **-dc variable,...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Removes the given variable(s) from the job's context. Multiple **-ac**, **-dc**, and **-sc** options may be given. The order is important.

*Qalter* allows changing this option even while the job executes.

The outcome of the evaluation of all **-ac**, **-dc**, and **-sc** options or corresponding values in *qmon* is passed to defined JSV instances as parameter with the name **ac**. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

### **-display display\_specifier**

Available for *qsh* and *qrsh*.

Directs *xterm* (1) to use **display\_specifier** in order to contact the X server. The **display\_specifier** has to contain the hostname part of the display name (e.g. myhost:1). Local display names (e.g. :0) cannot be used in grid environments. Values set with the **-display** option overwrite settings from the submission environment and from **-v** command line options.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **display**. This value will also be available in the job environment which might optionally be passed to JSV scripts. The variable name will be **DISPLAY**. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

### **-dl date\_time**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Specifies the deadline initiation time in [[CC]YY]MMDDhhmm[.SS] format (see **-a** option above). The deadline initiation time is the time at which a deadline job has to reach top priority to be able to complete within a given deadline. Before the deadline initiation time the priority of a deadline job will be raised steadily until it reaches the maximum as configured by the Grid Engine administrator.

This option is applicable only for users allowed to submit deadline jobs.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **dl**. The format for the date\_time value is CCYYMMDDhhmm.SS (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

**-e [[hostname]:]path,...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Defines or redefines the path used for the standard error stream of the job. For *qsh*, *qrsh* and *qlogin* only the standard error stream of prolog and epilog is redirected. If the **path** constitutes an absolute path name, the error-path attribute of the job is set to **path**, including the **hostname**. If the path name is relative, Grid Engine expands **path** either with the current working directory path (if the **-cwd** switch (see above) is also specified) or with the home directory path. If **hostname** is present, the standard error stream will be placed in the corresponding location only if the job runs on the specified host. If the path contains a ":" without a **hostname**, a leading ":" has to be specified.

By default the file name for interactive jobs is */dev/null*. For batch jobs the default file name has the form *job\_name.ejob\_id* and *job\_name.ejob\_id.task\_id* for array job tasks (see **-t** option below).

If **path** is a directory, the standard error stream of the job will be put in this directory under the default file name. If the pathname contains certain pseudo environment variables, their value will be expanded at runtime of the job and will be used to constitute the standard error stream path name. The following pseudo environment variables are supported currently:

\$HOME	home directory on execution machine
\$USER	user ID of job owner
\$JOB_ID	current job ID
\$JOB_NAME	current job name (see <b>-N</b> option)
\$HOSTNAME	name of the execution host
\$TASK_ID	array job task index number

Alternatively to \$HOME the tilde sign "~" can be used as common in *cs*(1) or *ks*(1). Note, that the "~" sign also works in combination with user names, so that "~<user>" expands to the home directory of <user>. Using another user ID than that of the job owner requires corresponding permissions, of course.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **e**. (see **-jsv** option below or find more information concerning JSV in *jsv*(1))

**-hard** Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Signifies that all **-q** and **-l** resource requirements following in the command line will be hard requirements and must be satisfied in full before a job can be scheduled.

As Grid Engine scans the command line and script file for Grid Engine options and parameters it builds a list of resources required by a job. All such resource requests are considered as absolutely essential for the job to commence. If the **-soft** option (see below) is encountered during the scan then all following resources are designated as "soft requirements" for execution, or "nice-to-have, but not essential". If the **-hard** flag is encountered at a later stage of the scan, all resource requests following it once again become "essential". The **-hard** and **-soft** options in effect act as "toggles" during the scan.

If this option or a corresponding value in *qmon* is specified then the corresponding **-q** and **-l** resource requirements will be passed to defined JSV instances as parameter with the names **q\_hard** and **l\_hard**. Find for information in the sections describing **-q** and **-l**. (see **-jsv** option



below or find more information concerning JSV in *jsv*(1))

**-h | -h {u|s|o|n|U|O|S}..**

Available for *qsub* (only **-h**), *qrsh*, *qalter* and *qresub* (hold state is removed when not set explicitly).

List of holds to place on a job, a task or some tasks of a job.

‘u’ denotes a user hold.

‘s’ denotes a system hold.

‘o’ denotes a operator hold.

‘n’ denotes no hold (requires manager privileges).

As long as any hold other than ‘n’ is assigned to the job the job is not eligible for execution. Holds can be released via *qalter* and *qrls*(1). In case of *qalter* this is supported by the following additional option specifiers for the **-h** switch:

‘U’ removes a user hold.

‘S’ removes a system hold.

‘O’ removes a operator hold.

Grid Engine managers can assign and remove all hold types, Grid Engine operators can assign and remove user and operator holds, and users can only assign or remove user holds.

In the case of *qsub* only user holds can be placed on a job and thus only the first form of the option with the **-h** switch alone is allowed. As opposed to this, *qalter* requires the second form described above.

An alternate means to assign hold is provided by the *qhold*(1) facility.

If the job is a array job (see the **-t** option below), all tasks specified via **-t** are affected by the **-h** operation simultaneously.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option is specified with *qsub* or during the submission of a job in *qmon* then the parameter **h** with the value **u** will be passed to the defined JSV instances indicating that the job will be in user hold after the submission finishes. (see **-jsv** option below or find more information concerning JSV in *jsv*(1))

**-help** Prints a listing of all options.

**-hold\_jid wc\_job\_list**

Available for *qsub*, *qrsh*, and *qalter* only. See *sgc\_types*(1). for **wc\_job\_list** definition.

