

BQplus™
Job scheduler

Advanced Job Scheduling for UNIX

User guide for version 5.0

(c) 1993-2002 Corporate Practical Solutions Limited.

All rights reserved. This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and decompilation. No part of this product may be reproduced in any form by any means without prior written authorization of Corporate Practical Solutions Limited.

This publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein; these changes may be incorporated in new editions of the publication. Corporate Practical Solutions Limited may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Solaris, the Solaris Sunburst, and the Solaris Compatible Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

SCO and the SCO OK icon are registered trademarks of The Santa Cruz Operation, Inc. in the U.S.A and other countries.

All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

Corporate Practical Solutions Limited.
End-User Binary Code
SOFTWARE LICENSE AGREEMENT

Corporate Practical Solutions Limited ("CPS") is willing to license the accompanying software to you only upon the condition that you accept all of the terms contained in the license agreement. Read the terms and conditions of this license carefully before installing and/or licensing the software. By installing and/or licensing the software, you agree to the terms and conditions of this agreement. If you are not willing to be bound by this agreement, uninstall the Software and return, if purchased, the license keys, unused within fifteen (15) days of purchase for a refund of the license fee paid.

1. License to Use. Customer is granted a non-exclusive and non-transferable license ("License") for use of the Bqplus (BQ+) or PQplus (PQ+) software in machine-readable form, together with accompanying documentation ("Software"), by the class of computer hardware for which the corresponding fee has been paid.

2. Restrictions. Software is copyrighted and CPS and/or its licensors retain title to all copies. Customers shall not make copies of Software, other than a single copy of Software for archival purposes. Exempt where applicable law prohibits enforcement of this provision. Customer shall not modify, decompile, disassemble, decrypt, extract or otherwise reverse engineer Software. **Software is not designed or licensed for use in on-line control equipment in hazardous environments such as operation of nuclear facilities, aircraft navigation or control, or direct life support machines.**

3. Confidentiality. Software is confidential and proprietary information of CPS. Customer agrees to take adequate steps to protect Software from unauthorized disclosure or use.

4. Limited Warranty. CPS warrants that for a period of ninety (90) days from the date of purchase, as evidenced by a copy of the purchase order, that the Software contains the features described in the CPS Website <http://www.grepit.com> ("Website") at the date of purchase. Otherwise, the software is provided "AS IS". This limited warranty extends only to the Customer as the original licensee. Customer's exclusive remedy and CPS's entire liability under this limited warranty will be, at CPS's option, to repair or replace the Software, or refund the license fee paid therefore.

5. Disclaimer of Warranty. EXCEPT AS SPECIFIED IN THIS LICENSE, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW.

6. Limitation of Liability. IN NO EVENT WILL CPS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF CPS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event shall CPS's liability to the Customer, whether in contract, tort (including negligence), or otherwise, exceed the license fee paid by Customer for Software. The foregoing limitations shall apply even if the above stated warranty fails of its essential purpose.

7. Termination. This License is effective until terminated. Customer may terminate this License at any time by destroying all copies of the Software including any documentation. This License will terminate immediately without notice from CPS if Customer fails to comply with any provision of the License. Upon termination, Customer must destroy all copies of Software.

8. Integration. This agreement is the entire agreement with Customer and CPS relating to Software and : (i) supercedes all prior or contemporaneous oral or written communications, proposals and representations with respect to its subject matter; and (ii) prevails over any conflicting or additional terms of any quote, order, acknowledgment, or similar communication between the parties during the term of this agreement. No modification to the Agreement will be binding, unless in writing and signed by a duly authorized representative of each party.

Contents

CHAPTER 1 : WELCOME	9
How to use this manual	9
Symbols.....	10
CHAPTER 2 : INSTALLATION	11
Core product installation	11
Prerequisites	11
Procedure.....	11
Machine types.....	14
Licensing the BQplus software	14
Extending the evaluation period	14
Permanently licensing the software	14
Configuring your machine.....	16
Defining an alternative BQplus location	16
Configuring the PATH environment variable.....	16
Upgrading BQplus	16
CHAPTER 3 : USER GUIDE	19
Overview	19
Adding a request	19
Options	19
Modifying a request	31
Turning off an option	32
More options	33
Unholding a job.....	33
Retrying a job.....	33
Deleting a request.....	34
Listing the queue	35
Basic listing.....	35
Full listing.....	36
Selective listing.....	36
CHAPTER 4 : ADVANCED USER GUIDE.....	41
Job History.....	41
Environment variables.....	42
Options requiring a parameter	42
Options not requiring a parameter	43
Detecting that a program is running in BQplus	43
Adding a request	44
Advanced options	44
Job Suites.....	47
Submitting a job to a suite.....	47

Listing jobs in a suite.....	47
Suite definition	47
Suite parameters.....	48
Suite status	49
Using suites from within the BQplus menus	49
BQplus IPC (inter-process communication)	51
Action files	53
Remote BQplus working	54
CHAPTER 5 : ADMINISTRATOR GUIDE	55
Overview.....	55
The BQplus Controller	55
Starting the Controller	55
Stopping the Controller	56
Checking the Controller status	56
Checking the Controller log file	57
Signaling the Controller.....	57
BQplus Queues	58
Pausing a queue.....	58
Unpausing a queue	58
Listing the queues	58
Queue maintenance	58
Fixing the queue	58
The BQplus netserver	59
Starting the netserver.....	59
Stopping the netserver	59
Checking the netserver status.....	61
Checking the netserver log file.....	61
Auditing the netserver activity	61
CHAPTER 6 : ADVANCED ADMINISTRATOR GUIDE.....	63
Configuration files	63
Actions on error : 'actions.cfg'	63
BQplus defaults : 'defaults.cfg'	63
BQplus queues : 'queues.cfg'	70
BQplus controller parameters : 'control.cfg'	71
Privileged BQplus users : 'admin.cfg' and 'operator.cfg'	74
BQplus character menus : 'control.keys'	74
BQplus schedules : 'scheds.cfg'	76
Remote BQplus users : 'remote.cfg'	78
Accounting files.....	79
CHAPTER 7 : MENU SYSTEMS	81
Description	81

Character based menus : 'bqmenu'	81
Starting the menus	82
Configuring the control keys	82
Using the menu system	82
Printing the output displayed in the view screen	84
Changing BQplus host	84
Changing the help text	85
Using select boxes	86
Command line options	87
Maintaining a list schedule	88
CHAPTER 8 : THE WORLD WIDE WEBSITE	91
Come and visit us	91
Online documentation	91
CHAPTER 9 : APPENDICES	93
Keywords	93
Description	93
Available Keywords	93
Options supporting keywords	94
BQplus job states	94
Date and Time format	95
Defining the country for date syntax	96
Running multiple copies of BQplus on the same machine	97
Description	97

Chapter 1 : Welcome



BQplus is a job scheduler developed specifically for UNIX platforms. First released in 1993, BQplus has proven an effective Job Scheduling system for sites migrating from legacy systems where such facilities typically featured as part of the Operating system environment.

BQplus has been implemented in a wide variety of computing environments and is employed for a range of tasks from daily backups to fully automated lights-out operations. Key features of BQplus that make it the ideal job scheduling solution include:

- *Flexible user interface*
- *Advanced scheduling facilities*
- *Job dependencies*
- *Job suites*
- *Multiple queue environment*
- *Full job history*

BQplus is currently available on the following UNIX platforms :

- *Solaris SPARC*
- *HP-UX*
- *Siemens SINIX*
- *AIX*
- *OSF*
- *Sequent Dynix/Ptx*

For further information please visit the BQplus World Wide Website :

www.grepit.com

How to use this manual

This manual consists of BQplus User and BQplus Administrator guides. Each of these guides is split into Basic and Advanced chapters, with the basic chapters allowing a quick start to the use of BQplus, while more advanced features are detailed separately and may not be required.

The menu interfaces are described in a chapter entitled 'Menus' and feature the ASCII (curses) menus.

Symbols

This following symbols are used throughout this manual :





<i>Symbol</i>	<i>Description</i>
	<i>Indicates a note, which should be read and understood before an option is used.</i>
	<i>Indicates an important piece of information.</i>
	<i>Indicates a tip, which may be used or influence the way in which BQplus is implemented.</i>
	<i>Indicates the default action taken by BQplus if an option/switch is not used.</i>

Table 1 : Symbols

Chapter 2 : Installation



Installing BQplus is a straightforward exercise, which should take no more than five minutes.

Core product installation

Prerequisites

- access to the 'root' account.
- approximately 2Mb of free disk space where BQplus is to be installed.
- approximately 2Mb of free disk space for use as a staging area. e.g. '/tmp'

Procedure

The following procedure should be followed to successfully install BQplus.

1. Login as 'root' on the machine on which BQplus is to be installed.	<input type="checkbox"/>
2. Create a UNIX group identifier (GID) named 'bqplus' ¹ .	<input type="checkbox"/>
3. Change directory to the staging area. (typically '/tmp')	<input type="checkbox"/>
4. Insert the BQplus distribution media in the drive.	<input type="checkbox"/>
5. Download the software from the CPS website. Open your browser and enter the following URL : www.grepite.com <ul style="list-style-type: none"> • Click on the download tab & select the correct Bqplus distribution for your machine type. • Save/FTP the tar set into the temporary directory. • Extract the tar set to create the install script and BQplus 	<input type="checkbox"/>

¹ This should be performed by the Systems Administrator

	<p>distribution tar set.</p> <ul style="list-style-type: none"> • Remove the web tar set. 	
6.	<p>Start the BQplus installation script using the following command:</p> <pre>sh installbq+ `machine_type`</pre> <p>Where `machine_type` should be replaced with your machine type².</p>	<input type="checkbox"/>
7.	<p>The `installbq+` script will first confirm that no BQplus processes are already running on the machine. If this is not the case you should exit the installation and stop the BQplus controller and any batch jobs before restarting.</p> <p>Having confirmed that BQplus is not already running on the machine you will be asked for the target directory into which BQplus should be installed. Pressing <return> will instruct the procedure to install into a `/usr/bq+` location.</p>	<input type="checkbox"/>
8.	<p>BQplus will now be extracted from the tar set into the specified target directory. A message will be displayed at the completion of the installation process.</p>	<input type="checkbox"/>

Table 2 : Core product installation procedure

² See table of machine types on page 14

Directory	Filename	Description	Mask ³
bq+		Home	775
➔	adm	Accounting files	770
	bin	Executables	775
	➔		6555
	batchq		6555
	bcon		500
	controller		6555
	bqmenu		500
	netserver		500
	bqhelper		700
	makehelp.sh		770
	config	Configuration files	770
	➔		770
	defaults.cfg		770
	control.cfg		770
	secure.cfg		770
	remote.cfg		770
	actions.cfg		770
	admin.cfg		770
	operator.cfg		770
	queues.cfg		770
	scheds.cfg		770
	control.keys		770
	bank.hols		770
	bqmenu.hlp		770
	<i>Example shell scripts</i>		555
	environments	Saved environment vars.	775
	tmp	BQplus temporary files	770
	scripts	Job execution scripts	775
	utilities	Utility programs	775
	➔		555
	nis_lookup.sh		555
	passwd_lookup.sh		555
	logs	BQplus logfiles	770
	queue	BQplus queue	770
	adm	Accounting files	770

Figure 1 : Core inventory

Congratulations, BQplus is now installed and will operate for an evaluation period of thirty days.

³ UID=root, GID=bqplus

Machine types

Machine	Distribution abbreviation
Solaris SPARC	solss
IBM AIX	aix
Hewlett Packard	hpux
Siemens Nixdorf	sinix
Sequent Dynix/Ptx	ptx4
Digital OSF	osf

Table 3 : Machine type abbreviations

Licensing the BQplus software

When BQplus is first installed an evaluation period of thirty days is allowed for use of the product. At the completion of the thirty days the BQplus controller process will not start. At this juncture two actions may be taken :

- Contact your BQplus distributor and obtain an evaluation extension license code.
- Contact your distributor and purchase the BQplus software to obtain a permanent license code.

Extending the evaluation period

1. Type the following command : 'bcon -extend'	<input type="checkbox"/>
2. You will now be provided with a unique security string which must be supplied to your software provider.	<input type="checkbox"/>
3. Your software provider will provide you with a unique license code which should now be typed in.	<input type="checkbox"/>

Table 4 : Evaluation license extension procedure

Your BQplus system will now be licensed for an additional 30 evaluation days. This can be checked by restarting the BQplus controller and inspecting the controller log file using the command 'bcon -log'.

Permanently licensing the software

To permanently license the BQplus software the following procedure should be followed :

1. Determine the machine name of the node on which BQplus has been purchased. This can easily be found by viewing the BQplus controller log file using the command 'bcon -log'.	<input type="checkbox"/>
2. Contact your software distributor with the machine name and UNIX platform information to hand. You will be given an eight character permanent license code.	<input type="checkbox"/>
3. License the software using the following procedure : <ul style="list-style-type: none">• Login as 'root'.• Change directory to the BQplus config directory. (e.g. '/usr/bq+/config')• Edit the controller configuration file 'control.cfg' and insert your unique 8 character license code where the 'license_code' directive is found.	<input type="checkbox"/>
4. BQplus will now be permanently licensed. You can validate the permanent license code by starting the controller and inspecting the controller log file using the command 'bcon -log'.	<input type="checkbox"/>

Table 5 : Permanent licensing procedure

Congratulations, you are now an official BQplus user!

Configuring your machine

While BQplus will function immediately, you may choose to make some simple modifications to integrate BQplus in the machine environment.

Defining an alternative BQplus location

BQplus will by default operate in the location `"/usr/bq+"`. If the software is installed in different location two ways of indicating this are possible :

- Create a symbolic link `"/usr/bq+"` which points to the actual BQplus directory.

eg. In -s /opt/bq+ /usr/bq+



Note :

If appropriate this will normally be performed during the installation procedure.

