DQS User Interface
Preliminary Design Document

Supercomputer Computations Research Institute
Florida State University
Tallahassee, Florida

Thomas P. Green

DQS is an experimental batch queuing system being developed at the *Supercomputer Computations Research Institute* at *Florida State University* funded by the United States Department of Energy through Contract DE-FC0585ER250000. This paper will provide an overview of the basic design of DQS with respect to the DQS *user*, *operator* and *manager* interface. Where feasible, DQS will adopt the symantics of the POSIX P1003.15/D12 draft with regards to the user/operator/manager interface. Some deviations from the standards will be necessary due to the unique features of DQS.

As with its predecessors, *DNQS* and *DQS Version2.X*[1], DQS will also provide an optional **X Window** graphical interface for all commands referenced in this paper. The discussion of the **X Window** interface is beyond the scope of this paper.

[1] DNQS and DQS Version 2.X were developed at Supercomputer Computations Research Institute and are available via anonymous ftp from ftp.scri.fsu.edu in the files pub/DNQS/DNQS.tar.Z and pub/DQS/DQS.tar.Z, respectively.

Contents

I Reference Manual

Part I

Reference Manual

Chapter 1

Commands

1.1 qacct - Accounting for DQS

1.1.1 Synopsis

qacct [**-help**] [**-l** *resource_list*] [**-long**] [**-q** *destination_identifier_list*] [**-u** *user_list*] [**-verify**]

1.1.2 Description

The `qacct` utility reports the *DQS* usage statistics.

A request for the breakdown of information can be made by using any combination of the available options.

All units of time are reported in seconds.

1.1.3 Options

The command options are as follows:

-help

Prints a listing of all options.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],...]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-long

Provide a complete listing for all finished jobs.

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-u *user_list*

Specifies the user name or access list.

The *user_list* shall conform to the following syntax:

user_name[@host][,user_name[@host],...]

The utility shall allow only one user name for a given host.

The utility shall allow only one user name that is missing a corresponding host name.

-verify

Causes DQS to print request to stdout. No action taken.

1.1.4 Operands

1.1.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.1.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.1.7 Extended Description

None.

1.1.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.1.9 Consequences of Errors

1.1.10 Unresolved Items

1.2 qalter - Alter Batch Job Attributes

1.2.1 Synopsis

qalter [**-a** *date_time*] [**-A** *account_list*] [**-cell** *cell_name*] [**-clean**] [**-cwd**] [**-C**] [**-e** *path_list*] [**-h**] [**-hard**] [**-help**] [**-j** *y* ‖ *n*] [**-jid** *job_identifier*] [**-l** *resource_list*] [**-m** *mail_options*] [**-master** *destination_identifier, ...*] [**-M** *mail_list*] [**-notify**] [**-N** *name*] [**-o** *path_list*] [**-p** *priority*] [**-passwd**] [**-passwd_file** *filename*] [**-q** *destination_identifier_list*] [**-r** *y*‖*n*] [**-reauth** *seconds*] [**-soft**] [**-S** *path_list*] [**-v** *variable_list*] [**-verify**] [**-V**] [**-z**] [*script*]

1.2.2 Description

The `qalter` utility is a user-accessible batch client that allows for the alteration of specific attributes of specified jobs by sending a Modify Job Request to the batch server that manages each job.

By default the `qalter` only modifies the specified attributes. if the -clean option is specified all attributes are cleared.

The `qalter` utility shall place the value of the following environment variables in the variable-list attribute of the job:

HOME , **LOGNAME** , **PATH** , **MAIL** , **SHELL** , and **TZ**.

The name of the environment variable shall be the current name prefixed with the string {DQS_O_}.

In addition to the variables described above, the The `qalter` utility shall add the following variables with the indicated values to the variable list:

— DQS_O_WORKDIR — the absolute path of the current working directory of the `qalter` utility process.
— DQS_O_HOST — the name of the host on which the `qalter` utility is running.

Following is a summary of information related to the environment variables:

Variable Name	Value at qalter time
DQS_O_HOME	$HOME
DQS_O_LOGNAME	$LOGNAME
DQS_O_PATH	$PATH
DQS_O_MAIL	$MAIL
DQS_O_SHELL	$SHELL
DQS_O_TZ	$TZ
DQS_O_WORKDIR	current working directory
DQS_O_HOST	client host name

1.2.3 Options

The command options are as follows:

-a *date_time*

The time at which a job is eligible for execution.

The *date_time* option-argument conforms to the following syntax:

[[CC]YY]MMDDhhmm where:

`CC` Specifies the first two digits of the year.

`YY` Specifies the last two digits of the year.

`MM` Specifies the month of the year (01 to 12).

`DD` Specifies the day of the month (01 to 31).

`hh` Specifies the hour of the day (00 to 23).

`mm` Specifies the minute of the hour (00 to 59).

The execute-time attribute of the job is defined as the number of seconds since Epoch that is equivalent to the local time expressed by the value of the *date_time* argument.

If the `-a` option is not presented to the `qalter` utility, the utility shall set the execute-time attribute of the job to a time that is earlier than the time at which the utility exits.

-A *account_list*

The account under which a job is processed.

The syntax of the *account_list* option-argument is as follows:

account_name[@cell][,account_name[@cell],...]

The `qalter` utility shall set the account-name attribute of the job to the value of the *account_list* option-argument.

If the **-A** option is not presented to the `qalter` utility, the utility shall omit the account-name attribute from the attributes of the job.

-cell *cell_name*

The cell to which the request is forwarded.

-clean

Specifies to start with a clean slate with respect to resoucres, options, etc.

-cwd

Specifies to use the current working directory at job exec time.

-C

Defines the prefix that declares a directive to the utility.

The directive prefix is not a job attribute, but affects the behavior ob the utility.

If the **-C** option is presented with the value of the *directive_prefix* as a null string, the utility will not scan the scriptfile for directives.

-e *path_list*

The path used for standard error stream of the job with the following form:

> [hostname1:]/usr[,hostname2:]/tmp,...]
>
> For example:
>
> -e *hostname:/user/users/green/aix*
> -e *hostname:aix*

A relative path will be relative to the *home* directory for the user on said machine.

If the *path_list* option-argument constitutes an absolute path name, the error-path attribute of the job shall be set to the value of the *path_list* option-argument, including the hostname, if present.

If the *path_list* option-argument constitutes a relative path name, and no hostname is specified, the error-path attribute of the job shall be set to the value of the absolute path name derived by expanding the *path_list* option-argument relative to the current directory of the process that executes the utility.

If the *path_list* option-argument constitutes a relative path name, and a hostname is specified, the error-path attribute of the job shall be set to the value of the *path_list* option- argument without expansion. The hostname shall be included.

If the *path_list* option-argument does not include a hostname, the path name in the error-path attribute of the job shall be given the prefix, *hostname:*, where *hostname* is the name of the host upon which the utility is being executed.

The default file name for standard error has the following form:

*job_name.***e***sequence_number*

If the **-e** option is not presented to the qalter utility, the utility shall set the error-path attribute of the job to the hostname and path of the current directory of the submitting process and the default file name.

-h

Specifies that a "USER" hold be placed on the job by setting the value of the hold-types attribute of the job to the value {NO_HOLD}.

If the **-h** option is not presented to the qalter utility, the utility shall set the hold-types attribute of the job to the value {NO_HOLD}.

-hard

Specifies that subsequent resource requests must be satisfied in full before a job is to be scheduled.

-help

Prints a listing of all options.

-j *y* || *n*

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

If the value of the option-argument is 'y' the join-paths of the job whall be set to {TRUE}.

If the value of the option-argument is 'n' the join-paths of the job whall be set to {FALSE}.

If both the **-j y** and the **-e** options are specified, the batch server shall set but ignore the error-path attribute.

If the **-j** option is not presented to the qalter utility, the utility shall set the value of the join-paths attribute to {FALSE}.

-jid *job_identifier*

Defines the job identifier to act upon.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],...]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

If the **l** option is not presented to the `qalter` utility, the utility shall omit the resource-list attribute from the job attributes.

-m *mail_options*

Defines the points during execution of the job at which the server is to send mail as to the state of the job.

The *mail_list* option argument shall conform to the following:

— 'a' — {MAIL_AT_ABORT}
— 'b' — {MAIL_AT_BEGINNING}
— 'e' — {MAIL_AT_EXIT}
— 'n' — {NO_MAIL}
— 's' — {MAIL_AT_SUSPENSION}

Repetition of any of the characters within the *mail_options* option-argument shall be permitted, but the utility will not assign additional meaning to the additional characters.

If the **-m** option is not presented to the `qalter` utility, the utility shall set the mail-points attribute to the value {MAIL_AT_ABORT}.

-master *destination_identifier, ...*

Defines the queue to serve as the master for said job.

The *destination_identifier* option argument shall conform to the following:

queue[-exec.eq.exec_str][,queue[-exec.eq.exec_str]...]

DQS will attempt to assign the *master* based on a first to last transvesal of the *destination_identifier_list*. If the *destination_identifier_list* is specified as a *soft* request and DQS is unable to fulfill the master request based on the respective *destination_identifier_list*, DQS will assign a master as available.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-M *mail_list*

Specifies the list of users to which mail about the job will be sent by the server, if the server sends such mail.