Defines or redefines the job dependency list of the submitted job. A reference by job name or pattern is only accepted if the referenced job is owned by the same user as the referring job. The submitted job is not eligible for execution unless all jobs referenced in the comma-separated job id and/or job name list have completed. If any of the referenced jobs exits with exit code 100, the submitted job will remain ineligible for execution.

With the help of job names or regular pattern one can specify a job dependency on multiple jobs satisfying the regular pattern or on all jobs with the requested name. The name dependencies are resolved at submit time and can only be changed via *qalter*. New jobs or name changes of other jobs will not be taken into account.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **hold\_jid**. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

#### **-hold\_jid\_ad wc\_job\_list**

Available for *qsub*, *qrsh*, and *qalter* only. See *sgen\_types* (1). for **wc\_job\_list** definition.

Defines or redefines the job array dependency list of the submitted job. A reference by job name or pattern is only accepted if the referenced job is owned by the same user as the referring job. Each sub-task of the submitted job is not eligible for execution unless the corresponding sub-tasks of all jobs referenced in the comma-separated job id and/or job name list have completed. If any array task of the referenced jobs exits with exit code 100, the dependent tasks of the submitted job will remain ineligible for execution.

With the help of job names or regular pattern one can specify a job dependency on multiple jobs satisfying the regular pattern or on all jobs with the requested name. The name dependencies are resolved at submit time and can only be changed via *qalter*. New jobs or name changes of other jobs will not be taken into account.

If either the submitted job or any job in **wc\_job\_list** are not array jobs with the same range of sub-tasks (see **-t** option below), the request list will be rejected and the job create or modify operation will error.

*qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **hold\_jid\_ad**. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

#### **-i [[hostname]:]file,...**

Available for *qsub*, and *qalter* only.

Defines or redefines the file used for the standard input stream of the job. If the *file* constitutes an absolute filename, the input-path attribute of the job is set to **path**, including the **hostname**. If the path name is relative, Grid Engine expands **path** either with the current working directory path (if the **-cwd** switch (see above) is also specified) or with the home directory path. If **hostname** is present, the standard input stream will be placed in the corresponding location only if the job runs on the specified host. If the path contains a ":" without a **hostname**, a leading ":" has to be specified.

By default /dev/null is the input stream for the job.

It is possible to use certain pseudo variables, whose values will be expanded at runtime of the job and will be used to express the standard input stream as described in the **-e** option for the standard error stream.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **i**. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

**-inherit**

Available only for *qrsh* and *qmake* (1).

*qrsh* allows the user to start a task in an already scheduled parallel job. The option **-inherit** tells *qrsh* to read a job id from the environment variable `JOB_ID` and start the specified command as a task in this job. Please note that in this case, the hostname of the host where the command will be executed must precede the command to execute; the syntax changes to

**qrsh -inherit [ other options ] hostname command [ command\_args ]**

Note also, that in combination with **-inherit**, most other command line options will be ignored. Only the options **-verbose**, **-v** and **-V** will be interpreted. As a replacement to option **-cwd** please use **-v PWD**.

Usually a task should have the same environment (including the current working directory) as the corresponding job, so specifying the option **-V** should be suitable for most applications.

*Note:* If in your system the qmaster tcp port is not configured as a service, but rather via the environment variable `GE_QMASTER_PORT`, make sure that this variable is set in the environment when calling *qrsh* or *qmake* with the **-inherit** option. If you call *qrsh* or *qmake* with the **-inherit** option from within a job script, export `GE_QMASTER_PORT` with the option "-v GE\_QMASTER\_PORT" either as a command argument or an embedded directive.

This parameter is not available in the JSV context. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

**-j y[es]|n[o]**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Specifies whether or not the standard error stream of the job is merged into the standard output stream.

If both the **-j y** and the **-e** options are present, Grid Engine sets but ignores the error-path attribute.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

The value specified with this option or the corresponding value specified in *qmon* will only be passed to defined JSV instances if the value is *yes*. The name of the parameter will be **j**. The value will be **y** also when then long form **yes** was specified during submission. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

**-js job\_share**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Defines or redefines the job share of the job relative to other jobs. Job share is an unsigned integer value. The default job share value for jobs is 0.

The job share influences the Share Tree Policy and the Functional Policy. It has no effect on the Urgency and Override Policies (see *share\_tree* (5), *sched\_conf* (5) and the *Grid Engine Installation and Administration Guide* for further information on the resource management policies supported by Grid Engine).

In case of the Share Tree Policy, users can distribute the tickets to which they are currently entitled among their jobs using different shares assigned via **-js**. If all jobs have the same job share value,

the tickets are distributed evenly. Otherwise, jobs receive tickets relative to the different job shares. Job shares are treated like an additional level in the share tree in the latter case.

In connection with the Functional Policy, the job share can be used to weight jobs within the functional job category. Tickets are distributed relative to any uneven job share distribution treated as a virtual share distribution level underneath the functional job category.

If both the Share Tree and the Functional Policy are active, the job shares will have an effect in both policies, and the tickets independently derived in each of them are added to the total number of tickets for each job.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **js**. (see **-jsv** option below or find more information concerning JSV in *jsv* (1))

#### **-jsv jsv\_url**

Available for *qsub*, *qsh*, *qrsh* and *qlogin* only.

Defines a client JSV instance which will be executed to verify the job specification before the job is sent to qmaster.