- Create a BQ_HOME environment variable that indicates the location of BQplus.

e.g. BQ_HOME=/opt/bq+; export BQ_HOME

Configuring the PATH environment variable

The PATH environment variable is used to indicate those directories that contain programs that are available for execution. Multiple directories are separated by colons `":"` and are searched in the order that they have been defined. BQplus programs reside in the BQplus `"bin"` directory. This directory may be included in the PATH environment variable so that commands such as `"batchq"` can be typed without the need for a full pathname. To include BQplus in the PATH the following command should be used :

***PATH=\$PATH:/usr/bq+/bin
export PATH***

Where `"/usr/bq+"` can be replaced by an alternative BQplus location, as appropriate.

If the BQ_HOME environment variable has been defined you may of course use the following command :

***PATH=\$PATH:\$BQ_HOME/bin
export PATH***

Upgrading BQplus

When BQplus is installed, a check is made for a previous version of the software. If

a previous version is found, the queue will be converted if necessary and all the jobs in it will be retained. This is performed by the installation script 'installbq+', and the BQplus administrator should not normally need to use the option shown below. *This is only possible for upgrades from BQplus v3.4 and higher, and is not generally supported for pre-release versions of BQplus.*



Note :

All configuration files will be retained by an upgrade of BQplus. This means that any new options will potentially be lost from configuration files when the old files are restored. Please refer to the section 'Configuration files' on page 63 to include any new options that you may want to use.

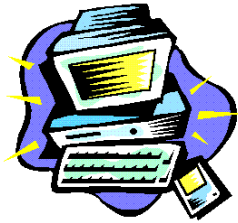
Converting an old BQplus system greatly reduces the amount of time for an upgrade of BQplus, particularly if the old queue has a lot of jobs in it.



Note :

New features in the updated software will be turned off for any jobs that are converted. You may however modify any queued jobs after converting the queue to take advantage of any new functionality.

Chapter 3 : User Guide



This section provides all the information for a user to be able to submit, modify and delete BQplus jobs. In addition the user will be able to inspect the BQplus queue for job progress and status.

Examples in this section use the command line interface, for details on the two menu systems please refer to the 'Menus' chapter later in this document.

Overview

BQplus is a job scheduling system that consists of one or many job queues. Each queue has individual characteristics such as limits on how many jobs can be started and what time of day this is permitted.

When you submit a job to BQplus it is placed in one of the job queues. Once submitted the job will be started by the BQplus controller at the appropriate time. Output generated by a BQplus job can be directed to a log file which can be examined when the job is running or has completed. The status of a BQplus job can be obtained by listing the queue which can be achieved in many different ways and to differing levels of granularity.



Note :

Although BQplus supports any number of queues, the actual file system representation of these queues is a single file '\$BQ_HOME/queue/queue'.

Adding a request

In order to add a request to BQplus, the 'batchq' command should be used.

\$ batchq filename

A new BQplus job will be created in the default BQplus queue which at runtime will start the command 'filename'.

```
# batchq backup.sh
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Request 2 added to queue immediate : /scripts/backup.sh
```

Figure 2 : Adding a request

Options

A variety of options to the 'batchq' command are provided to access all of the BQplus functionality. Each option starts with the '-' character and can be combined

with other options on the same command line. Options can be specified in any order on the command line, but must only occur once in each BQplus command.

Common options are covered in this chapter with more advanced options detailed in the 'Advanced User Guide'.



Note :

If any option argument contains any spaces or reserved characters then it must be protected by quotation marks. This is shown in most of the examples that follow.

Job arguments

-arguments 'argument(s)'

abbreviations : *-args, -arg*

Where 'argument(s)' are the arguments that should be supplied to the program being run by BQplus.

```
# batchq backup.sh -args daily
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Request 3 added to queue immediate : /scripts/backup.sh
```

Figure 3 : Job arguments

If more than one argument is required then the argument list should be enclosed by quotation marks. i.e. " or '. Quotation marks should also be used if the arguments contain any reserved UNIX characters. e.g. *,\$ or &.

```
# batchq backup.sh -args "daily incremental"
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Request 4 added to queue immediate : /scripts/backup.sh
```

Figure 4 : Multiple job arguments



Note :

Multiple arguments containing arguments which themselves contain a space can be used with single quotes, as follows :

"argument1 'argument2 containing spaces' argument3"

Using a specific BQplus queue

-queue 'queue_name'

Where 'queue_name' is a valid BQplus queue name.

The job will be placed in the specified BQplus queue. Should the queue not be

active, e.g. 'PAUSED' or outside the active time window, the job will remain in the queue until it becomes possible to start it.



Tip :
You can get a list of BQplus queues using 'bqmenu -list_the_queues'



Default :
If no queue is specified the default queue will be used.

Log files



Tip :
To direct the output of a job to your current terminal, use a logfile of 'tty'

-logfile 'logfile_name'
abbreviations : -log_file, -log

Where 'logfile_name' is the name of the file that should be created by the BQplus job and will contain all stdout and stderr generated when the job is started.



Default :
A logfile named 'bqnnn.log' will be created in the home directory of the user that has submitted the job. 'nnn' will be replaced by the entry number of the BQplus request.

```
# batchq backup.sh -log backup.log
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Request 7 added to queue immediate : /scripts/backup.sh
```

Figure 5 : Requesting a log file



Note :
It is possible to configure the location where default log files will be created. Please refer to the section 'BQplus defaults : 'defaults.cfg'' on page 63 for an explanation of how this can be achieved.

Turning off the log file

A log file can be turned off in two ways :

-no_log
-logfile -none-

Any output generated by the job will be directed to the standard UNIX null device '/dev/null'.

Appending output to an existing log file



Tip :
Use this option if you have a recurring job (eg. On a schedule) and you want to view job log history.

-append

If this option is used and the logfile already exists, any output will be added to the end of the file.



Default :

A log file will overwrite an existing file of the same name.



Note :

If logfiles are appended they will increase in size over a period of time, particularly if the job is running on a frequent schedule. It is important therefore that some kind of logfile housekeeping mechanism is put in place to ensure that disk space usage is controlled.

Start time

-start_time 'date/time'

abbreviations : -start, -after, -af

Where 'time' is a valid date or time string as defined in Table 17 : Date/time syntax on page 96.

```
# batchq backup.sh -start 16/6/03.22:00
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Earliest start time for job : 16 Jun 2003 @ 22:00:00
Request 12 added to queue immediate : /scripts/backup.sh
```

Figure 6 : Specifying a start time

The job will be placed in the BQplus queue and will not start until the specified date/time has been reached.



Default :

The job will start as soon as possible.



Note :

If a start time is specified in addition to a schedule, the job will run at the start time and will apply the schedule subsequently.

Setting a latest start time

In addition to being able to specify a start time ('-start'), users may also specify a time by which the job must have started. This is achieved by specifying the maximum number of minutes that can elapse from a specified start time before the job is started. This is available through the use of the following option :

-max_delay 'nmins'

For example :

To schedule a job that must start after 10:00 but before 11:00.

\$ batchq cheque_run.sh -start 10:00 -max_delay 60

Should the controller find a job that has expired beyond its 'start_by' time, the entry

will be considered to have failed. A message indicating the event will be written to the job logfile and the entry cancelled or rescheduled (if running on a schedule) as appropriate.



Note :

To turn off this feature the 'max_delay' time should be specified as '0'.



Default :

No latest start time is considered.

Shell type

-shell 'shell_type'

Where 'shell_type' is one of the following shells :

- sh (bourne)
- ksh (korn)
- csh ('C')

Having specified a shell type the BQplus request will be executed from within this shell.



Default :

If no shell is specified the default shell will be used. This is configured by the BQplus administrator (page 63).

Runtime directory

-runtime_dir 'directory_name' *abbreviations : -run_dir*

Specify the directory from which the BQplus job should be run. The directory name can be specified as an absolute or relative path.



Default :

The job will run from the directory where the job is submitted.

Environment variables



Tip :

Use this option to ensure that the current environment is fully restored when a job is run. For example, 'Oracle' jobs will not run without the appropriate environment variables in place.

-envs

Save the current environment and restore all environment variables prior to job execution.



Default :

Environment variables are not restored.

Job Reference

-reference 'reference text'

abbreviations : -ref

Attach a line of text to the queue entry, to be used as a reference when the queue is listed or the logfile inspected. Up to 32 characters of useful information can be provided as a job reference, and can greatly help the understanding of anyone looking at a large queue listing or job logfile.



Default :

No reference text is applied.

Dependency on other BQplus jobs (by entry number)

-wait_for e1,e2,...e16

abbreviations : -wf

Instruct BQplus to place a job in the queue with dependency on up to 16 other BQplus jobs, which are referenced by their entry number. The entry number may be prefixed by a special character to indicate dependency on success, failure or just completion.

Special character	Meaning	Example
s	Dependent on success, indicated by a zero exit code.	s99
e	Dependent on failure, indicated by a non zero exit code.	e99
c	Dependent on completion, exit code is not considered.	c99

Continued

Table 6 : Modes for dependency by job entry number



Tip :

If you use dependency on hand crafted programs, ensure that exit codes of the programs be executed are always correct.

```
# batchq backup.sh -wait_for 10
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Request 13 added to queue immediate : /scripts/backup.sh
```

Figure 7 : Waiting for a job to complete



Default :

If no special character is given, the job will wait for successful completion.

If wait fors are not specified, no job dependency will apply.

Dependency on other BQplus jobs (by job reference)

-wait_for_reference r1,r2,...r8

abbreviations : -wait_for_ref, -wfr

Instruct BQplus to place a job in the queue with dependency on up to 8 other BQplus jobs, which are referenced by their job reference. The job reference may be prefixed by a special character and a period ('.') to indicate dependency on success, failure or just completion.

Special character	Meaning	Example
s	Dependent on success, indicated by a zero exit code.	s."backup job"
e	Dependent on failure, indicated by a non zero exit code.	e."backup job"
c	Dependent on completion, exit code is not considered.	c."backup job"

Table 7 : Modes for dependency by job reference



Tip :

If you use dependency on hand crafted programs, ensure that exit codes of the programs being executed are always correct.



Tip :

Use job dependency by job reference to allow easy queue analysis, and the use of pre-coded scripts to submit job suites containing dependencies.



Note :

Because the job reference in the examples shown contains a space, it is necessary to include quotation marks.

```
# batchq backup.sh -wait_for_ref "check_tape"
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Request 14 added to queue immediate : /scripts/backup.sh
```

Figure 8 : Waiting for a job to complete (by reference)



Default :

If no special character is given, the job will wait for successful completion.

If wait fors are not specified, no job dependency will apply.

Flagfile Dependency

-flagfile 'flagfile'

abbreviations : -ff

This option will request that the job only runs if a particular file exists on the system and is readable by the UNIX ID that is executing the job.

Should the flagfile not exist, the job will be considered to have failed, and will be deleted from the queue or retained in an error state if the job was submitted using the '-retain' option.

 **Default :**

No flagfile dependency is applied.

 **Note :**

You may provide more than one flagfile by using a space character to separate the filenames. However, the maximum combined length of the flagfiles is 256 characters.

For example :

```
-flagfile "/flagfiles/first /flagfile/second"
```

In addition to the standard flagfile functionality, two additional flagfile parameters are available. These are provided so that if a flagfile does not exist at runtime, the entry can remain in the queue and check again for the flagfile at a later time

-flagfile_attempts 'n'

abbreviations : -ff_attempts, -ffa

Should the flagfile not exist at run time the job will remain in the queue and retry 'n' times before failing.

 **Default :**

'1', i.e. the existence of the flagfile will be checked once.

-flagfile_delay 'nmins'

abbreviations : -ff_delay, -ffd

Used in conjunction with the flagfile attempts option to specify the number of minutes to delay before testing the existence of the flagfile again.

 **Default :**

1 minute.

Startup scripts

-profile

Instruct the BQplus batch job to run the following UNIX startup scripts, before starting the job itself :

```
Bourne and Korn shell : .profile
'C' shell : .login & .cshrc
```

 **Default :**

If '-profile' is not specified the following BQplus startup scripts will be run, if found :

```
Bourne and Korn shell      :      .bqprofile
'C' shell                   :      .bqcshrc
```

Restarting a corrupt job

-restart

If there is a system failure, BQplus jobs that were executing will still be marked in the queue with an 'exec' executing state. When the queue is fixed (see page 58) this inaccuracy will be corrected by changing the state to 'new'..

Default :

Incorrectly marked jobs will be changed to have a state of 'error' and a comment applied to the job indicating that it had previously been incorrectly marked as executing.

Prioritisation of a job

-priority 'priority_level'

abbreviations : -pri

Using this option will place an entry in a BQplus queue with an associated job priority. When the controller polls the queue the jobs with the highest priority will be run first.

Priority level	Description
0	Lowest priority will be given
1	Standard priority
2	Highest priority

Table 8 : Job priority levels

Default :

A new job is added with a priority of 1.


Note :

Because job priorities involve heavy queue usage, the BQplus administrator may have disabled this feature.

Silent running

-quiet


Request that no BQplus banners or messages are displayed when a command is issued.

 *Default :*
Banners and messages will be displayed.

Holding a job

-hold

Using this option will place add an entry to the BQplus queue and place it in a 'held' state. When a job is held it will not be started by the BQplus controller.

 *Default :*
A new job is added to the queue in a 'new' state.


Reholding a job



Tip :
Use this option to control a recurring job (eg. a job on a schedule) that requires a manual start procedure for each execution.

-rehold


Using this option will cause the job to be placed in a 'held' state, *after* it has been executed by BQplus.

 *Default :*
A new job will run once and be deleted from the queue.

Notification

-mail

Request that a message is sent when the job has completed. Typically this will be via the UNIX mail system, but the Systems Administrator can configure alternative methods if required (See section starting on page 63)

 *Default :*
No notification will be sent.