If the implementation of the utility uses a name service to locate users, the utility shall require and accept the syntax used by the name service.

If the implementation of the utility does not use a name service, the following syntax for user names will be required and accepted:

mail_address [,mail_address,. . .]
where *mail_address* shall conform to:
user[@hostname]

The utility shall set the mail-users attribute of the job to the value of the *mail_list* option-argument.

If the **-M** option is not presented to the qalter utility, the utility shall place only the user name and host name for the current process in the mail-users attribute of the job.

-notify

Specifies that the job should be signaled via a *SIGUSR1* on an impending *SIGSTOP* and a *SIGUSR2* on an impending *SIGKILL*.

-N *name*

The name of a specific job.

The *name* option-argument shall be any printable ascii string up to 8 characters. Names in excess of 8 characters will be truncated.

The `qalter` utility shall set the value of the job-name attribute of the job to the value of the *name* option-argument.

If the **-N** option is not presented to the `qalter` utility, the utility shall set the job-name attribute of the job to the name of the script argument from which the directory specification, if any, has been removed.

If the **-N** option is not presented to the `qalter` utility, and the script is read from standard input, the utility shall set the job-name attribute of the job to the value {STDIN}.

-o *path_list*

The pathname for standard output.

For example:

-e *hostname:/user/users/green/aix*
-e *hostname:aix*

A relative path will be relative to the *home* directory for the user on said machine.

If the *path_list* option-argument constitutes an absolute path name, the stdout-path attribute of the job shall be set to the value of the *path_list* option-argument, including the hostname, if present.

If the *path_list* option-argument constitutes a relative path name, and no hostname is specified, the output-path attribute of the job shall be set to the value of the absolute path name derived by expanding the *path_list* option-argument relative to the current directory of the process that executes the utility.

If the *path_list* option-argument constitutes a relative path name, and a hostname is specified, the output-path attribute of the job shall be set to the value of the *path_list* option- argument without expansion. The hostname shall be included.

If the *path_list* option-argument does not include a hostname, the path name in the output-path attribute of the job shall be given the prefix, *hostname:*, where *hostname* is the name of the host upon which the utility is being executed.

The default file name for standard output has the following form:

*job_name.*o*sequence_number*

If the **-o** option is not presented to the `qalter` utility, the utility shall set the output-path attribute of the job to the hostname and path

-p *priority*

Specifies the priority of the job relative to other jobs.

The priority attribute of the job shall be set to the value of the *priority* option-argument.

The utility shall require and accept a value for the *priority* option-argument that conforms to the syntax for signed decimal integers, and which is not less than -1024 and not greater than 1023.

If the **-p** option is not presented to the `qalter` utility, the value of the priority attribute is implementation defined.

-passwd

Specifies that a password list will be requested for AFS and/or DFS reauthentication. If given, DQS shall utilize the password list to authenticate prior to executing the job and reauthenticate as necessary.

The utility shall either prompt for password(s) to be entered interactively until CTL-D is entered.

-passwd_file *filename*

The name of the file in which the password(s) reside.

Entries in the file will be in the format:

password cell principal passwd
password cell principal passwd
...

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

The `qalter` utility shall submit the script to the batch server named by the *destination* option-argument, or the server that owns the queue named in the *destination* option-argument.

If the option-argument describes only a queue the `qalter` utility shall queue the job at the queue of the specified name at the default batch server.

If the option-argument describes only a batch server, the `qalter` utility shall submit the script to that designated batch server.

If the option-argument describes both a queue and a batch server, the `qalter` utility shall queue the job in the specified queue at the specified server.

-r $y \| n$

Identifies the ability of a job to be rerun or not.

An option-argument value of '**y**' corresponds to a value of {TRUE} for the rerunable attribute of the job.

An option-argument value of '**n**' corresponds to a value of {FALSE} for the rerunable attribute of the job.

If the **-r** option is not presented to the `qalter` utility, the utility shall set the rerunable attribute of the job to {FALSE}.

-soft

Specifies that subsequent resource and destination identifiers are to be filled on an "as available" basis.

-reauth *seconds*

Specifies the interval in which to reauthenticate.

-S *path_list*

Specifies the interpreting shell.

The *path_list* shall conform to the following syntax:

pathname[@host][,pathname[@host],...]

The utility shall allow only one path name for a given host.

The utility shall allow only one path name that is missing a corresponding host name.

If the **-S** option is not presented to the `qalter` utility, the utility shall set the shell-path-list attribute of the job to the null string.

-v *variable_list*

Defines the environmental variables to be exported.

A *variable_list* is a set of strings delimited by commas of either the form $< variable >$ or $< variable = value >$.

A variable shall not be repeated in the environment *variable_list* option-argument.

If the `-v` option is presented to teh `qalter` utility, the utility shall also add to the environment variable-list attribute of the job every variable named in the *variable_list* option-argument and, optionally, values of specified variables.

-verify

Causes DQS to print request to stdout. No action taken.

-V

Specifies that all environment variables be exported to the context of the job.

The `qalter` utility shall place in the list every environment variable in the process in which the utility is executing. It shall also set the value of each variable in the attribute to the value of that variable in the process.

-z

Specifies that utility shall not write the job identifier to the standard output.

If the `-z` option is presented to the utility, the utility shall not write the job identifier of the created job to standard output.

If the `-z` option is not presented to the utility, the utility shall write the job identifier of the created job to standard output.

1.2.4 Operands

script

The path to the script of the job.

If the *script* operand is not presented to the utility, or if the operand is the single-character string '-', the utility shall read the script from standard input.

If the *script* represents a partial path, the utility shall expand the path relative to the current directory of the process executing the utility.

1.2.5 External Influences

Standard Input

The `qalter` utility reads the script of the job from standard input if the *script* operand is omitted or is the single character '-'.

Input Files

In addition to binding the file indicated by the *script* operand to the job, the `qalter` utility reads the *script* file and acts on directives in the script.

Environmental Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

The values of some or all of the environment variables of the process that executes the `qalter` utility are placed in the attributes of the job.

The environment variable DQS_DPREFIX determines the default prefix for directives within the script.

Asynchronous Events

Once created, a job exists until it exits, aborts, or is deleted.

After a job is created by the `qalter` utility, batch servers might route, execute, modify, or delete the job.

1.2.6 External Effects

Standard Output

The `qalter` utility writes the job identifier assigned to the job to standard output.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.2.7 Extended Description

Script Preservation

The `qalter` utility shall make the script available to the server executing the job in such a way that the server executes the script as it exists at the time of submission.

The `qalter` utility can send the script available to the server with the Queue Job Request or store a temporary copy of the script in a location specified to the server.

Option Specification

A script can contain directives to the `qalter` utility.

The `qalter` utility shall scan the lines of the script for directives until the first line that begins with a string other than the directive string; if directives occur on subsequent lines, the util shall ignore those directives.

The `qalter` utility shall process a line in the script as a directive if, and only if, the string of characters from the first non-white space character on the line until the first space or tab character on the line match the directive prefix.

The `qalter` utility shall ignore any blank lines and shell comment lines prior to the first executable line.

The `qalter` utility shall process the options and option-arguments contained on the directive prefix line using the same syntax as if the options were input on the `qalter` utility.

The `qalter` utility shall continue to process a directive prefix line until after a $< newline >$ or an unquoted comment character of the shell that will interpret the script is encountered.

If an option is present in both a directive and arguments to the `qalter` utility, the utility shall ignore the option and the corresponding option-argument, if any, in the directive.

If an option that is present in the directive is not present in the arguments to the `qalter` utility, the utility shall process the option and the option-argument, if any.

In order or preference, the `qalter` utility shall select the directive prefix from one of the following sources:

— if the **-C** option is presented to the utility, the value of the *directive_prefix* option-argument.

— if the environment variable POSIX2_DPREFIX is defined, the value of that variable.

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

1.2.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero. The utility shall have failed if any of the following conditions apply:

— if the argument list presented to the `qalter` utility does not conform to the synopsis.

— if any option-argument does not conform to the syntax described for it.

1.2.9 Consequences of Errors

1.2.10 Unresolved Items

The following options are yet to be implemented:

-A account_list
-B
-c interval
-h
-k
-m mail_options
-M mail_list -p
-u user_list
-z
-

The following environmental variables are yet to be incorporated:

DQS_DPREFIX
POSIX_DPREFIX

1.3 qconf - DQS Queue Configuration

1.3.1 Synopsis

qconf [**-ac** *complex_name*] [**-ah** *server_list*] [**-am** *user_list*] [**-ao** *user_list*] [**-aq** [*queue_template*]] [**-au** *user_list list_name_list*] [**-Aq** *fname*] [**-cell** *cell_name*] [**-cq** *destination_id_list*] [**-dh** *server_list*] [**-dm** *user_list*] [**-do** *user_list*] [**-dq** *destination_id_list*] [**-du** *user_list list_name_list*] [**-help**] [**-sc** *complex_list*] [**-scl**] [**-sconf** [*cell*]] [**-sh**] [**-sm**] [**-so**] [**-sq** *[destination_id_list]*] [**-sql**] [**-su** *list_name*] [**-sul**] [**-srl**]

1.3.2 Description

The `qconf` utility provides the user interface for adding, deleting and modifying a DQS queue.