In contrast to other options this switch will not be overwritten if it is also used in *sge\_request* files. Instead all specified JSV instances will be executed to verify the job to be submitted.

The JSV instance which is directly passed with the commandline of a client is executed as first to verify the job specification. After that the JSV instance which might have been defined in various *sge\_request* files will be triggered to check the job. Find more details in man page *jsv* (1) and *sge\_request* (5).

The syntax of the **jsv\_url** is specified in *sge\_types*(1). ()

#### **-l resource=value,...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Launch the job in a Grid Engine queue meeting the given resource request list. In case of *qalter* the previous definition is replaced by the specified one.

*complex* (5) describes how a list of available resources and their associated valid value specifiers can be obtained.

There may be multiple **-l** switches in a single command. You may request multiple **-l** options to be soft or hard both in the same command line. In case of a serial job multiple **-l** switches refine the definition for the sought queue.

*Qalter* allows changing the value of this option even while the job is running, but only if the initial list of resources does not contain a resource that is marked as consumable. However the modification will only be effective after a restart or migration of the job.

If this option or a corresponding value in *qmon* is specified the these hard and soft resource requirements will be passed to defined JSV instances as parameter with the names **l\_hard** and **l\_soft**. If regular expressions will be used for resource requests, then these expressions will be passed as they are. Also shortcut names will not be expanded. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

**-m b|e|a|s|n,...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Defines or redefines under which circumstances mail is to be sent to the job owner or to the users defined with the **-M** option described below. The option arguments have the following meaning:

- 'b' Mail is sent at the beginning of the job.
- 'e' Mail is sent at the end of the job.
- 'a' Mail is sent when the job is aborted or rescheduled.
- 's' Mail is sent when the job is suspended.
- 'n' No mail is sent.

Currently no mail is sent when a job is suspended.

*Qalter* allows changing the b, e, and a option arguments even while the job executes. The modification of the b option argument will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **m**. (see **-jsv** option above or find more information concerning JSV in

**-M user[@host],...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Defines or redefines the list of users to which the server that executes the job has to send mail, if the server sends mail about the job. Default is the job owner at the originating host.

*Qalter* allows changing this option even while the job executes.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **M**. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

**-masterq wc\_queue\_list**

Available for *qsub*, *qrsh*, *qsh*, *qlogin* and *qalter*. Only meaningful for parallel jobs, i.e. together with the **-pe** option.

Defines or redefines a list of cluster queues, queue domains and queue instances which may be used to become the so called *master queue* of this parallel job. A more detailed description of *wc\_queue\_list* can be found in *sge\_types* (1). The *master queue* is defined as the queue where the parallel job is started. The other queues to which the parallel job spawns tasks are called *slave queues*. A parallel job only has one *master queue*.

This parameter has all the properties of a resource request and will be merged with requirements derived from the **-l** option described above.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this hard resource requirement will be passed to defined JSV instances as parameter with the name **masterq**. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

**-notify** Available for *qsub*, *qrsh* (with command) and *qalter* only.

This flag, when set causes Grid Engine to send "warning" signals to a running job prior to sending the signals themselves. If a SIGSTOP is pending, the job will receive a SIGUSR1 several seconds before the SIGSTOP. If a SIGKILL is pending, the job will receive a SIGUSR2 several seconds before the SIGKILL. This option provides the running job, before receiving the SIGSTOP or SIGKILL, a configured time interval to do e.g. cleanup operations. The amount of time delay is controlled by the **notify** parameter in each queue configuration (see *queue\_conf*(5)).

Note that the Linux operating system "misused" the user signals SIGUSR1 and SIGUSR2 in some early Posix thread implementations. You might not want to use the **-notify** option if you are running multi-threaded applications in your jobs under Linux, particularly on 2.0 or earlier kernels.

*Qalter* allows changing this option even while the job executes.

Only if this option is used the parameter named **notify** with the value **y** will be passed to defined JSV instances. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-now y[es][n[o]]**

Available for *qsub*, *qsh*, *qlogin* and *qrsh*.

**-now y** tries to start the job immediately or not at all. The command returns 0 on success, or 1 on failure (also if the job could not be scheduled immediately). For array jobs submitted with the **-now** option, if all tasks cannot be immediately scheduled, no tasks are scheduled. **-now y** is default for *qsh*, *qlogin* and *qrsh*

With the **-now n** option, the job will be put into the pending queue if it cannot be executed immediately. **-now n** is default for *qsub*.

The value specified with this option or the corresponding value specified in *qmon* will only be passed to defined JSV instances if the value is *yes*. The name of the parameter will be **now**. The value will be **y** also when then long form **yes** was specified during submission. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-N name**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

The name of the job. The name should follow the "**name**" definition in *sge\_types*(1). Invalid job names will be denied at submit time.

If the **-N** option is not present, Grid Engine assigns the name of the job script to the job after any directory pathname has been removed from the script-name. If the script is read from standard input, the job name defaults to STDIN.

In the case of *qsh* or *qlogin* with the **-N** option is absent, the string 'INTERACT' is assigned to the job.

In the case of *qrsh* if the **-N** option is absent, the resulting job name is determined from the *qrsh* command line by using the argument string up to the first occurrence of a semicolon or whitespace and removing the directory pathname.

*Qalter* allows changing this option even while the job executes.

The value specified with this option or the corresponding value specified in *qmon* will be passed to defined JSV instances as parameter with the name *N*. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-noshell**

Available only for *qrsh* with a command line.

Do not start the command line given to *qrsh* in a user's login shell, i.e. execute it without the wrapping shell.