Job Comments



Tip :
Use job comments to provide more full information than can be supplied using the 32 characters allowed in a job reference; this might include restart procedures, contact names etc..

-comments 'comments_file'

Request that a file of job comments is attached to a job. This will be displayed when a full queue listing of the job is requested by supplying the job entry number. Alternatively it can be viewed through the 'Q+' facility in the 'bqmenu' ASCII menu system.

 *Default :*
No comments file will be applied.

Saving failed jobs



Tip :
Use this option while testing any new jobs to be run by BQplus. Should the job fail you can retest using the '-retry' option, rather than generating another BQplus job.

-retain

abbreviations : -ret

Request that should the job fail, the entry will remain in the queue in an 'error' state.

Default :

Failed BQplus jobs will be deleted from the queue.

```
# batchq -list -full 15
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

Queue      Req.   Time  User   Program                               State
-----
immediate  -----
16 Jun 2002
          15    11:00 root   /scripts/backup.sh                   error
          -> bourne shell, logfile /bq15.log, runtime_dir /scripts, retain
          n on error, previous #1
          -> (Last executed : 16 Jun @ 11:00:20)
          i> Job failed with exit code of 1
```

Figure 9 : A retained job



Note :

If a job is running on a schedule, the use of this option is not supported.

Automatic retrying of failed jobs

-error_attempts 'retries'

abbreviations : -ea

Request that if a job fails, the BQplus Controller will automatically put it back in the queue for retrying. The job will be set to have a start time of the previous start time + a pre-defined period. (see below)



Default :

No automatic retry will occur.

-error_delay 'minutes'

abbreviations : -ed

Used in conjunction with the '-error_attempts' option, this value will be used to determine the new start time of a failed job that is being automatically retried.




Default :


1 minute.

Taking action on a job failure


-action_on_error 'error_action'
abbreviations : -aoe

Request that a command is executed should a job complete with an exit status other than zero. The 'error_action' is an action that is defined in the configuration file '\$BQ_HOME/config/actions.cfg'.

 **Default :**
 No action on error will be applied.


 **Note :**
 Refer to 'Action files' on page 53 for an alternative method of taking action dependent on a jobs exit code.

Disabling filename validation on submission

 **Tip :**
 Use this option to submit a entry that will run a program that is created as part of a processing cycle.

-no_validate
abbreviations : -nv


Instruct BQplus to not validate the filename that has been supplied to the submission command.


 **Default :**
 Filename validation will be performed.

Running a job as someone else

-as 'UNIX identifier'

Instruct BQplus to run the job using the UNIX identifier specified. This feature is only available to BQplus Operators and Administrators. Extra security may be applied to sensitive identifiers by defining them in the security configuration file 'secure.cfg'. Once placed in the security configuration file the identifier may only be used with the '-as' option by BQplus administrators.

 **Default :**
 The job will be run using the UNIX identifier of the submitter.

 **Note :**
 Using this option in conjunction with the '-envs' option to save environment variables should be done with care. The job would run using the environment variables for the user that submitted the job, while running the job as someone else.

Modifying a request

Having submitted a job to BQplus the user may choose to change parameters associated with that job through the use of the modify option.

A BQplus request can only be modified while it is not executing; executing jobs are shown in the queue with a state of 'exec'.

To modify a job the BQplus entry number is used in conjunction with the '-mod' option. Options that are required to be changed are also provided as necessary.

```
$ batchq -modify entry1 entry2 .....  
abbreviations : -mod
```

Where 'entryn' is the BQplus entry number of the job to be modified.



Note :

From the command line it is not possible to partially modify an option, you must specify an entire new argument with the '-mod' parameter. The use of the menus however does allow for the editing of arguments associated with an option.

You may modify more than one job from the same command line.

Turning off an option

To turn off an option that has been specified for a BQplus entry, you should first identify whether the option is specified with or without a parameter.

Options not requiring a parameter can be turned off by inserting `'no_'` in front of the option itself. For example to turn off job notification :

... -no_mail ...

Options that fall into this category are as follows :

- mail
- retain
- profile
- restart
- envs
- rehold
- append
- bank_holiday

```
# batchq -mod 10 -no_mail
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Entry 10 modified
```

Figure 10 : Turning off an option not requiring a parameter

Options requiring a parameter can be turned off by respecifying the option with an argument of `'-none-'`.

For example to turn off any arguments that may have been specified use the following option :

... -args -none- ...

Options that fall into this category are :

- as
- schedule
- wait_for
- wait_for_reference
- arguments
- comments
- flagfile

```
# batchq -mod 10 -wait_for_ref -none-
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Entry 10 modified
```

Figure 11 : Turning off an option requiring a parameter

More options

Unholding a job

-unhold

BQplus jobs that are held will be shown in the queue with a 'held' state. Modifying the entry using the '-unhold' option will set the state to 'new' and allow the BQplus controller to start the job.



Default :

Failed jobs will remain in the BQplus queue until retried or deleted.

Retrying a job

-retry

Failed BQplus jobs will be shown in the queue with an 'error' state. Modifying the entry using the '-retry' option will reset the state to 'new' and allow the BQplus controller to restart the job.



Default :

Failed jobs that have been submitted using the '-retain' option will remain in the BQplus queue until retried or deleted.

Deleting a request

To delete a request from a BQplus queue the following command should be used :

```
$ batchq -delete entry1 entry2 ...  
abbreviations : -del
```

Where 'entryn' is the BQplus entry number of the request to be deleted.

```
# batchq -del 10  
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]  
[Nodename : solaris] (/usr/bq+)  
Request 10 deleted
```

Figure 12 : Deleting a job

If the BQplus request is executing at the time of the delete request the job will be instructed to abort. This is accomplished by signaling the job with UNIX signals that have been defined by the BQplus administrator. Aborted jobs will contain information to this effect in the log file.

```
JOB #16 /scripts/backup.sh started 16/06/02.11:04.02  
Welcome to Solaris 8  
Job command : /scripts/backup.sh  
JOB #16 /scripts/backup.sh aborted 16/06/02.11:04.07
```

Figure 13 : An aborted job log file

Listing the queue

In order to monitor and effectively control BQplus jobs, a variety of queue listing facilities are provided.

Basic listing

The format of the command to request a basic queue listing is as follows :

\$ batchq -list

When issued a display of jobs currently found in the BQplus queues are displayed. A number of characteristics associated with each job will be shown including the current job state.

```
# batchq -list
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

Queue      Req.   Time  User      Program                               State
-----
immediate
16 Jun 2002
          17    11:08 root      /usr/bin/date                         held
over
-----
16 Jun 2002
          16    11:08 root      /scripts/backup.sh                    time
```

Figure 14 : Batch queue listing



Note :

The `*' indicates that the job was submitted by another user, using the `'-as' option. (page 30)

BQplus job states

A variety of BQplus states can be associated with a BQplus job, and can be seen when a queue listing is requested. A table of job states can be found in Table 16 : Job states on page 94.

Full listing

To obtain a comprehensive listing of the queue the following command should be used :

\$ batchq -list -full

```
# batchq -list -full
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

Queue      Req.   Time  User   Program                               State
-----
immediate  -----
16 Jun 2002
      17    11:08 root   /usr/bin/date                         held
      -> bourne shell, runtime_dir /scripts

over       -----
16 Jun 2002
      16    11:08 root   /scripts/backup.sh                   time
      -> bourne shell, runtime_dir /scripts
      -> Earliest start : 01 Jul @ 00:00:00
```

Figure 15 : Detailed batch queue listing

Selective listing

A variety of options are provided to allow many different views of the BQplus queue.

- Entry Number
- Queue Name
- State
- Suite
- Start date/time
- Reference
- Schedule

Entry number

To display a listing of a particular BQplus entry the following command should be used :

\$ batchq -list 'entry_number'

```
# batchq -list 17
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

Queue      Req.   Time  User   Program                               State
-----
immediate
16 Jun 2002
           17    11:08 root   /usr/bin/date                         held
```

Figure 16 : Listing a particular entry number

Queue name

To display a listing of a specific BQplus queue the following command should be used :

\$ batchq -list -queue 'queue_name'

```
# batchq -list -queue over
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Warning : BQ+ queue over is not currently active!

Queue      Req.   Time  User   Program                               State
-----
over
16 Jun 2002
           16    11:08 root   /scripts/backup.sh                    time
```

Figure 17 : Listing a particular queue

State

To display entries found with a specific job state the following command should be used :

`$ batchq -list -state 'job_state'`

```
# batchq -list -state held
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

Queue      Req.   Time  User      Program                               State
-----
immediate  -----
16 Jun 2002
           17     11:08 root      /usr/bin/date                         held
```

Figure 18 : Listing all jobs with a particular state

For a list of job states please refer to page 94.

Start date/time

To display entries that are scheduled to run on a specific date the following command should be used :

`$ batchq -list -start_time 'date'`

For valid 'date' format please refer to page95.

Reference

To display a listing of all entries with a job reference the following command should be used :

`$ batchq -list -reference 'job_reference'`

```
# batchq -list -full -ref "test sleep job"
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

Queue      Req.   Time  User      Program                               State
-----
immediate
16 Jun 2002
          18    11:12 root      /usr/bin/sleep                       held
Arguments -> 60
r> test sleep job
-> bourne shell, runtime_dir /scripts, no log
```

Figure 19 : Listing jobs with a particular reference



Note :

you may specify a partial job reference to display a listing of all jobs with references containing the sub-reference.

```
# batchq -list -full -ref "test sleep"
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

Queue      Req.   Time  User      Program                               State
-----
immediate
16 Jun 2002
          18    11:12 root      /usr/bin/sleep                       held
Arguments -> 60
r> test sleep job
-> bourne shell, runtime_dir /scripts, no log

over
16 Jun 2002
          21    11:15 root      /usr/bin/sleep                       held
Arguments -> 60
r> test sleep second job
-> bourne shell, runtime_dir /scripts, no log
```

Figure 20 : Listing jobs with a particular sub-reference

Schedule

To display entries that running on a particular schedule the following command should be used :

```
$ batchq -list -schedule 'schedule_name'
```

Combining listing options

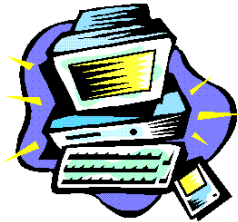
Queue listing options may be combined to provide a higher level of selection.

```
# batchq -list -full -ref "test sleep" -queue over
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Warning : BQ+ queue over is not currently active!

Queue      Req.   Time  User   Program                               State
over
-----
16 Jun 2002
      21    11:15 root   /usr/bin/sleep                       held
Arguments -> 60
r> test sleep second job
-> bourne shell, runtime_dir /scripts, no log
```

Figure 21 : Combining queue listing options

Chapter 4 : Advanced User Guide



This section contains the information which the advanced BQplus user may require.

Job History

In order to allow the easy checking of BQplus job activity a history file is created in the users home directory and is called ``.bq_history``. This ASCII file consists of a line of text per job and includes information such as the logfile name and exit code. This file is used by the menu systems in order to allow the examination of log files.

As a line per job is written to the BQplus history file, it can become quite large. In order to purge the history file the following command can be used :

`$ batchq -purge `ndays``

Where ``.ndays`` is the number of days of history that is to be kept.



Note :

To delete all the history ``.ndays`` should be specified as ``.asap``.

Environment variables

All BQplus options may be controlled through the use of environment variables. This avoids the user having to respecify an option or value that is always required; in addition the user may customise the environment for personal use, rather than having to request a change to the global configuration files.

By default, environment variables will only be consulted when a new job is being submitted to the queue. To enable consultation during a modify command the option **'-use_envs'** should be used.



Note :

Only the job submission screens in the menu systems will reflect environment variables, 'use_envs' should be used with care from the modify screen.



Note :

Should an environment variable exist to both turn off and turn on an option, the option will be turned on.

The two types of BQplus option require a slightly different use of environment variables in order to be controlled.

Options requiring a parameter

Any BQplus option requiring a parameter should be configured as follows :

```
BQ_option=value  
export BQ_option
```

Where 'option' is the actual BQplus option requiring to be controlled, specified in UPPERCASE.

For example to force all BQplus jobs entered by a particular user to be submitted to the 'over' queue, the following commands could be used :

```
$ BQ_QUEUE=over  
$ export BQ_QUEUE
```

Options not requiring a parameter

A BQplus option that does not require a parameter can either be turned on or turned off.

Turning on an option

To turn on an option not requiring a parameter, the following command format should be used :

```
BQ_option=Y
export BQ_option
```

Where 'option' is the actual BQplus option requiring to be controlled, specified in UPPERCASE.

Turning off an option

To turn off an option not requiring a parameter, the following command format should be used :

```
BQ_NO_option=Y
export BQ_NO_option
```

Where 'option' is the actual BQplus option requiring to be controlled, specified in UPPERCASE.

For example to force all BQplus jobs entered by a particular user to be submitted with a request for job completion notification :

```
$ BQ_MAIL=Y
$ export BQ_MAIL
```

Detecting that a program is running in BQplus



Tip :

Test for this variable in startup scripts before executing programs not required for batch jobs.

```
if [ -z "$BATCH_JOB" ]
then
```

```
fi    ...
```

An environment variable 'BATCH_JOB' is set for all BQplus jobs. The value associated with this variable is the BQplus request number.

Adding a request

This section continues to detail options that may be used when placing a job in the BQplus queue for execution using the 'batchq' command.


Advanced options


Scheduling a job.

-schedule 'schedule name'
abbreviations : -sched

Request that the job uses either an incremental or list schedule to determined run times.

Two types of BQplus schedule are available for use - "incremental" and "list".

 **Default :**
A job will not run on a schedule.

 **Note :**
If a start time is specified in addition to a schedule, the job will run at the start time and will apply the schedule subsequently.