The `qconf` utility also provides the means for displaying the current queue configuration.

The `qconf` may be utilized by either a manager, operator or user with increasing restrictions.

1.3.3 Options

The command options are as follows:

qconf

-ac *complex_name*

"Add complex."

Adds a complex to the DQS complex hash tables.

The *complex* entries shall conform to one of the following:

string
string=string
string=integer_quantity

Requires root/manager privileges.

-ah *server_list*

"Add host."

Adds a host to the DQS host list. A host must reside in the DQS host list to execute any of the DQS commands. The sole exception to this rule is the excution of `qconf` on the `qmaster` node.

The *hostname_list* shall conform to the following:

hostname[,hostname,...]

Requires root/manager privileges.

-am *user_list*

"Add a manager."

Adds a manager to the manager list.

The *user_list* option argument shall conform to the following:

user[,user,...]

Requires root/manager privileges.

-ao *user_list*

"Add an operator."

Adds an operator to the operator list.

The *user_list* option argument shall conform to the following:

user[,user,...]

Requires root/manager privileges.

-aq [*queue_template*]

"Add new queue."

Retrieves the default queue configuration, or the queue_template if specified, and executes vi (or $EDITOR if EDITOR is set)to allow the customization of the queue configuration.

Upon exit from the editor the queue is registered with the qmaster.

A minimal configuration requires only that the Qname and Qhostname be set.

Requires root/manager privileges.

-au *user_list list_name_list*

"Add user."

Adds one or more users to DQS ACLs. The user list is used for queue usage authentication.

The *user_list* option argument shall conform to the following:

user[,user,...]

The *list_name_list* option argument shall conform to the following:

list_name[,list_name,...]

Requires root/manager privileges.

-Aq *fname*

"Add new queue."

Allows the queue described in *fname* to be added.

Requires root/manager privileges.

-cell *cell_name*

The cell to which the request is forwarded.

-cq *destination_id_list*

"Clean queue."

Cleans queue from jobs which haven't been reaped.

The *destin_id_list* option argument shall conform to the following:

destin_id[,destin_id,...]

Primarily a development tool.

Requires root/manager privileges.

-dh *server_list*

"Delete host."

Deletes a host from the DQS host list.

The *server_list* option argument shall conform to the following:

host_name[,host_name,...]

Requires root/manager privileges.

-dm *user_list*

"Delete a manager."

Deletes a manager from the manager list.

The *user_list* option argument shall conform to the following:

user[,user,...]

Requires root/manager privileges.

-do *user_list*

"Delete an operator."

Deletes an operator from the operator list.

The *user_list* option argument shall conform to the following:

user[,user,...]

Requires root/manager privileges.

-dq *destination_id_list*

"Delete queue."

Removes the specified queue(s).

Active jobs will be allowed to run to completion.

The *destination_id_list* option argument shall conform to the following:

destination_id[,destination_id,...]

Requires root/manager privileges.

-du *user_list list_name_list*

"Delete user."

Deletes one or more users from one or more DQS ACLs.

The *user_list* option argument shall conform to the following:

user[,user,...]

The *list_name_list* option argument shall conform to the following:

list_name[,list_name,...]

Requires root/manager/operator privileges.

-help

Prints a listing of all options.

-sc *complex_list*

"Show complexes."

The *complex_list* option argument shall conform to the following:

complex_name[,complex_name,...]

-scl

"Show complex list names."

-sconf [*cell*]

"Show master configuration for cell."

-sh

"Show hosts."

Displays the DQS host list.

-sm

"Show managers."

Displays the manager list.

-so

"Show operators."

Displays the operator list.

-sq *[destination_id_list]*

"Show queue configurations."

-sql

"Show list of destination_id's."

-su *list_name*

"Show users."

Displays a DQS user list(s).

The *user_list* option argument shall conform to the following:

user[,user,...]

-sul

"Show user lists."

Displays all DQS user lists.

-srl

"Show resolve list."

Displays the DQS resolver list.

1.3.4 Operands

1.3.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.3.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.3.7 Extended Description

None.

1.3.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.3.9 Consequences of Errors

1.3.10 Unresolved Items

The following options are yet to be implemented:

-dc complex_name
-debug_on
-debug_off
-qmon
-kq server_list
-kqs server_list
-log_level
-mc complex_name
-mconf
-mq queue_name

The following options need to be cleaned up:

-sul

1.4 qdel - Delete Batch Jobs

1.4.1 Synopsis

qdel [**-help**] [**-verify**] [*job_identifier_list*]

1.4.2 Description

The `qdel` utility provides a means for a user to delete one or more jobs. A job that has been deleted is no longer subject by batch services.

The `qdel` utility shall delete jobs in the order in which their job identifiers are presented. If the `qdel` utility fails to process any identified job successfully, the utility shall proceed to process the remaining jobs in the *job_identifier* list, if any.

The `qdel` utility shall delete each job by sending a Delete Job Request to the batch server that manages the job.

The `qdel` utility shall not exit until the job corresponding to each successfully processed *job_identifier* has been deleted.

1.4.3 Options

The command options are as follows:

-help

Prints a listing of all options.

-verify

Causes DQS to print request to stdout. No action taken.

1.4.4 Operands

job_identifier_list

Specifies the interpreting shell, in the form:

sequence_number[@cell][,sequence_number[@cell],...]

If the application omits the *cell* portion of a job identifier, the utility shall use either:

— the name of the cell specified on the command line, if any.

— the name of the cell specified in the environment variable DQS_CELL, if it is set

— the name of the default cell

1.4.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None

1.4.6 External Effects

Standard Output

An implementation of the utility may write informative messages to Standard Output.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.4.7 Extended Description

None.

1.4.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.4.9 Consequences of Errors

1.4.10 Unresolved Items

1.5 qhold - Hold Batch Jobs

1.5.1 Synopsis

qhold [**-h** *hold_list*] [**-help**] [**-verify**] [*job_identifier_list*]

1.5.2 Description

The `qhold` utility provides a means for a user to place one or more types of hold on identified job(s). A job that has one or more holds is not eligible for execution.

The `qhold` utility shall place holds on jobs in the order in which their identifiers are presented to the utility.

If `qhold` fails to process any identified job successfully, the utility shall proceed to process the remaining job identifiers, if any.

The `qhold` utility shall place holds on each job by sending a Hold Job Request to the batch server that manages the job.

The `qhold` utility shall not exit until holds have been placed on the the job corresponding to each successfully processed job identifier.

1.5.3 Options

The command options are as follows:

-h *hold_list*

Defines the types of holds. The *hold_list* shall conform to to one or more of the following:

— 'u' — {USER}
— 's' — {SYSTEM}
— 'o' — {OTHER}
— 'n' — {NO_HOLD}

A conforming application shall not repeat any character within the *hold_list* option-argument. The **qhold** utility shall accept repeated characters within the *hold_list* option-argument, but shall not assign additional meaning to repeated characters.

For each character in the value of the *hold_list* option-argument, the bf qhold utility shall add the corresponding hold to the value of the hold_types job attribute.

If the **-h** option is not presented to the `qhold` utility, the implementation shall set the hold_types attribute to {USER}, the default hold type.

-help

Prints a listing of all options.

-verify

Causes DQS to print request to stdout. No action taken.

1.5.4 Operands

job_identifier_list

Specifies the interpreting shell, in the form:

sequence_number[@cell][,sequence_number[@cell],...]

If the application omits the *cell* portion of a job identifier, the utility shall use either:

— the name of the cell specified on the command line, if any.
— the name of the cell specified in the environment variable DQS_CELL, if it is set
— the name of the default cell

If the application presents the **server** portion of a job identifier to the `qhold` utility, the utility shall send the Hold Job Request to the specified server.

1.5.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.5.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.5.7 Extended Description

1.5.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.5.9 Consequences of Errors

1.5.10 Unresolved Items

1.6 qidle - Monitor X server for DQS

1.6.1 Synopsis

qidle [**-d** *display*] [**-g** *geometry*] [**-help**] [**-q** *destination_identifier_list*] [**-s** *seconds*] [**-verify**]

1.6.2 Description

The `qidle` utility monitors the system's console for keyboard and mouse activity to the X server, indicating that someone is using the console interactively. When this activity is detected, `qidle` automatically suspends the local DQS queue.

If the X server is idle for a certain period of time, then `qidle` resumes the local queue.

The sleep button in the `qidle` window displayed on the screen allows you to modify the length of time that the queue can be suspended.

1.6.3 Options

The command options are as follows:

-d *display*

Display name. See X(1) for more information.

-g *geometry*

Geometry specification for the window. See X(1).

-help

Prints a listing of all options.

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-s *seconds*

Length of time of inactivity befor re-enabling the queue(s).

Default length is 300 seconds.

-verify

Causes DQS to print request to stdout. No action taken.

1.6.4 Operands

1.6.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.6.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.6.7 Extended Description

None.