This option can be used to speed up execution as some overhead, like the shell startup and sourcing the shell resource files, is avoided.

This option can only be used if no shell-specific command line parsing is required. If the command line contains shell syntax like environment variable substitution or (back) quoting, a shell must be started. In this case, either do not use the **-noshell** option or include the shell call in the command line.

Example:

```
qrsh echo '$HOSTNAME'
```

Alternative call with the **-noshell** option

```
qrsh -noshell /bin/tcsh -f -c 'echo $HOSTNAME'
```

**-nostdin**

Available only for *qrsh*.

Suppress the input stream STDIN - *qrsh* will pass the option **-n** to the *rsh*(1) command. This is especially useful, if multiple tasks are executed in parallel using *qrsh*, e.g. in a *make*(1) process - it would be undefined, which process would get the input.

**-o [[hostname]:]path,...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

The path used for the standard output stream of the job. The **path** is handled as described in the **-e** option for the standard error stream.

By default the file name for standard output has the form *job\_name.ojob\_id* and *job\_name.ojob\_id.task\_id* for array job tasks (see **-t** option below).

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **o**. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-ot override\_tickets**

Available for *qalter* only.

Changes the number of override tickets for the specified job. Requires manager/operator privileges.

**-P project\_name**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Specifies the project to which this job is assigned. The administrator needs to give permission to individual users to submit jobs to a specific project. (see **-apri** option to *qconf*(1)).

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **ot**. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-p priority**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Defines or redefines the priority of the job relative to other jobs. Priority is an integer in the range -1023 to 1024. The default priority value for jobs is 0.

Users may only decrease the priority of their jobs. Grid Engine managers and administrators may also increase the priority associated with jobs. If a pending job has higher priority, it is earlier eligible for being dispatched by the Grid Engine scheduler.

If this option or a corresponding value in *qmon* is specified and the priority is not 0 then this value will be passed to defined JSV instances as parameter with the name **p**. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

**-pe parallel\_environment n[-m][[-m],...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Parallel programming environment (PE) to instantiate. For more detail about PEs, please see the *sge\_types* (1).

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then the parameters **pe\_name**, **pe\_min** and **pe\_max** will be passed to configured JSV instances where **pe\_name** will be the name of the parallel environment and the values **pe\_min** and **pe\_max** represent the values n and m which have been provided with the **-pe** option. A missing specification of m will be expanded as value 9999999 in JSV scripts and it represents the value infinity. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

**-pty y[es]|n[o]**

Available for *qrsh* and *qlogin* only.

-pty yes enforces the job to be started in a pseudo terminal (pty). If no pty is available, the job start fails. -pty no enforces the job to be started without a pty. By default, *qrsh without a command* and *qlogin* start the job in a pty, *qrsh with a command* starts the job without a pty.

This parameter is not available in the JSV context. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

**-q wc\_queue\_list**

Available for *qsub*, *qrsh*, *qsh*, *qlogin* and *qalter*.

Defines or redefines a list of cluster queues, queue domains or queue instances which may be used to execute this job. Please find a description of *wc\_queue\_list* in *sge\_types* (1). This parameter has all the properties of a resource request and will be merged with requirements derived from the **-l** option described above.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified the these hard and soft resource requirements will be passed to defined JSV instances as parameters with the names **q\_hard** and **q\_soft**. If regular expressions will be used for resource requests, then these expressions will be passed as they are. Also shortcut names will not be expanded. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))



**-R y[es]|n[o]**

Available for *qsub*, *qrsh*, *qsh*, *qlogin* and *qalter*.

Indicates whether a reservation for this job should be done. Reservation is never done for immediate jobs, i.e. jobs submitted using the **-now yes** option. Please note that regardless of the reservation request, job reservation might be disabled using `max_reservation` in *sched\_conf*(5) and might be limited only to a certain number of high priority jobs.

By default jobs are submitted with the **-R n** option.

The value specified with this option or the corresponding value specified in *qmon* will only be passed to defined JSV instances if the value is *yes*. The name of the parameter will be **R**. The value will be **y** also when then long form **yes** was specified during submission. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-r y[es]|n[o]**

Available for *qsub* and *qalter* only.

Identifies the ability of a job to be rerun or not. If the value of **-r** is 'yes', the job will be rerun if the job was aborted without leaving a consistent exit state. (This is typically the case if the node on which the job is running crashes). If **-r** is 'no', the job will not be rerun under any circumstances.

Interactive jobs submitted with *qsh*, *qrsh* or *qlogin* are not rerunnable.

*Qalter* allows changing this option even while the job executes.

The value specified with this option or the corresponding value specified in *qmon* will only be passed to defined JSV instances if the value is *yes*. The name of the parameter will be **r**. The value will be **y** also when then long form **yes** was specified during submission. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-sc variable[=value],...**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Sets the given name/value pairs as the job's context. **Value** may be omitted. Grid Engine replaces the job's previously defined context with the one given as the argument. Multiple **-ac**, **-dc**, and **-sc** options may be given. The order is important.

Contexts provide a way to dynamically attach and remove meta-information to and from a job. The context variables are **not** passed to the job's execution context in its environment.

*Qalter* allows changing this option even while the job executes.

The outcome of the evaluation of all **-ac**, **-dc**, and **-sc** options or corresponding values in *qmon* is passed to defined JSV instances as parameter with the name **ac**. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-shell y[es]|n[o]**

Available only for *qsub*.