If scheduled run times are missed due to machine failure etc., the job will run *once* and the schedule reapplied when the job completes.

Incremental schedules

An incremental schedule offers the simplest form of job scheduling. A job frequency is defined in a number of minutes and following each execution of a job it will be rescheduled to run after that period of time. For details on how to maintain BQplus schedules please refer to page 76.

List schedules

A list schedule offers a great deal of flexibility and is the core method of scheduling used by BQplus. A list of dates and/or times is created in an ASCII file and is used as the source of run times for a job submitted on that schedule. When the list of dates and/or times is exhausted the job will be deleted from the queue. Dates and times defined in the list schedule can following any of the permitted BQplus formats as defined in Table 17 : Date/time syntax. For details on how to maintain BQplus schedules please refer to page 76.

Preventing a job from running on a bank holiday

-no_bank_holiday

abbreviations : -no_bhol

Prevent a scheduled job from running on any date that is defined in the public holidays file - '\$BQ_HOME/config/bank.hols'.

This option is provided for use of jobs running on a schedule.

If the job is not required to run on a public holiday then the 'no bank holiday' option would force the job not to run on a public holiday but perform one of three actions :

- Run on the next date/time in the schedule list.
- Run on the next day if submitted with an additional argument of '**next_day**'.
- Run on the previous day if submitted with an additional argument of '**previous_day**'.

Previous day and next day options.

The use of previous day and next day options is complicated by days such as Monday and Friday, where the job might not be required to run on a weekend. To discount weekends in the calculation of run time the 'saturday_working' and 'sunday_working' switches should not be enabled in the 'defaults.cfg' default configuration file.

Default :

A job will run on the next scheduled working day in a schedule definition list.

Saving the BQplus request number

-save_job

Instruct BQplus to save the request number of a newly submitted batch job into shared memory for future reference.⁴

Default :

The request number is not saved.

Retrieving the BQplus request number

-last_job

Instruct BQplus to fetch the last request number that has been saved by the use of the '-save_job' option. The command will return a request number without a

⁴ Request numbers are saved using the UID of the user as the reference key.

line feed, which can be placed into a shell variable using a technique similar to that shown below. Should no request number exist in memory for retrieval then a return value of '-1' will occur.



Note :

This and the '-save_job' option are provided for automatic submission of interdependent jobs. Before using the job numbers to build interdependency, you should refer to page 25 which may provide an easier way of achieving the same result.

```
$ LAST_REQUEST_NUMBER= ``$BQ_HOME/bin/batchq -last_job`
```

Using environment variables in a modify operation

-use_envs

Instruct BQplus to consult the environment variables in addition to any options that have been supplied via the command line.



Note :

This should be used with care as incorrect environment variables will cause an apparently correct command line to fail.

Command line options will always override environment variables. For further information please refer to 'Environment variables' on page 42.

Updating job dependencies

-clear_wait_for 'entry_no'
abbreviations : -cwf

-clear_wait_for_ref 'reference'
abbreviations : -cwfr

Instruct BQplus to consult the queue and clear any dependencies based on either an entry number or job reference..

Job Suites

Job suites are provided to allow the user to group a set of BQplus jobs together. Jobs in a job suite may be found across many queues and may consist of as many jobs as necessary. A job may only be a part of one suite.

When BQplus is first installed, job suites are disabled. They can be enabled by the BQplus administrator by updating the defaults configuration file. (see section starting on page 63)

Submitting a job to a suite

To submit a job to a suite the following 'batchq' option should be used :

-suite 'job_suite'

Where 'job_suite' is a valid suite that has been defined in the configuration file 'suites.cfg;.



Default :

A job will not be placed in a suite.

Listing jobs in a suite

When a job has been submitted to a suite, it will be shown when a full queue listing is requested. The detail of the particular job will include detail such as 'suite suite_name'.

In addition the user may request a list of any jobs that have been placed in a particular suite, as shown below.

\$ batchq -list -suite 'job_suite'

Where 'job_suite' is a valid suite that has been defined in the configuration file 'suites.cfg;.

Suite definition

Job suites are defined in the configuration file 'suites.cfg'. The format of a suite definition is similar to that of a queue and looks like the following :

```
BEGIN 'suite_name'
directive
...
END 'suite_name'
```

Where suite name is a unique name of up to 16 characters.

Suite parameters

Within the definition of a suite a range of parameters may be supplied for extra functionality.

Make a suite wait for another suite to complete

WAIT_FOR 'suite_name'

The 'wait_for' parameter instructs that should any jobs exist in the suite that is supplied with the wait_for parameter, no jobs will be started.

For example :

```
BEGIN suite1
END suite1
BEGIN suite2
    WAIT_FOR suite1
END suite2
```

Where 'suite2' jobs will not be started until all 'suite1' jobs have completed and been deleted from the queue.



Note :

A suite may 'wait for' any number of other suites.

Suite interdependency is not compatible with jobs that are running on a schedule.

Argument validation

argument_parser 'shell_script'

Instructs that when a job is started all arguments that have been supplied to the job should be checked by the named shell script. If the argument validation script returns an error exit status the validation will be considered to have failed and the job will fail.

Sample shell scripts are supplied with BQplus and follow a simple concept.

Arguments are supplied to the shell script in traditional '\$1, \$2 ...' format. The shell script can perform whatever tests are appropriate on these arguments and should exit with a status of '0' or '1' depending on success or failure respectively.

The script can of course perform any function, which may not involve the use of arguments but the primary function is intended for argument parsing.



Note :

When a job is submitted or modified using the BQplus menus, argument validation will be applied on-the-fly if a schedule has been specified, and that schedule has argument validation defined. If the argument validation fails the user will be prompted whether the job should continue to be submitted or whether the form should be redisplayed for modification.

Suite status

The user may obtain a listing of suite status at any time, by using the following command :

\$ *bcon -status*

In addition to the normal status of the BQplus controllers and queues, a section entitled 'BQplus suites' will now be displayed.



Note :

The 'BQplus suite' section will only be shown if jobs are found in the queue and have been placed in suites.

Possible states of BQplus suites are 'READY' and 'WAITING'. Where 'WAITING' indicates that a suite is waiting for another suite to complete, having been defined with the 'wait_for' directive. 'READY' is used to indicate that jobs in the suite can be started, subject to queue and actual job requirements (eg. start time, dependencies etc.) being met.

In addition to the state being shown, the number of jobs in a particular suite is shown in brackets at the end of the suite status line. This figure includes both executing and waiting jobs.

To list suite dependency and argument parsing the following command should be used :

\$ *bcon -list -suites*

Using suites from within the BQplus menus

Tight integration of suites has been provided for the ASCII menu system - 'bqmenu'.

Argument validation

Argument validation is supported from within the 'Submit', 'Modify' and 'Submit Like' forms. When the job has been placed in a suite, any argument validation defined for that suite will be invoked when the form is executed. Should the validation fail, the user will be given the option to return to the form for alteration, or to continue regardless. If the latter option is chosen and not further job modification takes place, the job will fail at run time.

Suite scripts

Suite scripts are the means by which it is possible to pre-define a number of menu forms and parameters through which a user invoking the script will be guided. When all the forms have been completed the user will be asked whether to go ahead and perform all the completed forms at once.

Script parameters are as follows :

FORM Number of the bqmenu form to display. The form number should be supplied as a argument to this parameter.

INIT Supply values to be placed in the form fields when the form is activated. Use quotes or double quotes to supply values containing space characters.



Note :

To leave a field unchanged and still modify other fields beyond it in the form, a value of '-skip-' should be used.

DISABLE Allows fields to be disabled, preventing user change. Supply an argument of fields to be disabled.

TITLE Allows the title to be displayed at the top of the menu form to be changed from the default.

ARGUMENT_VALIDATION

Allows argument validation to be enforced for a menu form being driven from the 'Submit' or 'Modify' job screens.

INSTANT Does not allow the user to change the contents of the form.

Suite name

In addition to the suite script parameters shown above, the following two parameters can be defined once in each suite script.

script_name ***'name'***
exclusive_suite

These two parameters when used in conjunction prevent the suite script from being invoked if a job from the suite has already been placed, or remains in the BQplus queue.

A sample script might be as follows :

```
# This is a sample suite submission script
#
suite_name overnight          # You must name the suite if you want
                              # to check if there
                              # are any jobs already in the queue at
                              # submission time
exclusive_suite              # only run if no jobs in the queue in
                              # this suite (suite
                              # name must be provided if this is
                              # required)

#
# form numbers
# 1 :   submit a job
# 2 :   modify a job
# ? :   If you need to access any of the other bqmenu forms please
#       contact your software supplier for the appropriate form number.
#       Alternatively ask us via 'support@corp.powernet.co.uk'
#
begin job                    # Name ('job') must be unique
# Select form - in this case 'Submit a job' : form 1
    form    1
# Arguments to be used. Note : the " or ' marks to encapsulate spaces
    init    /backup.sh "monday full" over 20:00
# Change the title of the form
    title   Submit a full backup
# Disable fields 1 & 4. Note : fields start at 0, not 1.
    disable 1 4
end job

begin job2                   # Name ('job2') must be unique
    form    1
    init    /index.sh -none- over 06:00
end job2
```

The script should be saved in an area where the user has read access and can be called any name. The user would run the 'Run a Suite Script' form and type the full pathname of the script to be run.



Important :

If supplied, the suite name in the 'Submit a Job' screen will be defaulted to the name of the suite as defined by the suite script. (see next section).



Note :

A sample suite script is provided in the BQplus configuration directory.

BQplus IPC (inter-process communication)

BQplus IPC is the means by which the BQplus user can communicate with a BQplus job that is executing. This is achieved by developing the BQplus job script to request

information from a particular terminal, when assistance is required. For example, if a backup job completes the backup of a system, it could ask a terminal whether an index of the tape was required.

This functionality is achieved by the use of the following command :

```
$ batchq -ask 'terminal' -ipc 'IPC_number' "Is an index required?"
```

Following the execution of this command a message is sent to 'terminal' and the user can respond using the following command :

```
$ batchq -answer 'IPC_number' "yes"  
abbr : -reply
```



Note :

The 'IPC_number' is a unique number assigned to the 'conversation' and is identified to the user in the message that has been sent to the terminal.



Note :

The message will be sent to the terminal at intervals of every minute. The default number of attempts is 5, but can be overridden by the use of the '**retries**' option followed by a number greater than 0. When a question times-out the following message will be returned into the script variable : "No reply to BQplus IPC received!". The script should check for this value and take whatever action is appropriate.



Note :

If the argument contains any spaces or reserved characters then it must be protected by quotation marks.

A sample script for the BQplus job could be as follows :

```
# Sample BQplus job script that uses BQplus IPC
#
# Note : in this example the BQplus job number is used with
# the '-ipc' option to uniquely identify the ipc request.

# DO BACKUP HERE

CONTINUE=`batchq -ask /dev/pts/3 -ipc $BATCH_JOB "Finished backup, \
shall I index the tape?"`
FINISHED=FALSE
while [ "$FINISHED" != "TRUE" ]
do
    case "$CONTINUE" in
        y) FINISHED=TRUE ; DO_SLEEP=TRUE ;;
        n) FINISHED=TRUE ; DO_SLEEP=FALSE ;;
        *) CONTINUE=`batchq -ask /dev/pts/3 -ipc $BATCH_JOB \
"Valid response is 'y' or 'n', shall I index the tape?"` ;;
    esac
done

if [ "$CONTINUE" = "No reply to BQplus IPC received!" ]
then
    echo No response was received from the terminal.
```

```

        DO_SLEEP=FALSE
    fi
    if [ "$DO_SLEEP" = "TRUE" ]
    then
        echo Backup index will be performed
        # DO INDEX HERE
    else
        echo No backup index will be performed
    fi
    exit 0

```

A sample session using the script shown, might be as follows :

```

[BQplus 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
09:22 : You have a BQplus IPC "Finished backup, shall I index the tape?"
Please reply using 'batchq -answer 39', followed by your answer.

$ batchq -answer 39 no
[BQplus 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
Reply sent.

[BQplus 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
09:22 : You have a BQplus IPC "Valid response is 'y' or 'n', shall I index the tape?"
Please reply using 'batchq -answer 39', followed by your answer.

```



Note :

BQplus IPC uses 'FIFOs' (named pipes) as the means by which information is exchanged. These are created in the BQplus 'tmp' directory and are self-maintaining.

Action files

In addition to the 'action_on_error' parameter, action files provide the means to take an action dependent on the exit status of a job.

The 'action files' have a suffix of '.act' and are located in the BQplus configuration directory.

```

batchq 'jobname' -action_file 'action_filename'
abbr : -af

```

Where 'action_filename' is an ASCII file of exit codes and associated actions located in the BQplus configuration directory. Exit codes and their associated actions are defined on the same line and are separated by a ':' character.

For example :

```

# Action definition file : overnight.act
0:/bin/echo "The job has completed successfully"

```

```
1,2:/bin/echo "The job has failed"  
*/bin/echo "The job has exited with an unknown error code"
```

**Note :**

The use of the `*` character to indicate an action that should be performed if a matching exit status is not found in the action file.

Many exit codes can be defined on the same line by separating them by a `,' character.

Administration of the action files themselves is not possible through the menu system, and must typically be performed from an editor such as `vi`.

Remote BQplus working

-host 'hostname'

Instruct that the BQplus command should be executed on a remote machine.

In order for a remote command to be successful the BQplus Netserver must be started on the host on which the command is required to be executed. In addition the username must be authorised as a legitimate remote user in the configuration file `remote.cfg`.

Chapter 5 : Administrator Guide



This section contains the information which the BQplus Systems Administrator requires to get the job scheduler up and running and configured to a sites requirements. Examples in this section use the command line interface, for details on the two menu systems please refer to the 'Menu Systems' chapter starting on page 81.

