Xi Software

GNUbatch Release 1

MS Windows Interface

This manual is for GNUbatch (MS Windows Interface Manual).

Table of Contents

1 Introduction

GNUbatch is a fully functioned, high performance Job Scheduler and Management System which is available for a wide range of machines running a Unix-style Operating System.

The product consists of a "core product" or "basic product" which contains the scheduling software, command-line and character-based interfaces. Additional options provide for:

- An X-Windows Motif Toolkit Interface
- An X-Windows GTK+ Toolkit Interface
- An API for use with C and C++
- An Interface for MS-Windows
- An API for use with MS-Windows
- Browser Interfaces

The basic manuals cover the "basic product" and the X-Windows interfaces. Additional supplements cover the other interfaces.

The basic manuals are:

- User Guide - a quick introduction and "cookbook" for use of GNUbatch
- Reference manual - a complete description of all components of the basic product.
- Administrator Guide - information about installation and customisation of the software.

Also available are:

- API reference manual for Unix and MS-Windows API
- MS Windows Interface Manual (this manual)
- Browser Interface Manual

This manual for the MS Windows Interface describes the facilities of the Windows interface only and assumes knowledge of the basic product.

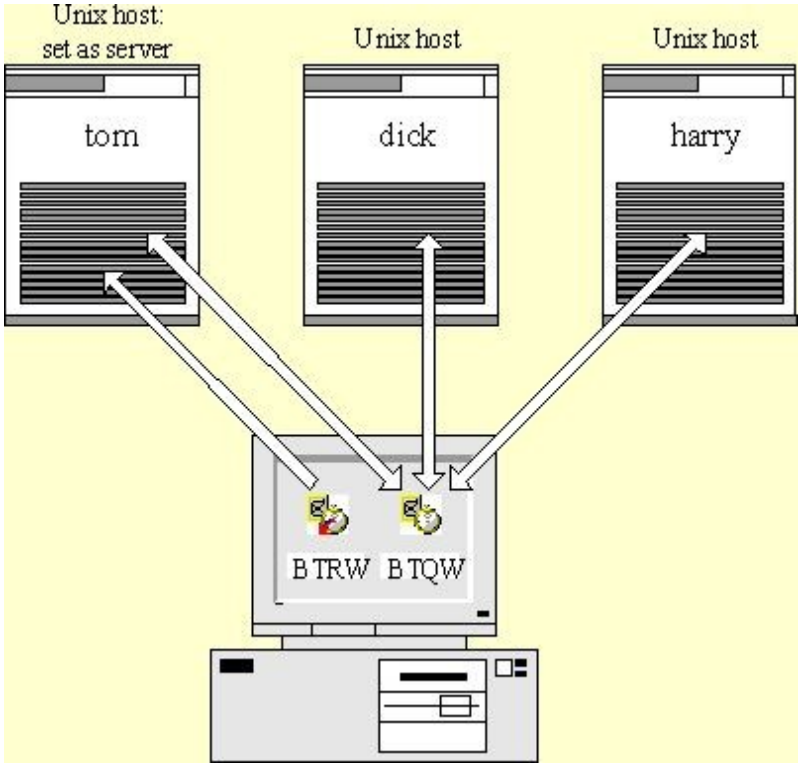
The original version of GNUbatch was written by John Collins at Xi Software Ltd between 1990 and 2009 as “Xi-Batch” and GNUbatch is based on Release 6 of Xi-Batch. The names, including most of the program and service names, have been changed to GNUbatch or derivatives and the installation directories have been changed to conform to GNU standards. (For this reason some of the diagrams may refer to Xi-Batch rather than GNUbatch).

The MS Windows interface is currently written for MS Visual C++ and is provided in source form but urgently needs rewriting to use only Free Software. This manual outlines how the MS Visual C++ version operates.

2 Overview

GNUbatch can run on a single Unix host or several co-operating machines. The Windows software can manage the batch jobs on one or more of these Unix hosts. It can also submit new jobs to any one host, known as the Server. The Server is specified at the PC, hence different PCs can use different Servers and each PC can change Server.