1.6.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.6.9 Consequences of Errors

1.6.10 Unresolved Items

1.7 qmod - Modify the State of a DQS queue

1.7.1 Synopsis

qmod [**-cell** *cell_name*] [**-s**]/ [**-us**] [**-f**] [**-help**] [**-verify**] [**-q** *destination_identifier_list*]
qmod [**-cell** *cell_name*] [**-e**]/ [**-d**] [**-f**] [**-help**] [**-verify**] [**-q** *destination_identifier_list*]
qmod [**-cell** *cell_name*] [**-s**]/ [**-us**] [**-f**] [**-help**] [**-verify**] [**-l** *resource_list*]
qmod [**-cell** *cell_name*] [**-e**]/ [**-d**] [**-f**] [**-help**] [**-verify**] [**-l** *resource_list*]

1.7.2 Description

The `qmod` command enables users classified as owners of a workstation to modify the state of a DQS queue for his/her machine. A manager, operator, or the superuser can execute `qmod` for any queue in a cluster.

1.7.3 Options

The command options are as follows:

-cell *cell_name*

The cell to which the request is forwarded.

-d

Disables the queue.

-e

Enables the queue.

-f

Forces action.

If *execd* can be contacted, this only affects queue status, not the job status.

-help

Prints a listing of all options.

-s

Suspends the queue and any jobs which might be active.

-us

Unsuspends the queue and any jobs which might be active.

-verify

Causes DQS to print request to stdout. No action taken.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],...]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

1.7.4 Operands

1.7.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.7.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.7.7 Extended Description

None.

1.7.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.7.9 Consequences of Errors

1.7.10 Unresolved Items

The following options are yet to be implemented:

-soc
-xsoc

1.8 qmon - X-Window interface for dqs

1.8.1 Synopsis

qmon [**-d** *display*] [**-fn** *font*] [**-rv**] [**-l** *n*] [**-help**] [**-verify**]

1.8.2 Description

The `qmon` command enables users to manipulate the dqs queues from an X-Window environment. The `qmon` environment consists of four main components:

— the main menu, which allows a user to selct one of the dqs utility programs;

— the submenus, which allow a user to enter the parameters for a selected main menu option and then execute it;

— the icon window, which displays which queues exist, and displays the status of those queues whose icons are selected by a click;

— the output window, to which all stdout and/or stderr messages from commands executed within qmon are redirected.

1.8.3 Options

The command options are as follows:

-d *display*

Display name. See X(1) for more information.

-fn *font*

Name of type font of display.

-rv

Display shown in reverse video.

-l *n*

Specifies the number of lines *n* used in the output window.

-help

Prints a listing of all options.

-verify

Causes DQS to print request to stdout. No action taken.

1.8.4 Operands

1.8.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.8.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.8.7 Extended Description

None.

1.8.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.8.9 Consequences of Errors

1.8.10 Unresolved Items

1.9 qmove - Move Batch Jobs

1.9.1 Synopsis

qmove [**-help**] [**-u** *user_list*] [**-verify**] [*destination_identifier*] [*job_identifier_list*]

1.9.2 Description

The `qmove` utility provides a means for a user to remove a job (or jobs) from the queue in which it resides and install it in another queue.

The `qmove` utility shall move those jobs, and only those jobs, for which a job identifier is presented.

The `qmove` utility shall move jobs in the order in which the corresponding job identifiers are presented to the utility.

If the `qmove` utility fails to process any job identifier successfully, the utility shall proceed to process the remaining job identifiers, if any.

The `qmove` utility shall move jobs by sending a Move Job Request to the batch server that manages each job.

The `qmove` utility shall not exit before the jobs corresponding to all successfully processed job identifiers have been moved.

1.9.3 Options

The command options are as follows:

-help

Prints a listing of all options.

-u *user_list*

Specifies the user name or access list.

The *user_list* shall conform to the following syntax:

user_name[@host][,user_name[@host],...]

The utility shall allow only one user name for a given host.

The utility shall allow only one user name that is missing a corresponding host name.

-verify

Causes DQS to print request to stdout. No action taken.

1.9.4 Operands

destination_identifier

The utility shall require the following form:

queue
@cell
queue@cell

If the *destination* operand describes only a queue, the `qmove` utility shall move jobs into the queue of the specified name at the default batch server.

If the *destination* operand describes only a cell the `qmove` utility shall move jobs into the default queue of the specified name at that batch server.

If the *destination* operand describes both a queue and a cell the `qmove` utility shall move jobs into the specifiied queue at the specified cell.

job_identifier_list

Specifies the interpreting shell, in the form:

sequence_number[@cell][,sequence_number[@cell],...]

If the application omits the *cell* portion of a job identifier, the utility shall use either:

— the name of the cell specified on the command line, if any.
— the name of the cell specified in the environment variable DQS_CELL, if it is set
— the name of the default cell

If the application presents the **cell** portion of a job identifier to the `qmove` utility, the utility shall send the Move Job Request to the specified cell.

1.9.5 External Influences

Standard Input

None.
subsubsectionInput Files
None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.9.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.9.7 Extended Description

None.

1.9.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.9.9 Consequences of Errors

1.9.10 Unresolved Items

1.10 qmsg - Write a Message to Job Output File

1.10.1 Synopsis

qmsg [**-E**] [**-help**] [**-O**] [**-verify**] [*message_string*] [*job_identifier_list*]

1.10.2 Description

The qmsg utility enables a user to request that a server write a message string into one or more output files of a specified job(s).

The qmsg utility shall write messages into the files of jobs by sending a Job Message Request to the batch server that manages each job. The qmsg utility shall not directly write the message into the files of the job.

The qmsg utility shall send a Job Message Request for those jobs, and only those jobs, for which a job identifier is presented.

The qmsg utility hall send Job Message Requests for jobs in the order in which the corresponding job identifiers are presented to the utility.

If the qmsg utility fails to process any job identifier successfully, the utility shall proceed to process the remaining job identifiers, if any.

The qmsg utility shall not exit before a Job Message Request has been sent to the server that manages the job corresponding to each successfully processed job identifier.

1.10.3 Options

The command options are as follows:

-E

Specifies that the message should be sent to standard error.

-help

Prints a listing of all options.

-O

Specifies that a message be sent to the standard output of each job.

-verify

Causes DQS to print request to stdout. No action taken.

1.10.4 Operands

message_string

The string to be written to one or more output files of the job followed by a $< new - line >$. If the string contains blanks, then the string must be quoted.

job_identifier_list

Specifies the interpreting shell, in the form:

sequence_number[@cell][,sequence_number[@cell],...]

If the application omits the *cell* portion of a job identifier, the utility shall use either:

— the name of the cell specified on the command line, if any.

— the name of the cell specified in the environment variable DQS_CELL, if it is set

— the name of the default cell

If the application presents the **cell** portion of a job identifier to the `qmsg` utility, the utility shall send the Job Message Request to the specified cell.

1.10.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.10.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.10.7 Extended Description

None.

1.10.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.10.9 Consequences of Errors

1.10.10 Unresolved Items

1.11 qrerun - Rerun Batch Jobs

1.11.1 Synopsis

qrerun [**-help**] [**-verify**] [*job_identifier_list*]

1.11.2 Description

The `qrerun` utility provides a means for a user to rerun a job (or jobs) from the queue in which it resides and install it in another queue.

The `qrerun` utility shall rerun those jobs, and only those jobs, for which a job identifier is presented.

The `qrerun` utility shall rerun jobs in the order in which the corresponding job identifiers are presented to the utility.

If the `qrerun` utility fails to process any job identifier successfully, the utility shall proceed to process the remaining job identifiers, if any.

The `qrerun` utility shall rerun jobs by sending a Rerun Job Request to the batch server that manages each job.

The `qrerun` utility shall not exit before the jobs corresponding to all successfully processed job identifiers have been rerun.

1.11.3 Options

The command options are as follows:

-help

Prints a listing of all options.

-verify

Causes DQS to print request to stdout. No action taken.

1.11.4 Operands

job_identifier_list

Specifies the interpreting shell, in the form:

sequence_number[@cell][,sequence_number[@cell],...]

If the application omits the *cell* portion of a job identifier, the utility shall use either:

— the name of the cell specified on the command line, if any.

— the name of the cell specified in the environment variable DQS_CELL, if it is set

— the name of the default cell

If the application presents the **cell** portion of a job identifier to the qrerun utility, the utility shall send the Rerun Job Request to the specified cell.

1.11.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.11.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.11.7 Extended Description

None.

1.11.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.11.9 Consequences of Errors

1.11.10 Unresolved Items

1.12 qrls - Remove Job Holds from Batch Jobs

1.12.1 Synopsis

qrls [**-h** *hold_list*] [**-help**] [**-verify**] [*job_identifier_list*]

1.12.2 Description

The `qmove` utility provides a means for a user to release one or more jobs by removing one or more holds which prevent the job(s) from executing.

The `qrls` utility shall remove holds jobs in the order in which the corresponding job identifiers are presented to the utility.

If the `qrls` utility fails to process any job identifier successfully, the utility shall proceed to process the remaining job identifiers, if any.

The `qrls` utility shall remove holds each job by sending a Release Job Request to the batch server that manages the job.

The `qrls` utility shall not exit before the holds have been removed from the jobs corresponding to each successfully processed job identifier.