Overview

BQplus is a job scheduling system that consists of one or many job queues. Each queue has individual characteristics such as limits on how many jobs can be started and what time of day this is permitted. When a job is submitted to BQplus it is placed in one of these queues. Once submitted the job will be started by the BQplus controller at the appropriate time. The BQplus controller is the single most important process in the BQplus system. If the controller is not started, no BQplus jobs will be run.

The BQplus Controller

Starting the Controller

To start the BQplus controller the following command should be used :

\$ bcon -start

Once started the BQplus controller will service the BQplus queue until requested to stop or is terminated.

```
# bcon -start
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
BQ+ controller started
```

Figure 22 : Starting the controller

Stopping the Controller

To stop the BQplus controller the following command should be used :

```
$ bcon -stop
```

Once stopped, no BQplus job will be started. Note that BQplus jobs that are already started will continue to run through to completion.

```
# bcon -stop
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)
BQ+ controller stopped
```

Figure 23 : Stopping the controller

Checking the Controller status

To check the status of the BQplus controller, the following command should be used:

```
$ bcon -status
```

If the controller is running the UNIX PID will be displayed by the 'STARTED' message, which may be useful to the Systems Administrator.

```
# bcon -status
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

BQ+ Servers
-----
controller   STARTED  (1411)
netserver    STOPPED

BQ+ Queues
-----
immediate    STARTED
over         OUT OF ACTIVE FRAME
reports      STARTED
backups      OUT OF ACTIVE FRAME
```

Figure 24 : Displaying the controller status

Checking the Controller log file

To check the BQplus controller log file, the following command should be used :

\$ bcon -log

```
# bcon -log
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
2101 BUILT ON Sun Sep 15 11:10:06 BST 2000 5.6 Generic_105181-29 sun4u

Visit BQ+ on the World Wide Web at 'www.grepite.com' and learn about
upcoming features and great new products from Corporate Practical Solutions.

Running on host solaris
BQ+ evaluation period will end on 16/07/02.11:07.31 (0)
BQ+ controller started at 16/06/02.11:19.18
Running in timezone GB

Reset rate is set every 24 hours
Poll rate is set to every 60 seconds
Quota poll rate is set to every 60 seconds
Logfile is /usr/bq+/logs/controller.log
Purging accounts files older than 31 days
Controller wake up signal is set to 20.
Suites are disabled.
Home grep command is set to : `/bin/grep "^%bq_as_user%:" /etc/passwd | /bin/awk
-F: '{print $6;exit}'`
```

Figure 25 : Displaying the controller logfile

Signaling the Controller

To signal the BQplus controller to poll the queue the following command should be used :

\$ bcon -jump_start *abbreviations : -jump*

The BQplus controller polls the queue at a pre-defined rate (see page 71). Should you want to wake up the controller and request an immediate poll of the queue you should use this command.

BQplus Queues

Pausing a queue

To pause a BQplus queue the following command should be used :

```
$ bcon -pause 'queue_name'
```

Once paused, no new BQplus jobs will be started from the queue.

Unpausing a queue

To unpause a BQplus queue the following command should be used :

```
$ bcon -unpause 'queue_name'
```

Once unpaused, BQplus jobs will be started from the queue.

Listing the queues

To display a listing of the BQplus queues the following command should be used :

```
$ bcon -list
```

Queue maintenance

The BQplus queue is self maintaining and should not normally require any manual intervention in its operation. An option is however provided to allow the BQplus administrator to fix the queue should it become necessary.



Note :

The controller and all jobs must be stopped before a queue fix is possible.

Fixing the queue



Tip :

Place a fix command in the BQplus startup script to fix the queue when a machine is rebooted.

```
$ batchq -fix
```

This command performs the following :

- *Correct any invalid entry states to be accurate*
- *Delete invalid entries*
- *Compress queue to a minimum size*
- *Order queue entries for maximum efficiency*



Note :

For the duration of the fix operation BQplus will be unavailable to users and should therefore only be used when absolutely necessary or out of normal processing hours.

In addition to the regular fix of the BQplus queue, it is possible to fix the queue and renumber all the entries to be starting from 1. This will be particularly useful for a site which has been running BQplus for some time and has entry numbers in the hundreds of thousands.

\$ batchq -fix -renumber



Note :

Any job dependencies using entry numbers will be adjusted by this process, and will therefore remain valid.

The BQplus netserver

Having installed BQplus on to more than one machine, it is possible to Administer the queues centrally from one of the BQplus machines.

The netserver is the process by which other UNIX machines can communicate with the BQplus system. Once started it will listen on a pre-defined TCP/IP port and service incoming remote BQplus requests.

If the netserver is not started no remote machines will be able to communicate with the BQplus system.

Security of the remote BQplus facilities is controlled through the 'remote' configuration file. (see page 78)



Note :

Although it is possible to centrally control BQplus, each queue is a separate entity and operations such as moving jobs between machines are not supported.

Starting the netserver

To start the BQplus netserver the following command should be used :

\$ bcon -start netserver

Once started the BQplus netserver will service remote BQplus requests.

Stopping the netserver

To start the BQplus netserver the following command should be used :

\$ bcon -stop netserver

Once stopped, no remote BQplus operation into the host will be possible.

Checking the netserver status

To check the status of the BQplus netserver, the following command should be used:

```
$ bcon -status
```

If the netserver is running the UNIX PID will be displayed in brackets by the 'STARTED' message, which may be useful to the Systems Administrator.

```
# bcon -status
[BQ+ 5.0 Copyright (c) 1993-2000, Corporate Practical Solutions.]
[Nodename : solaris] (/usr/bq+)

BQ+ Servers
-----
controller      STARTED (1411)
netserver       STARTED (1436)

BQ+ Queues
-----
immediate       STARTED
over            OUT OF ACTIVE FRAME
reports         STARTED
backups        OUT OF ACTIVE FRAME
```

Figure 26 : Displaying the netserver status

Checking the netserver log file

To check the BQplus netserver log file, the following command should be used :

```
$ bcon -log netserver
```

Auditing the netserver activity

To enable auditing of all requests received by the netserver, the 'remote_audit' switch should be enabled in the configuration file 'defaults.cfg'. This is more fully described in the 'Configuration files' section on page 63.

Chapter 6 : Advanced Administrator Guide



Configuration files

Configuration files are ASCII files that can be found in the BQplus 'config' directory. The majority of these can be maintained through the BQplus menu systems, but may also be manually configured using an editor such as 'vi'.

Actions on error : 'actions.cfg'

Description

Consists of any number of actions that can be used by jobs requiring action to be taken when completion is with a non zero exit status. The user can access the actions through use of the '-action_on_error' option.

Format

action_name 'action command'



Tip :

The 'action command' can include BQplus keywords.

Where the 'action_name' is a single word, up to 16 characters in length, and the action command is the actual command that is required to be run, should the job fail..

Example

***message_console /bin/echo BQplus job %BQ_JOBNO% has failed!
>/dev/console***

BQplus defaults : 'defaults.cfg'

Description

Contains a number of switches that allow the BQplus administrator to tailor the BQplus environment to suit the requirements of the site.

Switches

default_shell ***'shell'***

Instruct BQplus to use a specific default shell, when a user has not specified a shell using the `'-shell'` option of the `'batchq'` command. `'shell'` may be one of the following :

- ***sh*** (Borne)
- ***ksh*** (Korn)
- ***cs*** ('C' shell)

home_command ***'command'***

Sets the command to determine the user home directory that will be embedded in the BQplus job script that will be used to run the job.

Two shell scripts are provided with the BQplus distribution : `'passwd_lookup.sh'` and `'nis_lookup.sh'`. The first of which emulates the default operation of a `'passwd grep'` for the sixth field of a password file entry. The second script uses a `'nisgrep'` to obtain information about the userid that will be used to run the job. Both programs require an argument of the username to be used and have sample entries defined in the `'defaults.cfg'` configuration file. An example for `'passwd_lookup.sh'` would look something like :

```
home_command ` %bq_home%/config/passwd_lookup
%bq_as_user%`
```

default_queue ***'queue_name'***

Instruct BQplus to place any jobs not submitted with the `'-queue'` option into the defined default queue. This queue must be a valid BQplus queue that has been defined in the `'$BQ_HOME/config/queues.cfg'` configuration file.

display_args

Instruct BQplus to display any job arguments when a brief queue listing is requested.

display_reference

Instruct BQplus to display any job reference when a brief queue listing is requested.

display_schedule

Display the job schedule when a brief queue listing has been requested.

single_line_filename

Tip :
to limit a queue listing to a single line per entry the `'display_args'`, `'display_reference'` and `'display_schedule'` switches should not be used, while the `'single_line_filenames'` should be enabled

Limit the display of an entry's filename to 1 line. Overflow will be indicated at the start of the line with `'...'`

read_and_execute

Tip :
Useful if you are submitting a lot of shell scripts to BQplus - shell scripts require `'r+x'` to run.

Instruct that when a job is submitted to the queue, BQplus will check that the user has read and execute access to the program file. If not enabled BQplus will check for execute access only.

security_level ***'0', '1' or '2'***

Restrict the level of access for all users except BQplus administrators and operators.

- ***0 : no restriction***
- ***1 : list access only***
- ***2 : no access***

print_command ***'print_command'***

Allows the user to print the output that has been produced by a bqmenu action through use of the `'p'` key when prompted.



Note :

The `%bq_print%` keyword will be substituted with the name of the temporary file that has been created by the bqmenu, and should be included in the print command where appropriate.

For example :

print_command /usr/pq+/bin/printq %bq_print% -at laser

us_dates

Instruct BQplus to use US style dates. i.e. mm/dd/yy rather than the default UK style dates (dd/mm/yy).



Note :

Configuration files such as 'list' schedules created under one format will not be compatible with systems using the other. Similarly any '.bq_history' files that have been created by a BQplus system using one date format will be considered invalid by any purging or historical-type operations that are undertaken by a system using the other format. It is important therefore that if US dates are to be used, that the configuration directive should be enabled in 'defaults.cfg' as early as possible.

confirm_deletions

Instruct BQplus to ask for confirmation before deleting an item.

quick_menus or instant_menus

Configure BQplus to not require additional presses of the '<return>' key from within the bqmenu menus. eg. Pressing the 'A' key at the initial screen will action the 'Control queue' screen as well as moving the cursor.



Tip :
Saves time!

***console_device* '*console*'**

Configure BQplus to send messages to a console device found at a location other than '/dev/console'

list_full_queue

Instruct BQplus to display all the queue entries to any user with access to list the queue.



Tip :
Use this switch on AIX systems.

no_logname

When running a job, BQplus will normally attempt to reinstate the environment variable 'LOGNAME' to reflect the user who has submitted the job. Certain implementations of UNIX will not permit this as the 'LOGNAME' environment variable is considered read-only.



Tip :
Startup scripts can test for the presence of the environment variable 'BATCH_JOB' to detect whether it is being invoked by a BQplus job.

no_system_scripts

BQplus will normally attempt to run system scripts such as '/etc/profile'. In some cases this may not be required, and using this switch will inhibit this.

remote_audit

Instruct BQplus to maintain an audit log of all activity through the netserver service. This includes all validation attempts as well as actual BQplus commands. The logfile is called 'remote.log' and is maintained in the '/usr/bq+/logs' directory.

no_queue_priorities

Tip :

This will improve queue clearing as the entire queue will not be read before starting a job, which would be the case with queue priorities.

Instruct the BQplus controller to run jobs in the order in which they are found in the queue.

default_log_dir**'directory_name'**

Unless otherwise specified a default logfile is 'bqnn.log' and is created in the home directory of the user. Using this switch it is possible to specify where the 'bqnn.log' file is created.

For example :

To create default logfiles in a directory '/usr/bq_logs'.

```
default_log_dir /usr/bq_logs
```

If this directive is used, care should be taken to ensure that the directory exists and is writable by users of the batch queue.

See 'Log files' on page 21.



Note :

BQplus keywords can be used in the default_log_dir definition.

For example :

To create default logfiles in a directory '/usr/bq_logs/username'.

```
default_log_dir /usr/bq_logs/%bq_as_user%
```

check_queue_state

If the BQplus controller is not started or a queue is inactive a message will be issued when a 'batchq' command is issued.



Tip :
Without this option enabled the select list for logfiles within the menus will not function.

enable_history

Instruct BQplus to write an entry in the history file ``.bq_history`` each time a job completes. The history file is created in the users home directory.

display_run_time

Display the job run time within a brief queue listing.



Tip :
Use this option to reduce the number of hosts displayed when selecting a host to work on from within the menus.

hostfile**'hostfile'**

Instruct BQplus to use a file other than ``/etc/hosts`` as a list of possible other BQplus hosts.



Tip :
Use this option if working in a very busy BQplus environment, when the controller would be continually polling the queue - affecting queue performance to other users.

no_signal

Do not signal the controller to poll the queue when a job completes.



Tip :
Saves time!

quick_select

Requests the automatic display of a select box in certain menu forms such as ``delete an administrator``. This is to reduce the number of keystrokes in these forms.

bank_holiday_working

Instructs BQplus to consider bank holidays as working days. This means that any BQplus date containing the `'work'` keyword will be unaffected by dates defined in the bank holiday file `'bank.hols'`.

default_flagfile_delay**nmins**

Define the default delay in minutes that should elapse between checks for the existence of a flagfile by the BQplus controller. The default value for this

parameter is 1 minute.



Tip :
Specify the mail program as 'write_user' to write a message to the user rather than using the mail system..

mail_program**'program_name'**

Define an alternative program that will be used to send mail when a job that has been submitted with the '-mail' option, completes.



Default :

The default program that will be used is '/bin/mail'.

controller_jump_signal**'UNIX signal name'**

Configure the UNIX signal that will be sent to the BQplus controller, when a 'bcon - jump' is issued.