The program which manages jobs is called btqw and the program to submit jobs is called btrw. Their relationship to a group of three Unix hosts might look like this:



In this example, program btqw can see and manage jobs on hosts `tom`, `dick` and `harry`. Program btrw submits new jobs to host tom.

2.1 Jobs & Variables

GNUbatch maintains a queue of batch jobs. Each batch job consists of a script and a specification of what GNUbatch should do with it.

Job Control Variables are provided by GNUbatch to manage dependencies between jobs. Jobs can include specifications to set or modify the values of variables when they start or finish. On finishing, different operations can be specified for variables depending on whether the job worked or failed.

The job specification also includes conditions on variables, which GNUbatch tests before letting that job start.

2.2 Owners, Groups and Modes

Like Unix files, all jobs and variables are owned by a user and belong to a particular group. This is used to say who may see and edit a job or variable. These work with the protection modes.

Each job and variable is given a *protection mode*. This consists of a set of permissions dictating how various users may, or may not, access the job or variable. The modes are like those on Unix files, providing *user*, *group* and *other* access. An expanded set of permissions has been devised to enable the permissions to control separate operations.

The modes of jobs and variables are set when they are created, however users authorised by the mode may reset them subsequently.

In the case of *jobs*, the modes set by btrw are used, in default of which a set of default modes for the given user are set.

3 User Programs

There are three Programs which are installed on a Windows PC. The installation normally sets up a Program group called *GNUbatch Windows Interface* in the Program Manager. The programs are described in following chapters. Features that are common to more than one program, like the dialog for setting job start times, are described in one chapter.

The three programs are:

Batch Queue Management Tool Shows the Queue of Batch Jobs and Set of Job Control Variables, allowing users to change them according to the various permissions.

Job Creation Tool enables jobs to be specified and then submitted to the Batch Queue. Jobs can also be "unqueued" using btqw, then changed and re-submitted using btrw.

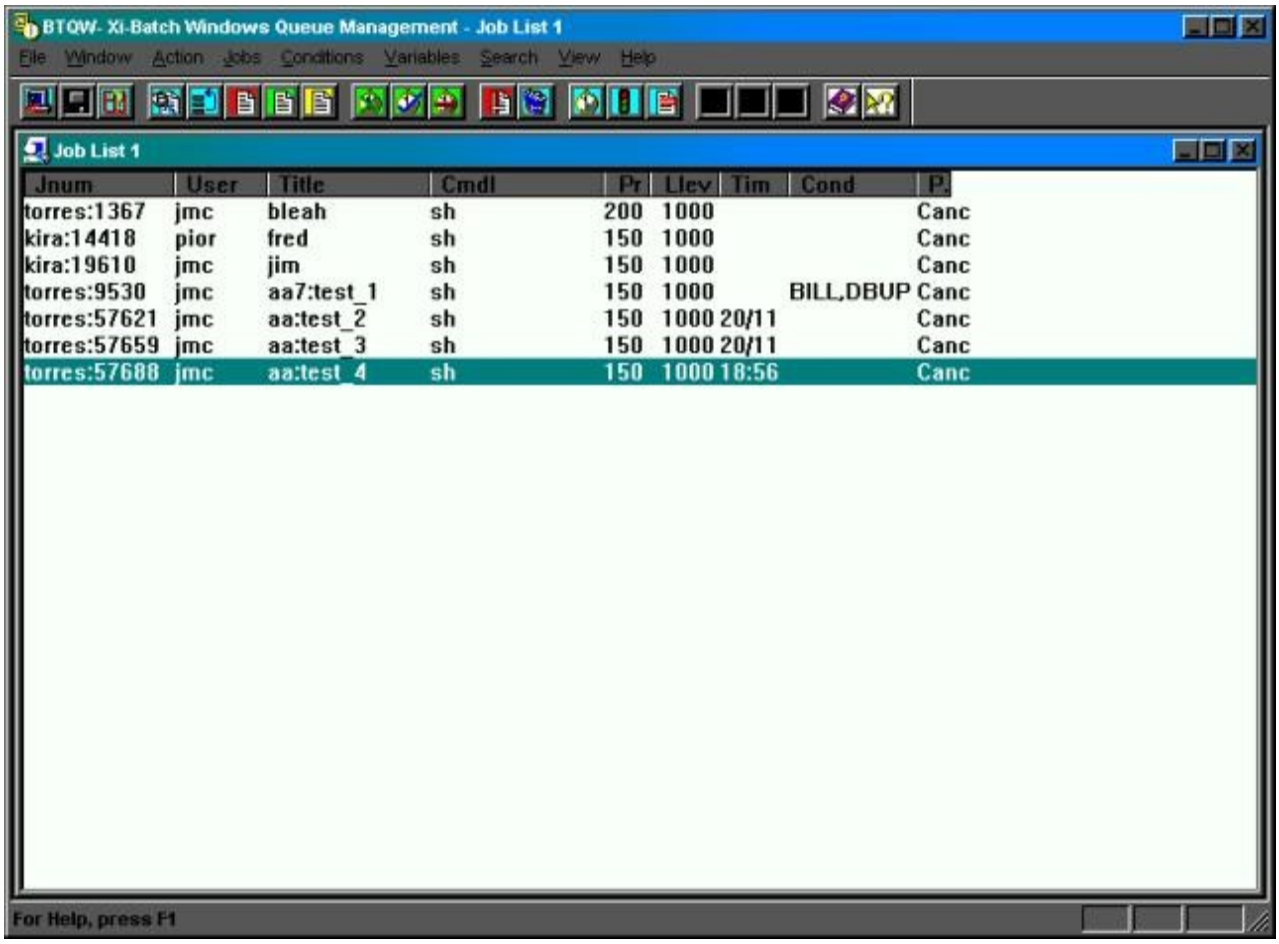
Set Up Tool configures which Unix hosts to display jobs from and which host to submit jobs to.