1.12.3 Options

The command options are as follows:

-h *hold_list*

Defines the types of holds. The *hold_list* shall conform to to one or more of the following:

— 'u' — {USER}
— 's' — {SYSTEM}
— 'o' — {OTHER}
— 'n' — {NO_HOLD}

A conforming application shall not repeat any character within the *hold_list* option-argument. The `qrls` utility shall accept repeated characters within the *hold_list* option-argument, but shall not assign additional meaning to repeated characters.

For each character in the value of the *hold_list* option-argument, the tt qrls utility shall remove the corresponding hold from the value of the hold_types job attribute.

If the **-h** option is not presented to the `qrls` utility, the implementation shall remove the {USER} hold from the hold_types attribute.

-help

Prints a listing of all options.

-verify

Causes DQS to print request to stdout. No action taken.

1.12.4 Operands

job_identifier_list

Specifies the interpreting shell, in the form:

sequence_number[@cell][,sequence_number[@cell],...]

If the application omits the *cell* portion of a job identifier, the utility shall use either:

— the name of the cell specified on the command line, if any.
— the name of the cell specified in the environment variable DQS_CELL, if it is set
— the name of the default cell

If the application presents the **cell** portion of a job identifier to the `qrls` utility, the utility shall send the Release Job Request to the specified cell.

1.12.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.12.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.12.7 Extended Description

None.

1.12.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.12.9 Consequences of Errors

1.12.10 Unresolved Items

1.13 qselect - Select Batch Jobs

1.13.1 Synopsis

qselect [**-a** [*op*] *date_time*] [**-A** *account_list*] [**-B**] [**-c** [*op*] *interval*] [**-h** *hold_list*] [**-help**] [**-l** *resource_list*] [**-N** *name*] [**-p** [*op*] *priority*] [**-q** *destination_identifier_list*] [**-r** *y*||*n*] [**-s** *states*] [**-u** *user_list*] [**-verify**]

1.13.2 Description

The `qselect` utility provides a means for a user to return the job identifiers for each job that meets a list of selection criteria.

Upon successful completion the `qselect` utility shall have returned a list of zero or more job identifiers that meet the criteria specified by the options and option-arguments presented to the utility.

The `qselect` utility shall select jobs by sending a Select Jobs Request to a batch server.

For each option presented to the `qselect` utility, the utility shall restrict the set of selected jobs as described in the Options section.

1.13.3 Options

The command options are as follows:

-a [*op*] *date_time*

The time at which a job is eligible for execution.

The *date_time* option-argument conforms to the following syntax:

[[CC]YY]MMDDhhmm where:

`CC` Specifies the first two digits of the year.

`YY` Specifies the last two digits of the year.

`MM` Specifies the month of the year (01 to 12).

`DD` Specifies the day of the month (01 to 31).

`hh` Specifies the hour of the day (00 to 23).

`mm` Specifies the minute of the hour (00 to 59).

If the *op* field is present the `qselect` command shall restrict selection based on the following criteria:

`.eq.` the value represented by the respective attribute must be equal to the option argument.

`.ge.` the value represented by the respective attribute must be greater than or equal to the option argument.

`.gt.` the value represented by the respective attribute must be greater than the option argument.

`.lt.` the value represented by the respective attribute must be less than the option argument.

`.le.` the value represented by the respective attribute must be less than or equal to the option argument.

`.ne.` the value represented by the respective attribute must not be equal to the option argument.

The execute-time attribute of the job is defined as the number of seconds since Epoch that is equivalent to the local time expressed by the value of the *date_time* argument.

If the *op* component of the option-argument is not presented to the `qselect` utility, jobs shall be selected for which the execute-time attribute is equal to the *date_time* component of the option-argument.

-A *account_list*

The account under which a job is processed.

The syntax of the *account_list* option-argument is as follows:

account_name[@cell][,account_name[@cell],...]

The `qselect` utility shall select only jobs for which the value of the sf account-name attribute of the job matches the value of the *account_string* option-argument.

-B

Specifies that the operand is a server name.

-c [*op*] *interval*

Defines whether a job should be checkpointed and if so, how often.

The *interval* option argument shall conform to the following:

— 'n' — no checkpointing.
— 's' — checkpointing only when server is shut down.
— 'm' — automatic periodic checkpointing shall be performed at the *minimum-interval* attribute of the queue, in CPU minutes.

— minutes — automatic periodic checkpointing shall be performed every *minutes* of CPU time or every **minimum-interval**, whichever is greater. *minutes* shall conform to the syntax for unsigned integers and shall be greater than zero.

If the *-c* option is not specified no checkpointing will be done.

If the *op* field is present the command shall restrict selection based on the *interval* attribute based on the following criteria:

.eq. the value represented by the respective attribute must be equal to the option argument.

.ge. the value represented by the respective attribute must be greater than or equal to the option argument.

.gt. the value represented by the respective attribute must be greater than the option argument.

.lt. the value represented by the respective attribute must be less than the option argument.

.le. the value represented by the respective attribute must be less than or equal to the option argument.

.ne. the value represented by the respective attribute must not be equal to the option argument.

The ordering relationship for the values of the *interval* option argument is defined to be:

'n' .lt. 's' .lt. 'm' .lt. integer .gt. 0

When comparing *checkpoint* attributes with an *interval* having the value of 'u', only equality or inequality are valid.

Requires operating system support.

-h *hold_list*

Defines the types of holds. The *hold_list* shall conform to to one or more of the following:

— 'u' — {USER}
— 's' — {SYSTEM}
— 'o' — {OTHER}
— 'n' — {NO_HOLD}

With this option the `qselect` utility restricts selection to jobs that have a specific type of hold, that is jobs for which the value of the **hold-types** attribute matches the value of the *hold_list* option-argument.

-help

Prints a listing of all options.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],. . .]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

When comparing resource values, the `qselect` utility shall use the following definitions for the *op* component of the option-argument:

.eq. – the value of the resource of the same name in the resource-list attribute of the job equals the value of the *value* component of the option-argument

.ge. – the value of the resource of the same name in the resource-list attribute of the job is greater than or equal to the value of the *value* component of the option-argument

.gt. – the value of the resource of the same name in the resource-list attribute of the job is greater than the the value of the *value* component of the option-argument

.lt. – the value of the resource of the same name in the resource-list attribute of the job is less than the the value of the *value* component of the option-argument

.le. – the value of the resource of the same name in the resource-list attribute of the job is less than or equal to the value of the *value* component of the option-argument

.ne. – the value of the resource of the same name in the resource-list attribute of the job is not equal to the the value of the *value* component of the option-argument

Only equality or inequality are valid comparisons if the limit, the *value*, or both are non-numeric.

The `qselect` utility shall select only jobs for which the values of the *resource_names* listed match the corresponding limits of the resource-list attribute of the job.

Limits of *resource_names* present in the resource-list attribute of the job that have no corresponding values in the *resource_list* shall not be considered when selecting jobs.

-N *name*

The name of a specific job.

The *name* option-argument shall be any printable ascii string up to 8 characters. Names in excess of 8 characters will be truncated.

The `qselect` utility shall select only jobs for which the value of the job-name attribute mateches the value of the *name* option-argument.

-p [*op*] *priority*

Specifies the jobs relative to the priority or to a range of priorities.

If the *op* field is present, the command shall restrict selection based on the *interval* attribute based on the following criteria:

.eq. the value represented by the priority attribute must be equal to the *priority* option-argument.

.ge. the value represented by the priority attribute must be greater than or equal to the *priority* option-argument.

.gt. the value represented by the priority attribute must be greater than the *priority* option-argument.

.lt. the value represented by the priority attribute must be less than the *priority* option-argument.

.le. the value represented by the priority attribute must be less than or equal to the *priority* option-argument.

.ne. the value represented by the priority attribute must not be equal to the *priority* option-argument.

If the *op* component of the option-argument is omitted, the utility shall select jobs for which the value of the priority attribute of the job is equal to the value of the *priority* component of the option-argument.

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

The `qselect` utility shall select only jobs that are located at the destination indicated by the value of the *destination_identifier* option-argument.

If the **-q** option is not presented to the `qselect` utility, the utility shall select jobs from all queues at the default batch server.

If the option-argument describes only a queue the `qselect` utility shall select only jobs from the queue of the specified name at the default batch server.

If the option-argument describes only a batch server, the `qselect` utility shall select jobs from all of the queues at that batch server.

If the option-argument describes both a queue and a batch server, the `qselect` utility shall select only jobs from the specified queue at the specified server.

-r *y||n*

Identifies the ability of a job to be rerun or not.

An option-argument value of '**y**' corresponds to a value of {TRUE} for the **rerunable** attribute of the job.

An option-argument value of '**n**' corresponds to a value of {FALSE} for the **rerunable** attribute of the job.

The `qselect` utility shall select only jobs for which the value of the **rerunable** attribute of the job matches the value of the option-argument.

-s *states*

Identifies specific states for selected job:

— ‘e’ — {EXITING}
— ‘h’ — {HELD}
— ‘m’ — {MIGRATING}
— ‘r’ — {RUNNING}
— ‘s’ — {SUSPENDED}
— ‘t’ — {TRANSITING}
— ‘w’ — {WAITING}

Repetition of characters will be allowed, but additional meaning shall not be assigned to repeated characters.