Default :

SIG_WINCH

**saturday_working &
sunday_working**

Instructs BQplus to consider Saturdays and Sundays as working days. This means that any BQplus date containing the 'work' keyword will be unaffected by the day of the week.

write_reference

Instructs BQplus to display any job reference at the start of the job logfile.

write_command

Instructs BQplus to display the job command at the start of the job logfile.

default_log_umask

Request that the BQplus controller does not set protections on newly created flagfiles, i.e. the default system protections will remain.

enable_suites

Enable the suites facility.

BQplus queues : 'queues.cfg'**Description**

BQplus operates an unlimited number of user-definable queues. Definitions of these queues are located in the 'queues.cfg' configuration file. The basic format of a queue definition is as follows :

```
BEGIN queue_name
    switch
    switch
    ...
END queue_name
```




Note :

The queue name may be up to 16 characters in length and is not case sensitive.

Switches


status **'queue status'**

Indicates the status of the queue.

 *Default :*
STARTED


max_concurrent **'njobs'**

Configure the maximum number of jobs that can be executing in the queue at any given time.

 *Default :*
Unlimited

max_elapsed **'nmins'**

Configure the maximum length of time that any job may take to execute. (in minutes)

 *Default :*
Unlimited

start **'date/time'**

Configure the date/time at which the queue comes active and may start to run jobs.

queue to the controller log file. This information can be used for problem diagnosis and to gain a detailed understanding of what the controller is doing.

 *Default :*

Poll information is not written to the logfile.



Note :

If poll information is written to the logfile of a busy controller, the log may become large.

license_code ***'code'***

The code is a unique string provided by your BQplus supplier and allows BQplus to operate in non-demonstration mode. Please refer to page 14 for more information about licensing the software.

poll_timer ***'nsecs'***

The period of time taken by the controller between polling the queue.



Note :

Try and keep the poll rate at a reasonable period of time (eg. 60 seconds). Reducing the delay to a very small period will cause the controller to stay busy and potentially use CPU time better spent elsewhere!

quota_time ***'nsecs'***

The period of time taken by the controller between checking that a running job has not exceeded the maximum allowed elapsed time, as defined in the queue definition.



Note :

As the maximum elapsed time of a queue is supplied in minutes, a sensible minimum value for this parameter is 60 seconds.

scheduler_priority ***'nice adjust'***

The UNIX scheduler adjustment for the priority of the BQplus controller. The value is exactly equivalent to the nice adjust value as shown in the 'nice' man pages.

 *Default :*

No adjustment to the priority of the process is made.

kill_sequence ***'signal1 signal2 ...'***

The set of UNIX signals that are to be sent to an executing job in order to abort. The signals may be supplied as the signal name or the equivalent signal number.

**Default :**

A job will be ultimately sent a SIGKILL after any signals supplied using this switch have failed to terminate the job.

kill_delay 'nsecs'

The number of seconds between sending the signals as defined in the 'kill_sequence' (above).

sh 'bourne shell location'

The full pathname of the Bourne shell executable, which is normally '/bin/sh'.

ksh 'korn shell location'

The full pathname of the Korn shell executable, which is normally '/bin/ksh'.

csh 'C shell location'

The full pathname of the 'C' shell executable, which is normally '/bin/csh'.

purge accounts 'ndays'

When the controller process is started, an automatic purge of the 'adm' accounts directory is performed. Use this switch to define how many days accounts are kept *after* the purge.

delete_scripts yes/no

When a BQplus job completes, the job creation script is left in the 'scripts' subdirectory. When this switch is set to 'no' (normal) the script will be deleted; you can however set the switch to 'yes' to prevent the deletion.

**Tip :**

If you have a job that is failing and needs to be debugged online, use this option to save the job script for re-running from the command line.

default_log_protection 'protection_mask'

Allows the BQplus administrator to define the protection mask that will be applied to a BQplus logfile when the job is run. Normally set to '755' the administrator may choose to disallow 'world' access with a mask of '750' or something similar.

Privileged BQplus users : 'admin.cfg' and 'operator.cfg'

Description

A number of BQplus options are available only to those who have been defined as being privileged. BQplus recognises two types of privileged user, an administrator and an operator.

A BQplus administrator has the highest privilege within the BQplus system.

If a user is defined as a BQplus administrator or operator he/she may perform the following operations :

- *Start, stop and configure the controller*
- *Add, delete and modify BQplus queues*
- *Modify, delete, abort and display any BQplus jobs*
- *Configure the BQplus default values*
- *Submit a job as someone else*

The main difference between a BQplus operator and administrator, is that an operator may not effect any UNIX accounts defined in the following file :

'\$BQ_HOME/config/secure.cfg'

This includes all of the operations shown above in bold italics.

BQplus character menus : 'control.keys'

Description

This file defines the keys that are used to control the character based menu system : 'bqmenu'. It consists of a number of keywords followed by a unique key definition for that function in the menu system. The administrator should ensure that the keys defined in this file are unique as no checking is performed by the menu system itself.

Typically the keys defined will be a control code or a function key. Function keys may range from F1 to F12⁵ and are easily defined by typing the representation 'F1'...'F12'. Control keys are harder to define and should be input as the actual control key itself. This can be achieved in 'vi' by entering insert mode and pressing '<CTRL>V' followed by the '<CTRL>'key'' required.

For example to type a <CTRL>P in 'vi' the following key sequence should be used in insert mode :

<CTRL>V<CTRL>P

⁵ Some terminals provide F1-F4 functions keys only.

Options

Directive	Description
CANCEL_CHAR	<i>'cancel form key'</i>
HELP_CHAR	<i>'help pop-up key'</i>
SELECT_CHAR	<i>'select list key'</i>
TOGGLE_CHAR	<i>'toggle view-screen key'</i>
EXECUTE_CHAR	<i>'execute form key'</i>
QUIT_CHAR	<i>'quit menu key'</i>
HOST_CHAR	<i>'host change key'</i>
EDIT_CHAR	<i>pop up free form text editor</i>
TIME_CHAR	<i>display time in 'q+' screen</i>

Table 9 : Menu keys and descriptions

Terminal type definition

In addition to the definition of the menu control keys, 'control.keys' can also be used to define line drawing characteristics for terminals. This supplements the 'terminfo' definition, and can be used if the terminfo file does not display the menus using proper lines.

The syntax of a terminal type definition is as follows :

'Terminal Name' ABCDEF1

Where characters are as follows :

Character	Description	Ideal
A	Upper left	┌
B	Upper right	┐
C	Lower left	└
D	Lower right	┘
E	Horizontal	─
F	Vertical	│
1	Disable bold	<i>This is optional</i>

Table 10 : Menu line drawing characters

An indication of what characters provide the required graphics may be found by typing the following command :

\$ bqmenu -create_terminal

The output from this command can be used to select which characters from the alternate character set would provide the best likeness to a rectangle.

Accessing a terminal type definition

Having defined the terminal definition, an environment variable 'BQ_TERM' should be set to the relevant 'terminal_name'.

For example :

```
$ BQ_TERM=MY_VT100
$ export BQ_TERM
```

BQplus schedules : 'scheds.cfg'

Description

BQplus schedules are used to allow a job to be scheduled for multiple run times according to a users requirements. Two types of schedule are available, incremental and list; each are defined differently.

Maintaining schedules

BQplus schedules are provided to allow a job to be run periodically or according to a list of pre-defined dates and times. These schedules are defined in the 'scheds.cfg' configuration file and can be maintained using an editor such as 'vi' or through the menu systems.

A sample 'scheds.cfg' configuration file is shown below :

# sample scheds		
# configuration file		
fivemins	INCREMENTAL	5
overnight	LIST	/usr/bq+/config/over.lst

Table 11 : Sample 'scheds.cfg' configuration file

The example above shows both types of BQplus schedule.

'fivemins' is an incremental schedule which will schedule a BQplus job to run every five minutes. This will continue until the job is modified or deleted from the queue.

'overnight' is a schedule which uses a 'list' of dates/times to calculate when the job should be run. When the list is exhausted (all dates/times refer to a date in the past) the job will complete and be deleted from the queue. In this case the list of dates/times is located in '/usr/bq+/config/over.lst'.

Schedule types

Incremental schedules

'schedule_name'

INCREMENT

'nmins'

Where 'schedule_name' is a unique name to be assigned to the schedule being defined, and 'nmins' is the frequency (in minutes) at which the job should execute. A number of schedules are pre-defined in the 'scheds.cfg' file and are shown below.

minute	INCREMENT	1
fivemins	INCREMENT	5
fifteen	INCREMENT	15
hourly	INCREMENT	60
daily	INCREMENT	1440
weekly	INCREMENT	10080

Table 12 : Supplied incremental schedules

List schedules

No list schedules are provided as standard, but as the system is used and schedules are built up through the menus, entries as shown below will be placed in the configuration file.

'schedule_name' **LIST** **'schedule definition file'**



Tip :
Maintaining list schedules using an editor such as 'vi' can be time consuming, try using the schedule maintenance facility in 'bqmenu'.

Where 'schedule_name' is a unique name to be assigned to the schedule being defined, and 'schedule definition file' is a fully qualified filename containing a list of dates/times which should be used for determining when a job should be run. Valid dates/times are shown in Table 17 : Date/time syntax. An example is shown below.

weekly_update LIST /usr/bq+/config/weekly_update.lst

Updating jobs running on a schedule

When a change has been made to a BQplus schedule you may want to update all jobs running on that schedule to the earliest start time possible.

To achieve this the job should be modified to have a start time of 'asap'.

For example :

\$ batchq -modify 456 -start_time asap

To modify all jobs in the queue running on a particular schedule, the '-update_schedule' option has been provided for your convenience.

For example :

To modify all jobs running on a schedule "overnight" :

\$ batchq -update_schedule overnight

This will effectively run a ``-modify -start_time asap`` on all the appropriate jobs on your behalf.

Should you want to run a job immediately and have its schedule applied afterwards, you should use the following command :

`$ batchq -modify 456 -start_time now`

Remote BQplus users : 'remote.cfg'

Description

In order for a remote user to access the BQplus system they must be defined in the ``remote.cfg`` configuration file. The format of an entry is shown below.



Note :

Care should be exercised before defining a remote BQplus user. Once the user is in the configuration file, he may access the BQplus system through the ``Netserver`` without the need for a password.

For security reasons it is recommended that no user is given 'root' access from a remote host.

Format

`remote_username.hostname` `local_username`



Tip :
Only allow low privileged accounts to be used by remote users.

Where ``remote_username`` is the username who is permitted to access the local queue from the host ``hostname``. ``local_username`` is the account that you are allowing the user to use.

Accounting files

Flat ASCII accounting files are provided for easy interpretation of BQplus job activity. They are located in the '\$BQ_HOME/adm' directory and are named 'bq_acct.DDMMYY' or 'bq_acct.MMDDYY', depending on the country code definition (see section starting on page 63). Inspection of the accounting files can be via a simple editor such as 'vi' or through the BQplus menu system. Access to the 'adm' directory is restricted to only UNIX users with the 'bqplus' GID, the non-privileged user can audit job activity through the '.bq_history' file that is created in the home directory.⁶

Accounting files are automatically purged by the BQplus controller and by default are cleared out to contain files from the last 31 days.

```
# cat /usr/bqt/adm/bq_acct.160602
16/06/02.11:00.20 JOB 15 root Executing /scripts/backup.sh
16/06/02.11:00.21 JOB 15 root Failed 0.00:01 0.00:00
16/06/02.11:04.02 JOB 16 root Executing /scripts/backup.sh
16/06/02.11:04.07 JOB 16 root Aborted 0.00:05 0.00:00
16/06/02.11:07.31 JOB 1 root Executing /usr/bin/date
16/06/02.11:07.31 JOB 2 root Executing /usr/bin/date
16/06/02.11:07.31 JOB 3 root Executing /usr/bin/date
16/06/02.11:07.31 JOB 4 root Executing /usr/bin/date
16/06/02.11:07.31 JOB 5 root Executing /usr/bin/date
16/06/02.11:07.31 JOB 1 root Completed 0.00:00 0.00:00
16/06/02.11:07.31 JOB 2 root Completed 0.00:00 0.00:00
16/06/02.11:07.31 JOB 3 root Completed 0.00:00 0.00:00
16/06/02.11:07.32 JOB 4 root Completed 0.00:00 0.00:00
16/06/02.11:07.32 JOB 5 root Completed 0.00:00 0.00:00
```

Figure 27 : A sample accounting file

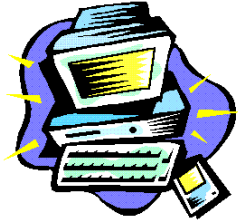


Note :

The two numeric fields shown at the end of the 'Completed' line are elapsed and CPU time respectively.

⁶ See Job History p.41

Chapter 7 : Menu systems



Description

BQplus comes complete with a set of 'curses' based menus which utilise the terminfo database and will therefore run on standard terminal emulation. This would include terminals such as VT100, VT220 etc.

Character based menus : 'bqmenu'

As described above the character based menus have been developed using the 'curses' libraries which in turn use the terminfo database to control the attributes associated with the terminal in use. The terminfo database uses the 'TERM' environment variable to determine which terminal definition is used, and must accordingly be set correctly.