4 Btqw - Queue Management Tool

Btqw is a Microsoft Windows alternative to the standard batch queue manager, btq. It is usually invoked from the Windows Program Manager, or from the desktop by clicking on the Btqw icon shown above.

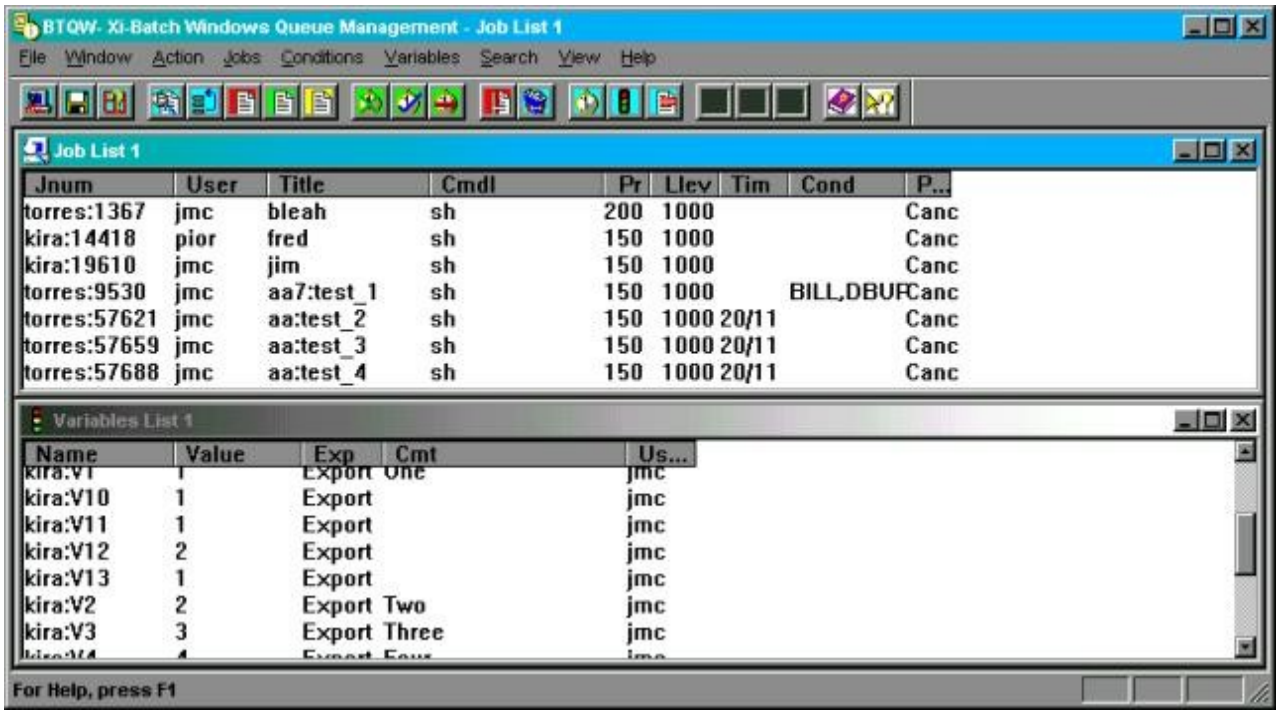
4.1 The Main Window

When btqw is invoked the main window will be displayed, with a sub-window for jobs. By default it will look something like this:



The main window has two key functional areas. The top area contains menus and short cut buttons for issuing