For each character in the *states* option-argument, the `qselect` utility whall select jobs in the corresponding state.

-u *user_list*

Specifies the user name or access list.

The *user_list* shall conform to the following syntax:

user_name[@host][,user_name[@host],...]

The utility shall allow only one user name for a given host.

The utility shall allow only one user name that is missing a corresponding host name.

-verify

Causes DQS to print request to stdout. No action taken.

1.13.4 Operands

1.13.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

None.

Asynchronous Events

None.

1.13.6 External Effects

Standard Output

The `qselect` utility shall write zero or more job identifiers to *Standard Output*, separated by white space.

The `qselect` utility shall srite job identifiers job identifiers in the following format:

sequence_nubmer.server_name@server

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.13.7 Extended Description

None.

1.13.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.13.9 Consequences of Errors

1.13.10 Unresolved Items

1.14 qsh - Interactive DQS Queue Shell

1.14.1 Synopsis

qsh [**-A** *account_list*] [**-cell** *cell_name*] [**-hard**] [**-help**] [**-l** *resource_list*] [**-master** *destination_identifier, ...*] [**-N** *name*] [**-p** *priority*] [**-q** *destination_identifier_list*] [**-soft**] [**-S** *path_list*] [**-u** *user_list*] [**-v** *variable_list*] [**-verify**] [**-V**]

1.14.2 Description

The qsh utility allows interactive access to the DQS queueing system. It supports single- and multiple-node jobs.

Only queues of the type Interactive may be used to host interactive DQS shells.

DQS flags may be entered as arguments to qsh. By default, all DQS requests are hard until the **-S** option is encountered. The value of the HARD/SOFT option remains in effect until its counterpart is encountered again.

1.14.3 Options

The command options are as follows:

-A *account_list*

The account under which a job is processed.

The syntax of the *account_list* option-argument is as follows:

account_name[@cell][,account_name[@cell],...]

-cell *cell_name*

The cell to which the request is forwarded.

-hard

Specifies that subsequent resource requests must be satisfied in full before a job is to be scheduled.

-help

Prints a listing of all options.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],...]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-master *destination_identifier, ...*

Defines the queue to serve as the master for said job.

The *destination_identifier* option argument shall conform to the following:

queue[-exec.eq.exec_str][,queue[-exec.eq.exec_str]...]

DQS will attempt to assign the *master* based on a first to last transvesal of the *destination_identifier_list*. If the *destination_identifier_list* is specified as a *soft* request and DQS is unable to fulfill the master request based on the respective *destination_identifier_list*, DQS will assign a master as available.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-N *name*

The name of a specific job.

The *name* option-argument shall be any printable ascii string up to 8 characters. Names in excess of 8 characters will be truncated.

-p *priority*

Specifies the priority of the job relative to other jobs.

The priority attribute of the job shall be set to the value of the *priority* option-argument.

The utility shall require and accept a value for the *priority* option-argument that conforms to the syntax for signed decimal integers, and which is not less than -1024 and not greater than 1023.

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-soft

Specifies that subsequent resource and destination identifiers are to be filled on an "as available" basis.

-S *path_list*

Specifies the interpreting shell.

The *path_list* shall conform to the following syntax:

pathname[@host][,pathname[@host],...]

The utility shall allow only one path name for a given host.

The utility shall allow only one path name that is missing a corresponding host name.

-u *user_list*

Specifies the user name or access list.

The *user_list* shall conform to the following syntax:

user_name[@host][,user_name[@host],...]

The utility shall allow only one user name for a given host.

The utility shall allow only one user name that is missing a corresponding host name.

-v *variable_list*

Defines the environmental variables to be exported.

A *variable_list* is a set of strings delimited by commas of either the form $< variable >$ or $< variable = value >$.

A variable shall not be repeated in the environment *variable_list* option-argument.

-verify

Causes DQS to print request to stdout. No action taken.

-V

Specifies that all environment variables be exported to the context of the job.

1.14.4 Operands

1.14.5 External Influences

Standard Input

None.

Environmental Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

1.14.6 External Effects

Standard Output

Standard Error

Output Files

1.14.7 Extended Description

1.14.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.14.9 Consequences of Errors

1.14.10 Unresolved Items

1.15 qsig - Signal Session Leader of Batch Jobs

1.15.1 Synopsis

qsig [**-cell** *cell_name*] [**-help**] [**-s** *signal*] [**-verify**] [*job_identifier_list*]

1.15.2 Description

The `qsig` utility provides a means for a user to send a signal to the session leader of a job.

The `qsig` utility shall signal those jobs, and only those jobs, for which a job identifier is presented.

The `qsig` utility shall signal jobs in the order in which the corresponding job identifiers are presented to the utility.

If the `qsig` utility fails to process any job identifier successfully, the utility shall proceed to process the remaining job identifiers, if any.

The `qsig` utility shall signal jobs by sending a Signal Job Request to the batch server that manages each job.

For each successfully processed job identifier, the `qsig` utility shall have received a completion reply to each Signal Job Request sent to the batch server at the time the utility exits.

1.15.3 Options

The command options are as follows:

-cell *cell_name*

The cell to which the request is forwarded.

-help

Prints a listing of all options.

-s *signal*

Specifies the signal to send to a job.

-verify

Causes DQS to print request to stdout. No action taken.

1.15.4 Operands

job_identifier_list

Specifies the interpreting shell, in the form:

sequence_number[@cell][,sequence_number[@cell],...]

If the application omits the *cell* portion of a job identifier, the utility shall use either:

— the name of the cell specified on the command line, if any.

— the name of the cell specified in the environment variable DQS_CELL, if it is set

— the name of the default cell

If the application presents the **cell** portion of a job identifier to the `qsig` utility, the utility shall send the Signal Job Request to the specified cell.

1.15.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.15.6 External Effects

Standard Output

An implementation of the utility may write informative messages to Standard Output.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.15.7 Extended Description

None.

1.15.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.15.9 Consequences of Errors

1.15.10 Unresolved Items

1.16 qstat - Display Status of Batch Jobs

1.16.1 Synopsis

qstat [**-cell** *cell_name*] [**-ext**] [**-f**] [**-help**] [**-jid** *job_identifier,[,destin_id,...]*] [**-l** *resource_list*] [**-u** *user_list*] [**-q** *destination_identifier_list*]

1.16.2 Description

The `qstat` utility provides a means for a user to display the status of one or more jobs, queues, or batch servers.

The `qstat` utility shall acquire job status information by sending a Job Status Request to a batch server. The `qstat` utility shall acquire queue status information by sending a Queue Status Request to a batch server. The `qstat` utility shall acquire server status information by sending a Server Status Request to a batch server.

1.16.3 Options

The command options are as follows:

-cell *cell_name*

The cell to which the request is forwarded.

-ext

Specifies that an extended description should be printed.

-f

Specifies full format display.

-help

Prints a listing of all options.

-jid *job_identifier,[,destin_id,...]*

Defines the job identifiers to act upon.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],...]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

A `resource_list` consists of one or more `resource` specifiers followed by an optional `op` field and a numeric or unsigned integer value.

When comparing resource values, the `qselect` utility shall use the following definitions for the *op* component of the option-argument:

.eq. – the value of the resource of the same name in the resource-list attribute of the job equals the value of the *value* component of the option-argument

.ge. – the value of the resource of the same name in the resource-list attribute of the job is greater than or equal to the value of the *value* component of the option-argument

.gt. – the value of the resource of the same name in the resource-list attribute of the job is greater than the the value of the *value* component of the option-argument

.lt. – the value of the resource of the same name in the resource-list attribute of the job is less than the the value of the *value* component of the option-argument

.le. – the value of the resource of the same name in the resource-list attribute of the job is less than or equal to the value of the *value* component of the option-argument

.ne. – the value of the resource of the same name in the resource-list attribute of the job is not equal to the the value of the *value* component of the option-argument

Only equality or inequality are valid comparisons if the `resource_field` is non-numeric.

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"

-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

1.16.4 Operands

None.

1.16.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.16.6 External Effects

Standard Output

If the **-f** option is not specified, the utility shall display the following items on a single line, in the stated order, with white space between each, for each successfully processed operand:

— the queue name

— the job identifier

— the job name

— the job owner

— the job state

— the job location

If the **-f** option is specified, the utility shall display the following items for each successfully processed operand:

— the queue name

— the type of queue (execution or routing)

— the maximum number of jobs allowed to run in the queue concurrently.

— the total number of jobs in the queue

— the enabled or disabled status of the queue

— the started or stopped status of the queue

— the job identifier

— the job name

— the job owner

— the job state

— the job location

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.16.7 Extended Description

None.