**-shell n** causes *qsub* to execute the command line directly, as if by *exec*(2). No command shell will be executed for the job. This option only applies when **-b y** is also used. Without **-b y**, **-shell n** has no effect.

This option can be used to speed up execution as some overhead, like the shell startup and sourcing the shell resource files is avoided.

This option can only be used if no shell-specific command line parsing is required. If the command line contains shell syntax, like environment variable substitution or (back) quoting, a shell must be started. In this case either do not use the **-shell n** option or execute the shell as the command line and pass the path to the executable as a parameter.

If a job executed with the **-shell n** option fails due to a user error, such as an invalid path to the executable, the job will enter the error state.

**-shell y** cancels the effect of a previous **-shell n**. Otherwise, it has no effect.

See **-b** and **-noshell** for more information.

The value specified with this option or the corresponding value specified in *qmon* will only be passed to defined JSV instances if the value is *yes*. The name of the parameter will be **shell**. The value will be **y** also when then long form **yes** was specified during submission. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-soft** Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter* only.

Signifies that all resource requirements following in the command line will be soft requirements and are to be filled on an "as available" basis.

As Grid Engine scans the command line and script file for Grid Engine options and parameters, it builds a list of resources required by the job. All such resource requests are considered as absolutely essential for the job to commence. If the **-soft** option is encountered during the scan then all following resources are designated as "soft requirements" for execution, or "nice-to-have, but not essential". If the **-hard** flag (see above) is encountered at a later stage of the scan, all resource requests following it once again become "essential". The **-hard** and **-soft** options in effect act as "toggles" during the scan.

If this option or a corresponding value in *qmon* is specified then the corresponding **-q** and **-l** resource requirements will be passed to defined JSV instances as parameter with the names **q\_soft** and **l\_soft**. Find for information in the sections describing **-q** and **-l**. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-sync y[es][n[o]]**

Available for *qsub*.

**-sync y** causes *qsub* to wait for the job to complete before exiting. If the job completes successfully, *qsub*'s exit code will be that of the completed job. If the job fails to complete successfully, *qsub* will print out a error message indicating why the job failed and will have an exit code of 1. If *qsub* is interrupted, e.g. with CTRL-C, before the job completes, the job will be canceled.

With the **-sync n** option, *qsub* will exit with an exit code of 0 as soon as the job is submitted successfully. **-sync n** is default for *qsub*.

If **-sync y** is used in conjunction with **-now y**, *qsub* will behave as though only **-now y** were given until the job has been successfully scheduled, after which time *qsub* will behave as though only **-sync y** were given.

If **-sync y** is used in conjunction with **-t n[-m[:i]]**, *qsub* will wait for all the job's tasks to complete before exiting. If all the job's tasks complete successfully, *qsub*'s exit code will be that of the first completed job tasks with a non-zero exit code, or 0 if all job tasks exited with an exit code of 0. If any of the job's tasks fail to complete successfully, *qsub* will print out an error message indicating why the job task(s) failed and will have an exit code of 1. If *qsub* is interrupted, e.g. with CTRL-C, before the job completes, all of the job's tasks will be canceled.

Information that this switch was specified during submission is not available in the JSV context. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-S [[hostname]:]pathname,...**

Available for *qsub*, *qsh* and *qalter*.

Specifies the interpreting shell for the job. Only one **pathname** component without a **host** specifier is valid and only one path name for a given host is allowed. Shell paths with host assignments define the interpreting shell for the job if the host is the execution host. The shell path without host specification is used if the execution host matches none of the hosts in the list.

Furthermore, the pathname can be constructed with pseudo environment variables as described for the **-e** option above.

In the case of *qsh* the specified shell path is used to execute the corresponding command interpreter in the *xterm*(1) (via its **-e** option) started on behalf of the interactive job. *Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameter with the name **S**. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-t n[-m[:s]]**

Available for *qsub* and *qalter* only.

Submits a so called *Array Job*, i.e. an array of identical tasks being differentiated only by an index number and being treated by Grid Engine almost like a series of jobs. The option argument to **-t** specifies the number of array job tasks and the index number which will be associated with the tasks. The index numbers will be exported to the job tasks via the environment variable **GE\_TASK\_ID**. The option arguments n, m and s will be available through the environment variables **GE\_TASK\_FIRST**, **GE\_TASK\_LAST** and **GE\_TASK\_STEPSIZE**.

Following restrictions apply to the values n and m:

$$\begin{aligned} 1 &\leq n \leq \text{MIN}(2^{31}-1, \text{max\_aj\_tasks}) \\ 1 &\leq m \leq \text{MIN}(2^{31}-1, \text{max\_aj\_tasks}) \\ n &\leq m \end{aligned}$$

*max\_aj\_tasks* is defined in the cluster configuration (see *sge\_conf*(5))

The task id range specified in the option argument may be a single number, a simple range of the form n-m or a range with a step size. Hence, the task id range specified by 2-10:2 would result in the task id indexes 2, 4, 6, 8, and 10, for a total of 5 identical tasks, each with the environment variable **GE\_TASK\_ID** containing one of the 5 index numbers.

All array job tasks inherit the same resource requests and attribute definitions as specified in the *qsub* or *qalter* command line, except for the **-t** option. The tasks are scheduled independently and, provided enough resources exist, concurrently, very much like separate jobs. However, an array job or a sub-array thereof can be accessed as a single unit by commands like *qmod*(1) or *qdel*(1). See the corresponding manual pages for further detail.

Array jobs are commonly used to execute the same type of operation on varying input data sets correlated with the task index number. The number of tasks in a array job is unlimited.