commands. The larger, bottom area, holds the sub-windows for batch jobs and variables. These may be selected to have commands performed upon them.

The bottom area may have any number of windows in it. It may be empty, or it may have windows that are minimised. For example:



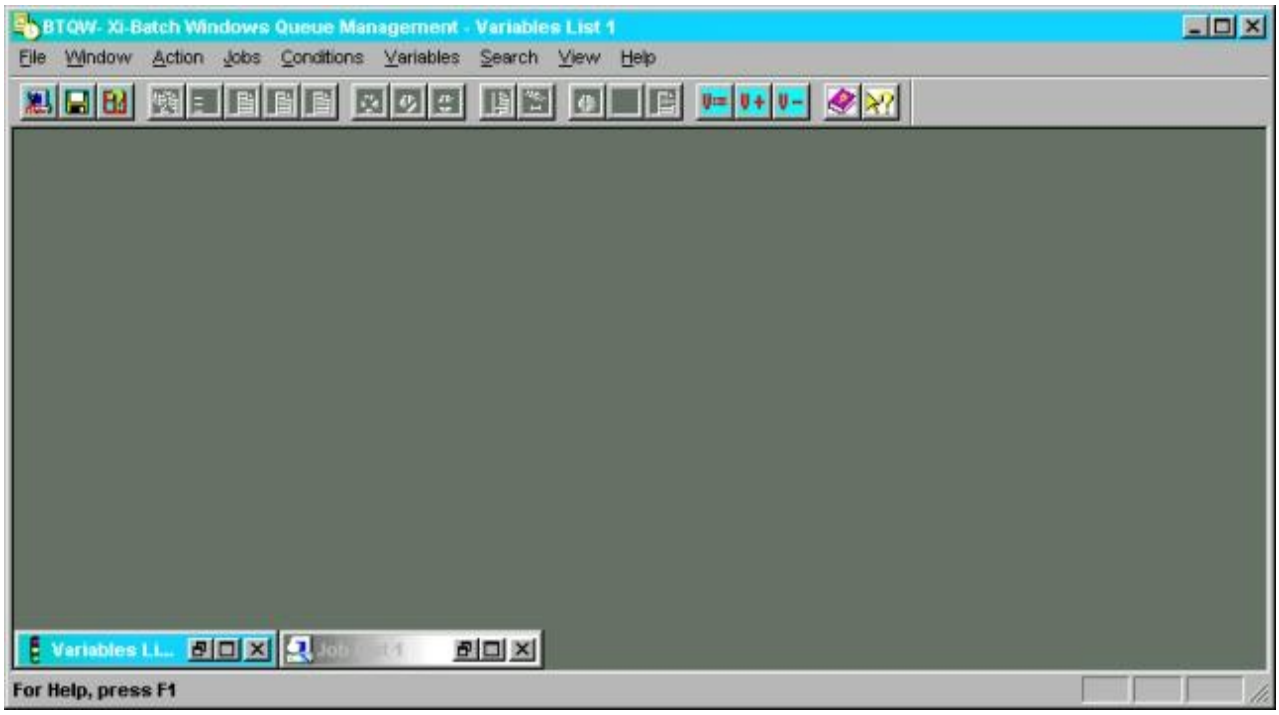
This shows a sub-window with a Job List and another with a Variable list tiled horizontally. This is an option which divides the available space equally between each of the sub-windows, laying them out one above the other.