1.16.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.16.9 Consequences of Errors

1.16.10 Unresolved Items

1.17 qsub - Submit Script

1.17.1 Synopsis

qsub [**-a** *date_time*] [**-A** *account_list*] [**-cell** *cell_name*] [**-cwd**] [**-C**] [**-e** *path_list*] [**-h**] [**-hard**] [**-help**] [**-j** *y* ‖ *n*] [**-l** *resource_list*] [**-m** *mail_options*] [**-master** *destination_identifier, ...*] [**-M** *mail_list*] [**-notify**] [**-N** *name*] [**-o** *path_list*] [**-p** *priority*] [**-passwd**] [**-passwd_file** *filename*] [**-q** *destination_identifier_list*] [**-r** *y*‖*n*] [**-reauth** *seconds*] [**-soft**] [**-S** *path_list*] [**-v** *variable_list*] [**-verify**] [**-V**] [**-z**] [*script*]

1.17.2 Description

The qsub utility provides a means for a user to create a job that will execute the submitted script.

The qsub utility shall submit a script by sending a Queue Job Request to a batch server.

The qsub utility shall place the value of the following environment variables in the variable-list attribute of the job:

HOME , **LOGNAME** , **PATH** , **MAIL** , **SHELL** , and **TZ**.

The name of the environment variable shall be the current name prefixed with the string {DQS_O_}.

In addition to the variables described above, the The qsub utility shall add the following variables with the indicated values to the variable list:

— DQS_O_WORKDIR — the absolute path of the current working directory of the qsub utility process.

— DQS_O_HOST — the name of the host on which the qsub utility is running.

Following is a summary of information related to the environment variables:

Variable Name	Value at qsub time
DQS_O_HOME	$HOME
DQS_O_LOGNAME	$LOGNAME
DQS_O_PATH	$PATH
DQS_O_MAIL	$MAIL
DQS_O_SHELL	$SHELL
DQS_O_TZ	$TZ
DQS_O_WORKDIR	current working directory
DQS_O_HOST	client host name

1.17.3 Options

The command options are as follows:

-a *date_time*

The time at which a job is eligible for execution.

The *date_time* option-argument conforms to the following syntax:

[[CC]YY]MMDDhhmm where:

`CC` Specifies the first two digits of the year.
`YY` Specifies the last two digits of the year.
`MM` Specifies the month of the year (01 to 12).
`DD` Specifies the day of the month (01 to 31).
`hh` Specifies the hour of the day (00 to 23).
`mm` Specifies the minute of the hour (00 to 59).

The execute-time attribute of the job is defined as the number of seconds since Epoch that is equivalent to the local time expressed by the value of the *date_time* argument.

If the **-a** option is not presented to the `qsub` utility, the utility shall set the execute-time attribute of the job to a time that is earlier than the time at which the utility exits.

-A *account_list*

The account under which a job is processed.

The syntax of the *account_list* option-argument is as follows:

account_name[@cell][,account_name[@cell],...]

The `qsub` utility shall set the account-name attribute of the job to the value of the *account_list* option-argument.

If the **-A** option is not presented to the `qsub` utility, the utility shall omit the account-name attribute from the attributes of the job.

-cell *cell_name*

The cell to which the request is forwarded.

-cwd

Specifies to use the current working directory at job exec time.

-C

Defines the prefix that declares a directive to the utility.

The directive prefix is not a job attribute, but affects the behavior ob the utility.

If the **-C** option is presented with the value of the *directive _prefix* as a null string, the utility will not scan the scriptfile for directives.

-e *path_list*

The path used for standard error stream of the job with the following form:

[hostname1:]/usr[,hostname2:]/tmp,...]

For example:

-e *hostname:/user/users/green/aix*
-e *hostname:aix*

A relative path will be relative to the *home* directory for the user on said machine.

If the *path_list* option-argument constitutes an absolute path name, the error-path attribute of the job shall be set to the value of the *path_list* option-argument, including the hostname, if present.

If the *path_list* option-argument constitutes a relative path name, and no hostname is specified, the error-path attribute of the job shall be set to the value of the absolute path name derived by expanding the *path_list* option-argument relative to the current directory of the process that executes the utility.

If the *path_list* option-argument constitutes a relative path name, and a hostname is specified, the error-path attribute of the job shall be set to the value of the *path_list* option- argument without expansion. The hostname shall be included.

If the *path_list* option-argument does not include a hostname, the path name in the error-path attribute of the job shall be given the prefix, *hostname:*, where *hostname* is the name of the host upon which the utility is being executed.

The default file name for standard error has the following form:

*job_name.*e*sequence_number*

If the **-e** option is not presented to the qsub utility, the utility shall set the error-path attribute of the job to the hostname and path of the current directory of the submitting process and the default file name.

-h

Specifies that a "USER" hold be placed on the job by setting the value of the hold-types attribute of the job to the value {NO_HOLD}.

If the **-h** option is not presented to the qsub utility, the utility shall set the hold-types attribute of the job to the value {NO_HOLD}.

-hard

Specifies that subsequent resource requests must be satisfied in full before a job is to be scheduled.

-help

Prints a listing of all options.

-j *y* || *n*

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

If the value of the option-argument is 'y' the join-paths of the job whall be set to {TRUE}.

If the value of the option-argument is 'n' the join-paths of the job whall be set to {FALSE}.

If both the **-j y** and the **-e** options are specified, the batch server shall set but ignore the error-path attribute.

If the **-j** option is not presented to the qsub utility, the utility shall set the value of the join-paths attribute to {FALSE}.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],...]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

If the **l** option is not presented to the `qsub` utility, the utility shall omit the resource-list attribute from the job attributes.

-m *mail_options*

Defines the points during execution of the job at which the server is to send mail as to the state of the job.

The *mail_list* option argument shall conform to the following:

— 'a' — {MAIL_AT_ABORT}
— 'b' — {MAIL_AT_BEGINNING}
— 'e' — {MAIL_AT_EXIT}
— 'n' — {NO_MAIL}
— 's' — {MAIL_AT_SUSPENSION}

Repetition of any of the characters within the *mail_options* option-argument shall be permitted, but the utility will not assign additional meaning to the additional characters.

If the **-m** option is not presented to the qsub utility, the utility shall set the mail-points attribute to the value {MAIL_AT_ABORT}.

-master *destination_identifier, ...*

Defines the queue to serve as the master for said job.

The *destination_identifier* option argument shall conform to the following:

queue[-exec.eq.exec_str][,queue[-exec.eq.exec_str]...]

DQS will attempt to assign the *master* based on a first to last transvesal of the *destination_identifier_list*. If the *destination_identifier_list* is specified as a *soft* request and DQS is unable to fulfill the master request based on the respective *destination_identifier_list*, DQS will assign a master as available.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-M *mail_list*

Specifies the list of users to which mail about the job will be sent by the server, if the server sends such mail.

If the implementation of the utility uses a name service to locate users, the utility shall require and accept the syntax used by the name service.

If the implementation of the utility does not use a name service, the following syntax for user names will be required and accepted:

> *mail_address [,mail_address,...]*
> where *mail_address* shall conform to:
> *user[@hostname]*

The utility shall set the mail-users attribute of the job to the value of the *mail_list* option-argument.

If the **-M** option is not presented to the qsub utility, the utility shall place only the user name and host name for the current process in the mail-users attribute of the job.

-notify

Specifies that the job should be signaled via a *SIGUSR1* on an impending *SIGSTOP* and a *SIGUSR2* on an impending *SIGKILL*.

-N *name*

The name of a specific job.

The *name* option-argument shall be any printable ascii string up to 8 characters. Names in excess of 8 characters will be truncated.

The qsub utility shall set the value of the job-name attribute of the job to the value of the *name* option-argument.

If the **-N** option is not presented to the qsub utility, the utility shall set the job-name attribute of the job to the name of the script argument from which the directory specification, if any, has been removed.

If the **-N** option is not presented to the qsub utility, and the script is read from standard input, the utility shall set the job-name attribute of the job to the value {STDIN}.

-o *path_list*

The pathname for standard output.

For example:

-e *hostname:/user/users/green/aix*

-e *hostname:aix*

A relative path will be relative to the *home* directory for the user on said machine.

If the *path_list* option-argument constitutes an absolute path name, the **stdout-path** attribute of the job shall be set to the value of the *path_list* option-argument, including the hostname, if present.

If the *path_list* option-argument constitutes a relative path name, and no hostname is specified, the **output-path** attribute of the job shall be set to the value of the absolute path name derived by expanding the *path_list* option-argument relative to the current directory of the process that executes the utility.

If the *path_list* option-argument constitutes a relative path name, and a hostname is specified, the **output-path** attribute of the job shall be set to the value of the *path_list* option- argument without expansion. The hostname shall be included.

If the *path_list* option-argument does not include a hostname, the path name in the **output-path** attribute of the job shall be given the prefix, *hostname:*, where *hostname* is the name of the host upon which the utility is being executed.

The default file name for standard output has the following form:

*job_name.***o***sequence_number*

If the **-o** option is not presented to the `qsub` utility, the utility shall set the **output-path** attribute of the job to the hostname and path

-p *priority*

Specifies the priority of the job relative to other jobs.

The **priority** attribute of the job shall be set to the value of the *priority* option-argument.

The utility shall require and accept a value for the *priority* option-argument that conforms to the syntax for signed decimal integers, and which is not less than -1024 and not greater than 1023.

If the **-p** option is not presented to the `qsub` utility, the value of the **priority** attribute is implementation defined.

-passwd

Specifies that a password list will be requested for AFS and/or DFS reauthentication. If given, DQS shall utilize the password list to authenticate prior to executing the job and reauthenticate as necessary.