STDOUT and STDERR of array job tasks will be written into different files with the default location

<jobname>.[ 'e' 'o' ]<job\_id>'. '<task\_id>

In order to change this default, the **-e** and **-o** options (see above) can be used together with the pseudo environment variables \$HOME, \$USER, \$JOB\_ID, \$JOB\_NAME, \$HOSTNAME, and \$GE\_TASK\_ID.

Note, that you can use the output redirection to divert the output of all tasks into the same file, but the result of this is undefined.

If this option or a corresponding value in *qmon* is specified then this value will be passed to defined JSV instances as parameters with the name **t\_min**, **t\_max** and **t\_step** (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

#### **-tc max\_running\_tasks**

-allow users to limit concurrent array job task execution. Parameter *max\_running\_tasks* specifies maximum number of simultaneously running tasks. For example we have running SGE with 10 free slots. We call *qsub -t 1-100 -tc 2 jobscript*. Then only 2 tasks will be scheduled to run even when 8 slots are free.

#### **-terse** Available for *qsub* only.

-terse causes the *qsub* to display only the job-id of the job being submitted rather than the regular "Your job ..." string. In case of an error the error is reported on stderr as usual. This can be helpful for scripts which need to parse *qsub* output to get the job-id.

Information that this switch was specified during submission is not available in the JSV context. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

#### **-u username,...**

Available for *qalter* only. Changes are only made on those jobs which were submitted by users specified in the list of usernames. For managers it is possible to use the **qalter -u '\*'** command to modify all jobs of all users.

If you use the **-u** switch it is not permitted to specify an additional *wc\_job\_range\_list*.

#### **-v variable[=value],...**

Available for *qsub*, *qrsh* (with command argument) and *qalter*.

Defines or redefines the environment variables to be exported to the execution context of the job. If the **-v** option is present Grid Engine will add the environment variables defined as arguments to the switch and, optionally, values of specified variables, to the execution context of the job.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

All environment variables specified with **-v**, **-V** or the DISPLAY variable provided with **-display** will be exported to the defined JSV instances only optionally when this is requested explicitly during the job submission verification. (see **-jsv** option above or find more information concerning JSV in *jsv* (1))

#### **-verbose**

Available only for *qrsh* and *qmake* (1).

Unlike *qsh* and *qlogin*, *qrsh* does not output any informational messages while establishing the session, compliant with the standard *rsh* (1) and *rlogin* (1) system calls. If the option **-verbose** is set, *qrsh* behaves like the *qsh* and *qlogin* commands, printing information about the process of establishing the *rsh* (1) or *rlogin* (1) session.

**-verify** Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter*.

Instead of submitting a job, prints detailed information about the would-be job as though *qstat*(1) -j were used, including the effects of command-line parameters and the external environment.

**-V** Available for *qsub*, *qsh*, *qrsh* with *command* and *qalter*.

Specifies that all environment variables active within the *qsub* utility be exported to the context of the job.

All environment variables specified with **-v**, **-V** or the DISPLAY variable provided with **-display** will be exported to the defined JSV instances only optionally when this is requested explicitly during the job submission verification. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**-w e|w|n|p|v**

Available for *qsub*, *qsh*, *qrsh*, *qlogin* and *qalter*.

Specifies a validation level applied to the job to be submitted (*qsub*, *qlogin*, and *qsh*) or the specified queued job (*qalter*). The information displayed indicates whether the job can possibly be scheduled assuming an empty system with no other jobs. Resource requests exceeding the configured maximal thresholds or requesting unavailable resource attributes are possible causes for jobs to fail this validation.

The specifiers e, w, n and v define the following validation modes:

- 'e' error - jobs with invalid requests will be rejected.
- 'w' warning - only a warning will be displayed for invalid requests.
- 'n' none - switches off validation; the default for *qsub*, *qalter*, *qrsh*, *qsh* and *qlogin*.
- 'p' poke - does not submit the job but prints a validation report based on a cluster as is with all resource utilizations in place.
- 'v' verify - does not submit the job but prints a validation report based on an empty cluster.

Note, that the necessary checks are performance consuming and hence the checking is switched off by default. It should also be noted that load values are not taken into account with the verification since they are assumed to be too volatile. To cause -w e verification to be passed at submission time, it is possible to specify non-volatile values (non-consumables) or maximum values (consumables) in *complex\_values*.

**-wd working\_dir**

Available for *qsub*, *qsh*, *qrsh* and *qalter* only.

Execute the job from the directory specified in *working\_dir*. This switch will activate Grid Engine's path aliasing facility, if the corresponding configuration files are present (see *ge\_aliases*(5)).

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however. The parameter value will be available in defined JSV instances as parameter with the name **cwd** (see **-cwd** switch above or find more information concerning JSV in *jsv*(1))

**command**

Available for *qsub* and *qrsh* only.

The job's scriptfile or binary. If not present or if the operand is the single-character string '-', *qsub* reads the script from standard input.

The command will be available in defined JSV instances as parameter with the name **CMDNAME** (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**command\_args**

Available for *qsub*, *qrsh* and *qalter* only.

Arguments to the job. Not valid if the script is entered from standard input.

*Qalter* allows changing this option even while the job executes. The modified parameter will only be in effect after a restart or migration of the job, however.

The number of command arguments is provided to configured JSV instances as parameter with the name **CMDARGS**. Also the argument values can be accessed. Argument names have the format **CMDARG<number>** where **<number>** is a integer between 0 and **CMDARGS - 1**. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**xterm\_args**

Available for *qsh* only.