Starting the menus

To start the menu system you should use the following command :

\$ bqmenu

This will bring up the front menu screen and will appear as follows

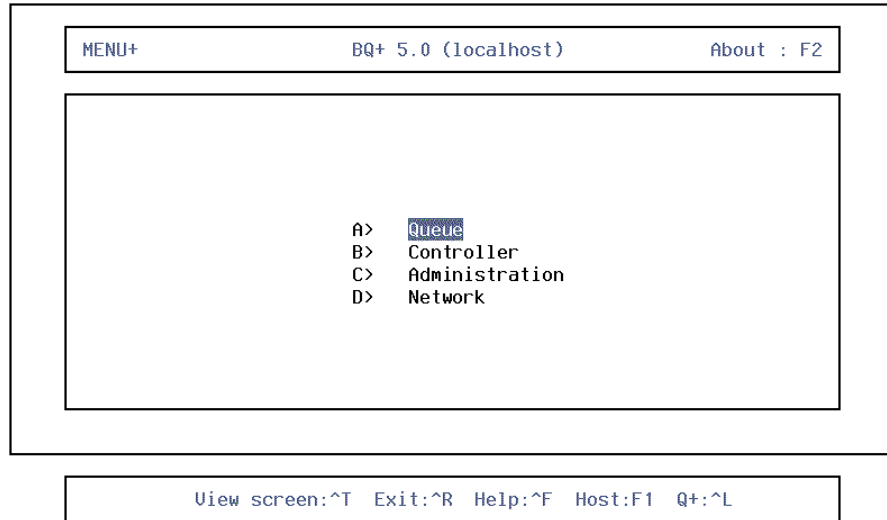


Figure 28 : The bqmenu front screen



Note :

Options 'C' & 'D' are available to administrators and operators only.

Configuring the control keys

In order to allow maximum flexibility the control keys used by the menu system are completely user-definable. The configuration file 'control.keys' is provided for this purpose and is discussed in more detail on page 74 :Configuring the control keys.

Using the menu system

The menu system works by presenting the user with a number of menus and submenus which are selected by the use of cursor keys and the <return> key. Alternatively the user can select a submenu by pressing the alpha character associated with that submenu.

Having reached the submenu with the required operation the user is then presented with a pop-up form which can be completed as appropriate and

executed by pressing the 'execute' key. Focus then switches to the 'BQplus view screen' where the corresponding BQplus action is performed. An example of this process is shown below, where the user has entered the menu system and is submitting a new request.

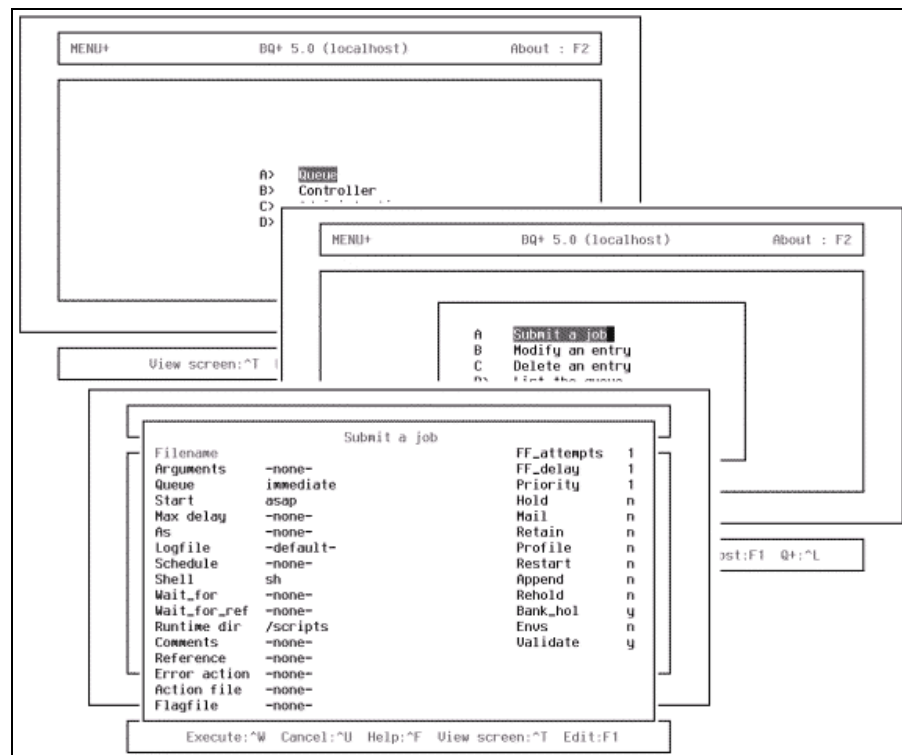


Figure 29 : An example menu operation



Note :

Certain fields in bqmenu forms are only valid if another option has been specified, until this is true the field will be disabled. This means that if an attempt is made to move to such a field the cursor will in fact be placed on the next enabled field. When the necessary options have been supplied, disabled fields will be enabled automatically.

Printing the output displayed in the view screen

When the view screen is displayed you may see a prompt of 'p : print' displayed at the bottom of the screen. If this is the case it means that your BQplus administrator has set up a print command for the menu system to use. When you want to print the screen contents you should press the 'p' key, this will instruct BQplus to send the display to the pre-defined print command. If no print command has been defined printing is not possible and you should contact your BQplus administrator to install an appropriate print_command. Details of how this can be achieved are found in the section BQplus defaults : 'defaults.cfg' on page 63.

Changing BQplus host

When the bqmenu is first started you will notice that on the top of the menus a name of 'localhost' is displayed in brackets after the BQplus title. This indicates that any operations carried out through the menus will be performed on the machine which you are currently logged on to. ie. the local host. It is possible to administer remote systems through the menus by using the 'netserver' inter-host facility. To start the process of changing the menus to work on another BQplus host you should follow this procedure :

- | |
|--|
| 1. Press the host key, as displayed at the bottom of the screen. |
| 2. Select the required host from the pop up list. |
| 3. Wait while you are authorised on the remote host. This will be confirmed with a pop up message box, which will also indicate the account the you will use on the remote machine and the corresponding level of BQplus access. |

Table 13 : Changing BQplus host

Should you require to return to working on the local machine, you should follow the procedure as shown previous and select 'localhost' from the pop up list of hosts.

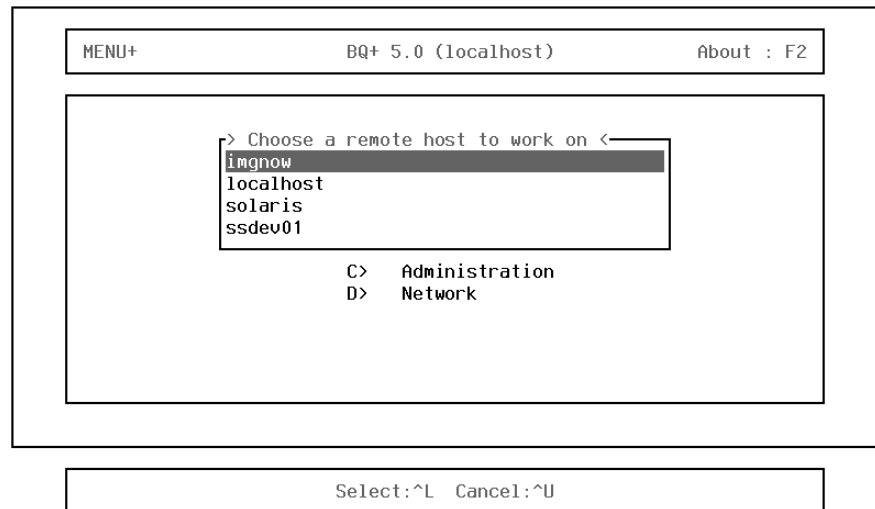


Figure 30 : Administering a remote BQplus system

Changing the help text

Changing the help text is a simple exercise which requires knowledge of a UNIX editor such as 'vi'.

The help text for the menu system is held in the file '\$BQ_HOME/config/bqmenu.hlp', this file is used by both the character and X-Terminal menu systems. The file is not an ASCII file and cannot be edited directly. To change the help text of a particular item in the menus you should follow this procedure :

1. Use the menu system and note down the help text which you want to change.
2. Edit the file '\$BQ_HOME/bin/makehelp.sh'
3. Locate the help text that you want to change.
4. Make the changes.
5. Save the file.

6. Update the 'bqmenu.hlp' file by executing the following command :

```
$ $BQ_HOME/bin/makehelp.sh
```

Table 14 : Changing the menu help text

The help text will now be updated and can be viewed by returning to the appropriate menu screen and requesting help.

Using select boxes

Select boxes are a useful way of saving time and ensuring that any values entered into a field are valid.

From a form, the user moves between fields by using either the <return> or cursor keys. Alternatively the first character can be used to move down a select list quickly.

If a select box is available for a particular field the 'select key' prompt will be displayed at the bottom of the screen. If no 'select key' prompt is displayed, no select box is available.

Having pressed the select key the user is presented with a pop-up select box from which a value can be chosen. This is achieved by moving up and down the list using the cursor keys and pressing the select key again when the required item is highlighted. Having selected an item the pop-up select box is removed and control returned to the form, where the selected value is now displayed in the current field.

At any time in the select pop-up the user can press the 'cancel key' and return to the form without having selected a value.

MENU+	BQ+ 5.0 (localhost)	About : F2
Standard queue listing		
Entry nos	-any-	> Choose a queue < -any- backups immediate over reports
Queue name	-any-	
User name	-any-	
State	-any-	
Start date	-any-	
Reference	-any-	
Schedule	-any-	
Full	n	
G Update jobs on a schedule		
Select: ^L Cancel: ^U		

Figure 31 : Using a menu select box**Command line options**

A number of command line options are available for the bqmenu program. In previous versions of BQplus these have not been made known to the user as the majority of the facilities offered by these options are specific to the menu system. However a number of installations have found these options useful for developing in-house BQplus scripts and these are now made available for all users.

\$ bqmenu 'option'



Note :

Should a command line option be incorrectly typed the user will by default enter the menu system.

Listing the BQplus queues

-list_the_queues

Will display all currently configured BQplus queues in a simple list, one queue will be displayed per line.

Listing a queue definition

-list_queue_def 'queue_name'

Will display a line containing the following information about a BQplus queue :

- **Maximum concurrent jobs**
- **Maximum elapsed time**
- **Start time of the queue**
- **Stop time of the queue**
- **Nice adjustment**
- **Pre-process command**

Each queue characteristic is separated by a space character which can be used as a delimiter in a 'cut', 'strtok' or similar operation. If the pre-process command contains any spaces, it will have had these converted to a <CTRL-A> character. It will be necessary to reconvert these back to spaces if you want to use this value.

Displaying job information

-list_entry 'entry_no'

Will display a line containing the information about a BQplus entry. Each entry characteristic is separated by a space character which may be used as

a delimiter in a 'cut', 'strtok' or similar operation. Any characteristic that contains a space itself, will have had any spaces converted to a <CTRL A> character. It will be necessary to reconvert these back to spaces if you want to use these characteristics.

Brief listing of the queue

-list_queue 'queue_name'

Will display a brief listing of any jobs that are found in a BQplus queue, using the following format :

- ***Entry number***
- ***UNIX userid***
- ***Filename***

Each job characteristic is separated by a space character which can be used as a delimiter in a 'cut', 'strtok' or similar operation.

Modifying a queue



Tip :

Use this option to automatically open a queue up overnight. eg. create a batch job that extends the maximum concurrent jobs overnight and another job that reduces it back again for daytime processing.

-mod_queue 'queue_name' 'options'

Allows the user to modify any characteristics associated with a queue. Options are as follows :

- max_cc 'max_jobs'***
- max_elapsed 'max_elapsed_time'***
- start 'start_time'***
- stop 'stop_time'***
- nice 'nice_adjust'***

Care should be used with these options, as they change queue characteristics immediately and therefore are very powerful.

Listing the logs

-list_the_logs

Allows the user to generate a list of all valid logfiles for inspection.



Note :

This option uses the '.bq_history' file to obtain a list of logs for validation. If the use of history files is disabled by the BQplus administrator, no history select operations will be possible.

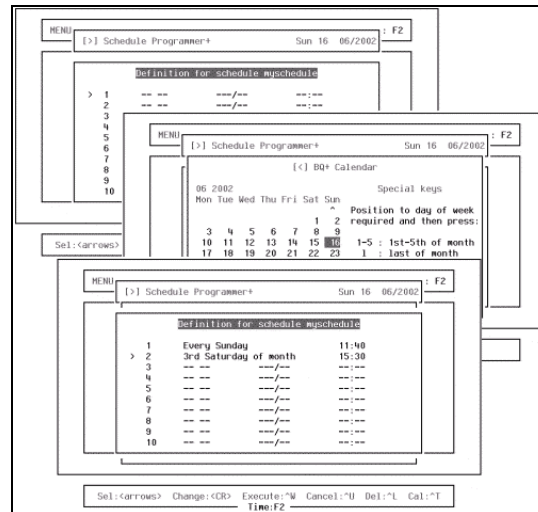
Maintaining a list schedule

In order to maintain list schedules from the menu system and special form has been provided and is called 'Schedule Programmer+'. From this form it is possible to maintain a list schedule definition that contains up to 999 different entries.

Use of the form requires some practice, but the fundamentals are shown on the next page.

Procedure

- From the initial screen you can move the cursor up and down to select the 'slot' to be programmed.
- Having positioned to the required slot the <return> key should be pressed to begin programming.



- The first item to be programmed is the date field and this can be achieved by either using the up/down arrows or by using the 'BQplus Calendar' screen by pressing the 'Calendar' key.
- If you choose to use the 'Calendar' screen you will be presented with a typical look and feel screen which allows the selection of a date or special day (as displayed on the right hand side of the screen)

- Having selected the appropriate date the <return> key should be pressed again and the cursor will move to set the hour required.
- Select the hour required by use of the up/down arrows and press <return> to complete and move the cursor to set the minute required.
- Use the up/down arrows to set the minute required and press <return> one more time to complete programming the slot.
- Return to the start of this procedure and repeat as required ...

Chapter 8 : The World Wide Website



A website has been set up by Corporate Practical Solutions and features a set of pages for the BQplus product. Facilities for download, feedback, tips and late breaking news are available for viewing.