The utility shall either prompt for password(s) to be entered interactively until CTL-D is entered.

-passwd_file *filename*

The name of the file in which the password(s) reside.

Entries in the file will be in the format:

password cell principal passwd
password cell principal passwd
...

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"

-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

The `qsub` utility shall submit the script to the batch server named by the *destination* option-argument, or the server that owns the queue named in the *destination* option-argument.

If the option-argument describes only a queue the `qsub` utility shall queue the job at the queue of the specified name at the default batch server.

If the option-argument describes only a batch server, the `qsub` utility shall submit the script to that designated batch server.

If the option-argument describes both a queue and a batch server, the `qsub` utility shall queue the job in the specified queue at the specified server.

-r *y||n*

Identifies the ability of a job to be rerun or not.

An option-argument value of '**y**' corresponds to a value of {TRUE} for the rerunable attribute of the job.

An option-argument value of '**n**' corresponds to a value of {FALSE} for the rerunable attribute of the job.

If the **-r** option is not presented to the `qsub` utility, the utility shall set the rerunable attribute of the job to {FALSE}.

-soft

Specifies that subsequent resource and destination identifiers are to be filled on an "as available" basis.

-reauth *seconds*

Specifies the interval in which to reauthenticate.

-S *path_list*

Specifies the interpreting shell.

The *path_list* shall conform to the following syntax:

pathname[@host][,pathname[@host],...]

The utility shall allow only one path name for a given host.

The utility shall allow only one path name that is missing a corresponding host name.

If the **-S** option is not presented to the `qsub` utility, the utility shall set the shell-path-list attribute of the job to the null string.

-v *variable_list*

Defines the environmental variables to be exported.

A *variable_list* is a set of strings delimited by commas of either the form $< variable >$ or $< variable = value >$.

A variable shall not be repeated in the environment *variable_list* option-argument.

If the **-v** option is presented to teh `qsub` utility, the utility shall also add to the environment variable-list attribute of the job every variable named in the *variable_list* option-argument and, optionally, values of specified variables.

-verify

Causes DQS to print request to stdout. No action taken.

-V

Specifies that all environment variables be exported to the context of the job.

The `qsub` utility shall place in the list every environment variable in the process in which the utility is executing. It shall also set the value of each variable in the attribute to the value of that variable in the process.

-z

Specifies that utility shall not write the job identifier to the standard output.

If the **-z** option is presented to the utility, the utility shall not write the job identifier of the created job to standard output.

If the **-z** option is not presented to the utility, the utility shall write the job identifier of the created job to standard output.

1.17.4 Operands

script

The path to the script of the job.

If the *script* operand is not presented to the utility, or if the operand is the single-character string '-', the utility shall read the script from standard input.

If the *script* represents a partial path, the utility shall expand the path relative to the current directory of the process executing the utility.

1.17.5 External Influences

Standard Input

The `qsub` utility reads the script of the job from standard input if the *script* operand is omitted or is the single character '-'.

Input Files

In addition to binding the file indicated by the *script* operand to the job, the `qsub` utility reads the *script* file and acts on directives in the script.

Environmental Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

The values of some or all of the environment variables of the process that executes the `qsub` utility are placed in the attributes of the job.

The environment variable DQS_DPREFIX determines the default prefix for directives within the script.

Asynchronous Events

Once created, a job exists until it exits, aborts, or is deleted.

After a job is created by the `qsub` utility, batch servers might route, execute, modify, or delete the job.

1.17.6 External Effects

Standard Output

The qsub utility writes the job identifier assigned to the job to standard output.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.17.7 Extended Description

Script Preservation

The qsub utility shall make the script available to the server executing the job in such a way that the server executes the script as it exists at the time of submission.

The qsub utility can send the script available to the server with the Queue Job Request or store a temporary copy of the script in a location specified to the server.

Option Specification

A script can contain directives to the qsub utility.

The qsub utility shall scan the lines of the script for directives until the first line that begins with a string other than the directive string; if directives occur on subsequent lines, the util shall ignore those directives.

The qsub utility shall process a line in the script as a directive if, and only if, the string of characters from the first non-white space character on the line until the first space or tab character on the line match the directive prefix.

The qsub utility shall ignore any blank lines and shell comment lines prior to the first executable line.

The qsub utility shall process the options and option-arguments contained on the directive prefix line using the same syntax as if the options were input on the qsub utility.

The qsub utility shall continue to process a directive prefix line until after a < *newline* > or an unquoted comment character of the shell that will interpret the script is encountered.

If an option is present in both a directive and arguments to the qsub utility, the utility shall ignore the option and the corresponding option-argument, if any, in the directive.

If an option that is present in the directive is not present in the arguments to the qsub utility, the utility shall process the option and the option-argument, if any.

In order or preference, the `qsub` utility shall select the directive prefix from one of the following sources:

— if the **-C** option is presented to the utility, the value of the *directive_prefix* option-argument.

— if the environment variable POSIX2_DPREFIX is defined, the value of that variable.

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

1.17.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero. The utility shall have failed if any of the following conditions apply:

— if the argument list presented to the `qsub` utility does not conform to the synopsis.

— if any option-argument does not conform to the syntax described for it.

1.17.9 Consequences of Errors

1.17.10 Unresolved Items

The following options are yet to be implemented:

-A account_list
-B
-c interval
-h
-k
-m mail_options
-M mail_list -p
-u user_list
-z
-

The following environmental variables are yet to be incorporated:

DQS_DPREFIX
POSIX_DPREFIX

1.18 qusage - Show DQS Queue Usage

1.18.1 Synopsis

qusage [**-d** *display*] [**-days** *n*] [**-start** *n*] [**-q** *destination_identifier_list*] [**-u** *user_list*] [**-cpu**] [**-l** *resource_list*] [**-help**] [**-verify**]

1.18.2 Description

The `qusage` utility provides an W-Window interface to graphically display the utilization of the DQS queues over a given period of time.

The `qusage` utility uses two different formats:

— The default format displays a graph based on the DQS jobs that have already finished. Jobs that were lost due to a machine crash or jobs that are still running will not be depicted in this graph.

— The second format requires the **-l** option. The graph that results from this format depicts the DQS usage over periodic polling of the DQS states. It therefore also shows the queues that were running, suspended or idle. The usage percentage of this graph is based upon only the queues that are either running or idle.

1.18.3 Options

The command options are as follows:

-d *display*

Display name. See X(1) for more information.

-days *n*

Show the usage for the past *n* days since *-start*.

-l *resource_list*

The resources requested by the job.

The *resource_list* option argument shall conform to the following:

resource[.op.resource_field][,resource[.op.resource_field]],...]

There are 2 reserved words in the resource list. *qty* which specifies the number of such requests. *qty* is optional and defaults to 1.

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-start *n*

Begin *n* days ago.

Default is 7.

-q *destination_identifier_list*

Specifies a *destination_id_list*.

The *destination_id* is not a job attribute; it determines the queue, server or queue at a server to which the job is queued.

The *destination_id_list* option argument shall conform to the following:

destination_id[-exec.eq.exec_str][,destination_id[-exec.eq.exec_str]...]

The *exec_str* may optionally contain the following wildcards which will be expanded at runtime:

%execer_id - set to "DQS"
%master_host - expanded to the master's hostname at runtime
%my_hostname - expanded to my hostname at runtime
%my_nodenum - expanded to my node number at runtime
%total_numnodes - expanded to the total number of nodes at runtime
%remote_info - expanded to remote exec_strs at runtime. Only passed to "master"

A few examples of an *exec_str*:

-master ibm1-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-q ibm1-exec.eq."my_prog arg1 arg2 %execer_id %total_numnodes"
-l group.eq.ibm,-exec.eq."my_prog arg1 arg2 %execer_id %master_host %my_nodenum arg3"
-l qty.eq.3,exec.eq."my_prog arg1 %remote_info"

Additionally the environmental variable *REMOTE_INFO* is passed in at exec time of the script_file.

Starting of P4 and MPI jobs only requires the program names and optionally any arguments YOU might need - no wildcards.

-u *user_list*

Specifies the user name or access list.

The *user_list* shall conform to the following syntax:

user_name[@host][,user_name[@host],...]

The utility shall allow only one user name for a given host.

The utility shall allow only one user name that is missing a corresponding host name.

-cpu

Report cpu utilization.

-help

Prints a listing of all options.

-verify

Causes DQS to print request to stdout. No action taken.

1.18.4 Operands

1.18.5 External Influences

Standard Input

None.

Input Files

None.

Environment Variables

DQS_CELL If set, specifies the default DQS cell.

DQS_DEBUG If set, specifies that debug information should be written to stderr.

DQS_EXT_DEBUG If set, specifies that extended debug information should be written to stderr.

Asynchronous Events

None.

1.18.6 External Effects

Standard Output

None.

Standard Error

The utility writes diagnostic messages to Standard Error.

Output Files

None.

1.18.7 Extended Description

None.

1.18.8 Exit Status

Upon successful processing of all the operands presented to it, the utility shall exit with a value of zero.

If the utility fails to process any operand, it shall exit with a value greater than zero.

1.18.9 Consequences of Errors

1.18.10 Unresolved Items