Arguments to the *xterm*(1) executable, as defined in the configuration. For details, refer to *ge\_conf*(5)).

Information concerning **xterm\_args** will be available in JSV context as parameters with the name **CMDARGS** and **CMDARG<number>**. Find more information above in section **command\_args**. (see **-jsv** option above or find more information concerning JSV in *jsv*(1))

**ENVIRONMENTAL VARIABLES**

**GE\_ROOT** Specifies the location of the Grid Engine standard configuration files.

**GE\_CELL** If set, specifies the default Grid Engine cell. To address a Grid Engine cell *qsub*, *qsh*, *qlogin* or *qalter* use (in the order of precedence):

The name of the cell specified in the environment variable **GE\_CELL**, if it is set.

The name of the default cell, i.e. **default**.

**GE\_DEBUG\_LEVEL** If set, specifies that debug information should be written to stderr. In addition the level of detail in which debug information is generated is defined.

**GE\_QMASTER\_PORT** If set, specifies the tcp port on which *ge\_qmaster*(8) is expected to listen for communication requests. Most installations will use a services map entry for the service "sge\_qmaster" instead to define that port.

**DISPLAY** For *qsh* jobs the **DISPLAY** has to be specified at job submission. If the **DISPLAY** is not set by using the **-display** or the **-v** switch, the contents of the **DISPLAY** environment variable are used as default.

In addition to those environment variables specified to be exported to the job via the **-v** or the **-V** option (see above) *qsub*, *qsh*, and *qlogin* add the following variables with the indicated values to the variable list:

<b>GE_O_HOME</b>	the home directory of the submitting client.
<b>GE_O_HOST</b>	the name of the host on which the submitting client is running.
<b>GE_O_LOGNAME</b>	the LOGNAME of the submitting client.
<b>GE_O_MAIL</b>	the MAIL of the submitting client. This is the mail directory of the submitting client.
<b>GE_O_PATH</b>	the executable search path of the submitting client.
<b>GE_O_SHELL</b>	the SHELL of the submitting client.
<b>GE_O_TZ</b>	the time zone of the submitting client.
<b>GE_O_WORKDIR</b>	the absolute path of the current working directory of the submitting client.

Furthermore, Grid Engine sets additional variables into the job's environment, as listed below.

#### ARC

<b>SGE_ARCH</b>	The Grid Engine architecture name of the node on which the job is running. The name is compiled-in into the <i>ge_execd</i> (8) binary.
<b>GE_CKPT_ENV</b>	Specifies the checkpointing environment (as selected with the <b>-ckpt</b> option) under which a checkpointing job executes. Only set for checkpointing jobs.
<b>GE_CKPT_DIR</b>	Only set for checkpointing jobs. Contains path <i>ckpt_dir</i> (see <i>checkpoint</i> (5) ) of the checkpoint interface.
<b>GE_STDERR_PATH</b>	the pathname of the file to which the standard error stream of the job is diverted. Commonly used for enhancing the output with error messages from prolog, epilog, parallel environment start/stop or checkpointing scripts.
<b>GE_STDOUT_PATH</b>	the pathname of the file to which the standard output stream of the job is diverted. Commonly used for enhancing the output with messages from prolog, epilog, parallel environment start/stop or checkpointing scripts.
<b>GE_STDIN_PATH</b>	the pathname of the file from which the standard input stream of the job is taken. This variable might be used in combination with <b>GE_O_HOST</b> in prolog/epilog scripts to transfer the input file from the submit to the execution host.
<b>GE_JOB_SPOOL_DIR</b>	The directory used by <i>ge_shepherd</i> (8) to store job related data during job execution. This directory is owned by root or by a Grid Engine administrative account and commonly is not open for read or write access to regular users.
<b>GE_TASK_ID</b>	The index number of the current array job task (see <b>-t</b> option above). This is an unique number in each array job and can be used to reference different input data records, for example. This environment variable is set to "undefined" for non-array jobs. It is possible to change the predefined value of this variable with <b>-v</b> or <b>-V</b> (see options above).
<b>GE_TASK_FIRST</b>	The index number of the first array job task (see <b>-t</b> option above). It is possible to change the predefined value of this variable with <b>-v</b> or <b>-V</b> (see options above).
<b>GE_TASK_LAST</b>	The index number of the last array job task (see <b>-t</b> option above). It is possible to change the predefined value of this variable with <b>-v</b> or <b>-V</b> (see options above).
<b>GE_TASK_STEPSIZE</b>	The step size of the array job specification (see <b>-t</b> option above). It is possible to change the predefined value of this variable with <b>-v</b> or <b>-V</b> (see options above).
<b>ENVIRONMENT</b>	The <b>ENVIRONMENT</b> variable is set to <b>BATCH</b> to identify that the job is being executed under Grid Engine control.
<b>HOME</b>	The user's home directory path from the <i>passwd</i> (5) file.