Come and visit us

CPS
Corporate Practical Solutions
specializing in Unix and Microsoft Windows utilities since 1992

Sun Nov 17 2002

Download BQplus today for a free 30 day trial

This site © 2002 CPS Limited

Home Products Download Support Contact News About

Please select a product datasheet: BQplus job scheduling

BQplus™
Job scheduler

Job Scheduling for UNIX

Over one million jobs are processed by BQplus every month

BQplus is a mainframe class batch job scheduler designed with *Operations staff in mind*, ensuring batch job management on UNIX computer systems is as versatile and straight-forward as possible.

"BQplus is the easiest to install & quickest to learn"

Testimonials

next comment
bqplus plays a key role in scheduling our major batch operations

leading university

- Quickly move your processing cycles into BQplus with the easy to use scheduling and dependency features
- All command-line and menu tools have been designed with Operations staff and are "machine-room approved"
- Security options allows you to restrict access to BQplus queues and jobs as necessary

Need more information?
A flash presentation is available with more information on BQplus

<http://www.grepit.com>

Online documentation

Any documentation that is available online is distributed in Adobe Acrobat (TM) format. This documentation can be viewed using an Acrobat Reader, which is available for Windows 3.11, Windows 95, Windows NT, Macintosh, OS/2 and variety of UNIX implementations.



The Acrobat Reader is freely available for download from the following World Wide Website :

www.adobe.com



Note :

A link direct to the download page can be located on the Corporate Practical Solutions website, in the 'download' sections.

Chapter 9 : Appendices



Keywords

Description

BQplus keywords are special words which are enclosed in percentage ('%') marks. They may be specified in certain strings that are supplied to BQplus and will be replaced by appropriate values at job execution time. The following table indicates all of the BQplus keywords and their meaning.

Available Keywords

bq_request	The entry number of the BQplus request that is currently being executed.
bq_argument	Any arguments to the BQplus job that is currently being executed.
bq_queue	The name of the BQplus queue from which the job is running.
bq_reference	The job reference text attached to the current BQplus request.
bq_user	The UNIX username of the id used to submit the BQplus request.
bq_as_user	The UNIX username of the id that will be used to run the BQplus job.
bq_home	The home directory of the BQplus installation. This would normally be '/usr/bq+' or '/opt/bq+'.
bq_argument	The arguments being passed to the program being run.
bq_logfile	The name of the logfile that is currently in use for the BQplus job.
bq_attempts	The number of times that BQplus has started the entry being executed.

Table 15 : BQplus keywords

Options supporting keywords

The following BQplus options support parameters that contain keywords :



- action on error commands
- pre process commands
- default_log_dir switch
- home_command switch

For example :

An action on error utilising keywords could be defined as follows :

```
error_message /bin/echo Request %bq_request% \  
failed with arguments of %bq_argument% >/dev/console
```

BQplus job states

State	Description
(new)	Job is ready to be run by the BQplus controller.  <i>Note :</i> A state of 'new' is sometimes displayed for clarification purposes, although the field is usually blank to indicate a job that can be run.
exec	The job is currently executing.
held	The job is in a held state and will not be started by the BQplus controller.
error	The job has executed and failed to complete with a successful exit code of zero.  <i>Note :</i> A full listing of the queue will display the information field which will explain why a job has failed.
time	The job has been requested to start at a particular date or time, which is in the future.

Continued

wfor	The job has one or more job dependencies which have to be satisfied prior to execution.
------	---

Table 16 : Job states

Date and Time format

A valid date/time string consists of a date, a time or both. The format of each must comply with the following :

Date : DD/MM/YY (UK) or MM/DD/YY (US)
 Time : HH:MM
 Date/Time :
 DD/MM/YY.HH:MM (UK)
 MM/DD/YY.HH:MM (US)

Examples of these are as follows :

19/06/96 (UK) or 06/19/96 (US)
 17:30
 19/06/96.17:30 (UK) or 06/19/96.17:30 (US)



Note :

From BQplus 4.5 onwards the year may now be specified in a four digit format, ie. 1998 and 2000.

Date keywords may be substituted for date components of a date/time string.

Description	BQplus date
Day of the week	mon, tue, wed, thu, fri, sat, sun
1 st , 2 nd 3 rd etc. of the month	1mon, 1tue ... 1sun 2mon, 2tue ... 2sun 3mon, 3tue ... 3sun 4mon, 4tue ... 4sun 5mon, 5tue ... 5sun

Continued

Every day	emon, etue ... esun
Last day of the month	lmon, ltue ... lsun
1 st , 2 nd , 3 rd etc. working day of the month	work1, work2 ... work31
Weekends	wend
Weekday	wday
Day of the month	mday1, mday2 ... mday31
Special days	workl : last working day mdayl : last month day asap : as soon as possible now : current date & time

Table 17 : Date/time syntax**Note :**

Whilst it is possible to use any of the above keywords with the '-start' option, some of the keywords such as every and working days are probably of more use in list schedules (page 44).

Defining the country for date syntax

As standard the BQplus environment is configured for UK style dates, ie. DD/MM/YY. You may however choose to use US style dates (MM/DD/YY) by enabling the switch 'us_dates' in the configuration file 'defaults.cfg' (page 68).

Running multiple copies of BQplus on the same machine

Description

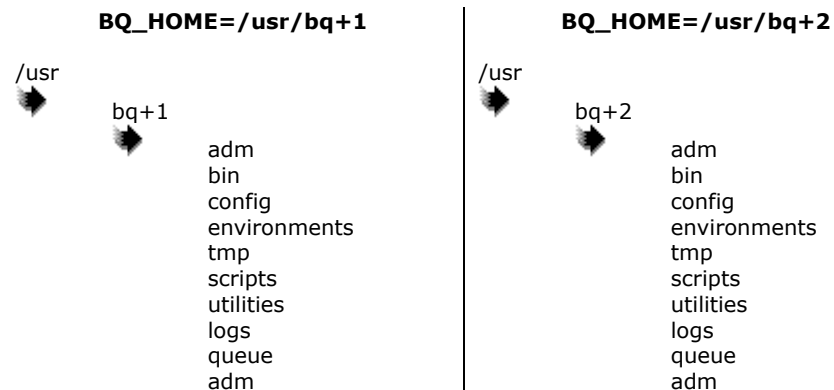


Tip :

In order to save space, the 'bin' directory in one of the BQplus systems could be a symbolic link to the 'bin' directory in the other.

A particular installation may require a specific machine to run more than one BQplus system at the same time. This might be useful for example, in a clustered configuration. This is possible, and can be achieved by using more than one BQplus directory subsystem and an environment variable defining the BQplus home directories.

For example to run two BQplus systems the following might be set up :



In this scenario, two BQplus systems 'bq+1' and 'bq+2' have been installed onto the '/usr' partition.

In order to access the correct BQplus system, the environment variable 'BQ_HOME' must be defined as the appropriate home directory; i.e. '/usr/bq+1' or '/usr/bq+2'.



Note :

If no 'BQ_HOME' variable is defined, '/usr/bq+' will be used. In this case '/usr/bq+' should be set up as a symbolic link to either '/usr/bq+1' or '/usr/bq+2'.

In order for both systems to operate, two BQplus controller processes must be started. This can easily be achieved by using the 'BQ_HOME' environment variable to select the appropriate system and the 'bcon -start' command.



Note :

Running more than one BQplus system on the same machine can get a little complicated. If you have difficulty, first check that the 'BQ_HOME' environment variable is set correctly.

INDEX

A			
accounting	77	-flagfile_delay	25
acrobat reader	89	-hold	28
action files	53	-host	54
action_on_error	61	-last_job	45
actions.cfg	61	-list	35
admin.cfg	72	-list -full	36
administrator	55	-list -queue	37
api		-list -reference	39
menu command line options	85	-list -state	38
appending output	21	-list -suite	47
applications programming interface		-logfile	21
menu command line options	85	-mail	28
argument_parser	48	-max_delay	22
arguments	20	-no_bank_holiday	45
audit		-no_log	21
netserver	60	-no_validate	30
audit trail	77	-priority	27
automatic options	42	-profile	26
not requiring a parameter	43	-purge	41
requiring a parameter	42	-queue	20
Automatic retrying of failed jobs	29	-quiet	27
B		-reference	24
batchq	19	-rehold	28
-action_file	53	-reply	51
-action_on_error	29, 30	-restart	27
-answer	51	-retain	29
-append	21	-retries	51
-arguments	20	-retry	33
-as	30	-runtime_dir	23
-ask	51	-save_job	45
-clear_wait_for	46	-schedule	44
-clear_wait_for_ref	46	-shell	23
-comments	28	-start_time	22
-delete	34	-terminal	51
-envs	23	-unhold	33
-error_attempts	29	-update_schedule	75
error_delay	29	-use_envs	46
-fix	58	-wait_for	24
-flagfile	25	-wait_for_reference	25
-flagfile_attempts	25	bcon	
		-extend	14
		-jump_start	57

-list	58
-list -suites	49
-log	15, 57
-log netserver	60
-pause	58
-start	55
-start netserver	59
-status.....	49, 56, 60
-stop.....	56
-stop netserver	59
-unpause	58
BQ_HOME	16
BQ+ IPC	51
BQ+ location.....	16
bqmenu	80
-create_terminal	73
-list_entry	85
-list_queue	86
-list_queue_def	85
-list_the_logs	86
-list_the_queues	85
-mod_queue	86
starting	80
C	
character based menus	
bqmenu	79
check_queue_state	65
combining listing options.....	40
comments.....	28
communicating with a running job.....	51
configuration files.....	61
actions.cfg.....	61
admin.cfg	72
control.cfg.....	69
control.keys	72
defaults.cfg.....	61
operator.cfg	72
queues.cfg	68
remote.cfg	76
scheds.cfg.....	74
configuring menu control keys.....	80
confirm_deletions.....	64
console_device	64
control.cfg.....	15, 69
control.keys	72
controller	
log file	57
overview.....	55
signaling.....	57
starting	55
status.....	56
stopping	56
controller overview	55
controller_jump_signal	67
corrupt job	27
csh	71
D	
date/time string	93
default_flagfile_delay.....	63, 66
default_log_dir.....	65
default_log_protection	71
default_log_umask	67
default_queue	62
default_shell	62
defaults.cfg	61
delete_scripts	71
deleting a request	34
dependency	
entry number	24
job reference.....	25
developing in house scripts	
menu command line options	85
DISABLE	50
disk space.....	11
display_args	62
display_poll_info.....	69
display_reference.....	62
display_run_time	66
documentation	
online.....	89
E	
enable_history	66
enable_suites	67
entry number	37
environment variables.....	23, 42
modifying a job	46

options not requiring a parameter	43	options supporting	92
options requiring a parameter	42	kill_delay	71
evaluation period	14	kill_sequence	70
exclusive_suite	50	kill_signal	70
exit status actions	53	ksh	71
F		L	
failed jobs	29	latest start time	22
filename validation	30	license_code	15, 70
fixing the queue	58	licensing	14
flagfile dependency	25	extending evaluation	14
FORM	50	permanent	14
full listing	36	list schedules	44
G		list_full_queue	64
GID	11	listing a queue	58
group	69	listing the queue	35
H		log file	
help text	83	controller	57
history	41	netserver	60
holding a job	28	log files	21
home_command	62	logs	
hostfile	66	listing	86
I		M	
incremental schedules	44	machine name	15
INIT	50	Machine types	14
installation	11	mail_program	67
installbq+	12	max_concurrent	68
instant_menus	64	max_elapsed	68
inter process communication	51	menus	
J		changing host	82
job		changing the help text	83
information API	85	character based description	80
job dependencies		command line options	85
updating	46	control.keys	72
job failure action	29, 30	printing output	82
job states	36, 92	starting character based menus	80
job suites	47	using select boxes in character based menus	84
K		modifying a request	31
keywords	91	multiple BQ+	95
		multiple flagfiles	25
		N	
		netserver	
		audit	60

log file	60	R	
starting	59	read_and_execute	63
status	60	reference	24, 39
stopping	59	remote working	54
nice_adjust	69	remote.cfg	76
no_logname	64	remote_audit	65
no_queue_priorities	65	request number	
no_signal	66	retrieving	45
no_system_scripts	64	saving	45
notification	28	retrying a job	33
O		running a job using another identifier	30
online documentation	89	runtime directory	23
operator.cfg	72	S	
P		saturday_working	67
PATH	16	scheds.cfg	74
pausing a queue	58	schedule	40
permanently licensing	14	incremental	74
platforms	9	list	75
poll_timer	70	scheduled jobs	
pre_process	69	updating	75
print_command	63	scheduler_priority	70
prioritisation of a job	27	schedules	74
purge_accounts	71	maintaining	74
Q		maintaining list	86
queue		types	74
brief listing API	86	scheduling	44
listing	58	scheduling a job	44
maintenance	58	security_level	63
pausing	58	selective listing	36
unpausing	58	sh71	
queue name	37	shell type	23, 94
queues	20	signal	70
definition API	85	signaling the controller	57
listing API	85	silent running	27
modification API	86	single_line_filename	63
queues.cfg	68	staging area	11
quick_menus	64	start time	22
quick_select	66	start time of a queue	68
quota_timer	70	starting the controller	55
		starting the netserver	59
		startup scripts	26
		state	38
		states	36, 92

status	68	line drawing	73
stop time of a queue	69	TITLE.....	50
stopping the controller	56	turning off a log file	21
stopping the netserver	59	turning off an option	32
suite		U	
name	50	unholding a job	33
suite_name	50	unpausing a queue	58
suites	47	us_dates	63
argument validation	48, 49	V	
definition	47	validation	
definition parameters.....	48	job arguments	48
interdependency	48	W	
listing jobs	47	wait for	
scripts	50	by entry number.....	24
status	49	by reference	25
submitting a job	47	wait_for	48
using from within the menus.....	49	Website	9
wait_for	48	write_command	67
sunday_working	67	write_reference.....	67
symbolic link.....	16		
symbols	10		
T			
terminal			