HOSTNAME	The hostname of the node on which the job is running.
JOB_ID	A unique identifier assigned by the <i>ge_qmaster</i> (8) when the job was submitted. The job ID is a decimal integer in the range 1 to 99999.
JOB_NAME	The job name. For batch jobs or jobs submitted by <i>qrsh</i> with a command, the job name is built as basename of the <i>qsub</i> script filename resp. the <i>qrsh</i> command. For interactive jobs it is set to 'INTERACTIVE' for <i>qsh</i> jobs, 'QLOGIN' for <i>qlogin</i> jobs and 'QRLOGIN' for <i>qrsh</i> jobs without a command.  This default may be overwritten by the <i>-N</i> option.
JOB_SCRIPT	The path to the job script which is executed. The value can not be overwritten by the <i>-v</i> or <i>-V</i> option.
LOGNAME	The user's login name from the <i>passwd</i> (5) file.
NHOSTS	The number of hosts in use by a parallel job.
NQUEUES	The number of queues allocated for the job (always 1 for serial jobs).
NSLOTS	The number of queue slots in use by a parallel job.
PATH	A default shell search path of: /usr/local/bin:/usr/ucb/bin:/usr/bin
SGE_BINARY_PATH	The path where the Grid Engine binaries are installed. The value is the concatenation of the cluster configuration value <b>binary_path</b> and the architecture name <b>\$SGE_ARCH</b> environment variable.
PE	The parallel environment under which the job executes (for parallel jobs only).
PE_HOSTFILE	The path of a file containing the definition of the virtual parallel machine assigned to a parallel job by Grid Engine. See the description of the <b>\$pe_hostfile</b> parameter in <i>ge_pe</i> (5) for details on the format of this file. The environment variable is only available for parallel jobs.
QUEUE	The name of the cluster queue in which the job is running.
REQUEST	Available for batch jobs only.  The request name of a job as specified with the <i>-N</i> switch (see above) or taken as the name of the job script file.
RESTARTED	This variable is set to 1 if a job was restarted either after a system crash or after a migration in case of a checkpointing job. The variable has the value 0 otherwise.
SHELL	The user's login shell from the <i>passwd</i> (5) file. <b>Note:</b> This is not necessarily the shell in use for the job.
TMPDIR	The absolute path to the job's temporary working directory.
TMP	The same as TMPDIR; provided for compatibility with NQS.
TZ	The time zone variable imported from <i>ge_execd</i> (8) if set.
USER	The user's login name from the <i>passwd</i> (5) file.
SGE_JSV_TIMEOUT	If the response time of the client JSV is greater than this timeout value, then the JSV will attempt to be re-started. The default value is 10 seconds, and this value must be greater than 0. If the timeout has been reached, the JSV will only try to re-start once, if the timeout is reached again an error will occur.

## RESTRICTIONS

There is no controlling terminal for batch jobs under Grid Engine, and any tests or actions on a controlling terminal will fail. If these operations are in your .login or .cshrc file, they may cause your job to abort.



Insert the following test before any commands that are not pertinent to batch jobs in your .login:

```
if ( $?JOB_NAME) then
    echo "Grid Engine spooled job"
    exit 0
endif
```

Don't forget to set your shell's search path in your shell start-up before this code.

## EXIT STATUS

The following exit values are returned:

- 0        Operation was executed successfully.
- 25       It was not possible to register a new job according to the configured *max\_u\_jobs* or *max\_jobs* limit. Additional information may be found in *sge\_conf*(5)
- >0       Error occurred.

## EXAMPLES

The following is the simplest form of a Grid Engine script file.

```
=====

#!/bin/csh
a.out
```

The next example is a more complex Grid Engine script.

```
=====

#!/bin/csh

# Which account to be charged cpu time
#$ -A santa_claus

# date-time to run, format [[CC]yy]MMDDhhmm[.SS]
#$ -a 12241200

# to run I want 6 or more parallel processes
# under the PE pvm. the processes require
# 128M of memory
#$ -pe pvm 6- -l mem=128

# If I run on dec_x put stderr in /tmp/foo, if I
# run on sun_y, put stderr in /usr/me/foo
#$ -e dec_x:/tmp/foo,sun_y:/usr/me/foo

# Send mail to these users
#$ -M santa@nothpole,claus@northpole

# Mail at beginning/end/on suspension
#$ -m bes

# Export these environmental variables
```

```
#$ -v PVM_ROOT,FOOBAR=BAR
```

```
# The job is located in the current
# working directory.
```

```
#$ -cwd
```

```
a.out
```

```
=====
```

## FILES

```
$REQUEST.oJID[.TASKID]   STDOUT of job #JID
$REQUEST.eJID[.TASKID]   STDERR of job
$REQUEST.poJID[.TASKID]  STDOUT of par. env. of job
$REQUEST.peJID[.TASKID]  STDERR of par. env. of job
```

```
$cwd/.ge_aliases        cwd path aliases
$cwd/.ge_request        cwd default request
$HOME/.ge_aliases       user path aliases
$HOME/.ge_request       user default request
<ge_root>/<cell>/common/ge_aliases
                           cluster path aliases
<ge_root>/<cell>/common/ge_request
                           cluster default request
<ge_root>/<cell>/common/act_qmaster
                           Grid Engine master host file
```

## SEE ALSO

*ge\_intro*(1), *qconf*(1), *qdel*(1), *qhold*(1), *qmod*(1), *qrls*(1), *qstat*(1), *accounting*(5), *ge\_aliases*(5), *ge\_conf*(5), *ge\_request*(5), *ge\_pe*(5), *complex*(5).

## COPYRIGHT

If configured correspondingly, *qrsh* and *qlogin* contain portions of the *rsh*, *rshd*, *telnet* and *telnetd* code copyrighted by The Regents of the University of California. Therefore, the following note applies with respect to *qrsh* and *qlogin*: This product includes software developed by the University of California, Berkeley and its contributors.

See *ge\_intro*(1) as well as the information provided in <ge\_root>/3rd\_party/qrsh and <ge\_root>/3rd\_party/qlogin for a statement of further rights and permissions.