Just like any other windows the job and variable list could be overlapping like this:

Windows can be overlapped using the cascade option.

All of these options are available from the Window menu, which is described later. The windows can be moved and resized just like those in any other program.

If both sub-windows are minimised, just their icons will be shown at the bottom of the window like this:



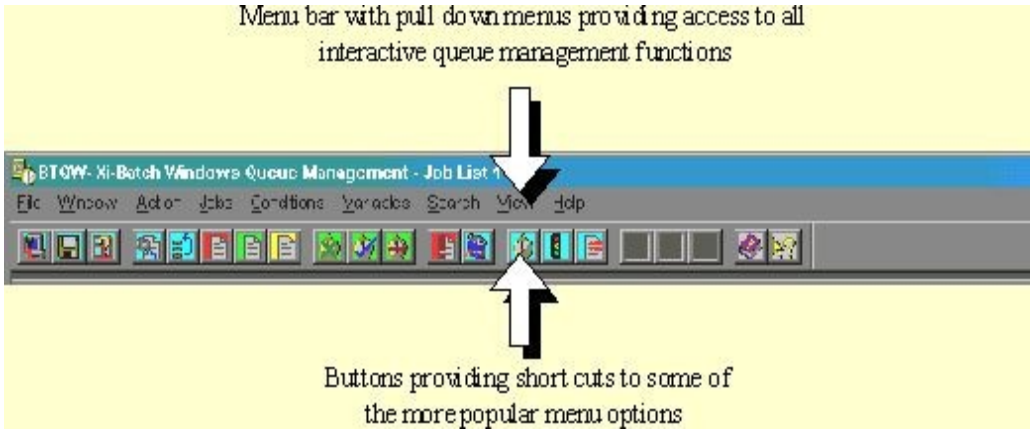
Variables and jobs can be operated upon using the appropriate menu options, or shortcut buttons. The job or variable must first be selected. This is done by clicking on the line identifying it using the mouse. Once selected the line will be high lit.

If you cannot see the variable or job that you want then you may:

- Use the scroll bar or search menu options to find it
- Change the view to add the item or remove unwanted items from the display.

If a menu option or button is "greyed out" it means that a suitable job or variable has not been selected. It may be that the required item is selected but not the sub-window that contains it. Look at the "title bar" of the sub-window to see if it is the "active window".

The key features of the top area are:

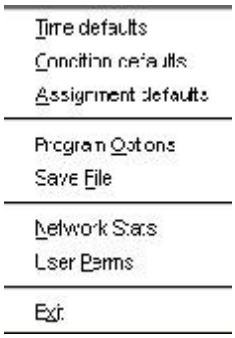


4.2 The Menus and Shortcut Buttons

All commands are performed by selecting a menu option or clicking on the equivalent shortcut button. Some of the menu options may also be selected using shortcut keys, which are indicated to the right of the relevant options in each menu.

4.2.1 The File Menu

For tailoring the look and feel of btqw, setting up defaults, saving the tailored settings and quitting.



Time defaults opens a dialog for setting default time specifications for jobs.

Condition defaults opens a dialog for setting default conditions for jobs.

Assignment defaults opens a dialog for setting default assignments for jobs.

Program options brings up the Program options dialog, to tailor the look and feel of btqw.

Save File saves various program options, in a file on the PC, as the default settings for the next time btqw is invoked.

Network Stats opens a diagnostic window showing connection details. Ignore this option unless Technical Support ask for information from it.

User Perms brings up a dialog showing permissions and parameters like User Load Levels. This can also be ignored for everyday use.

Exit quits btqw.

The following shortcut buttons are provided on the toolbar.

 Is a shortcut button for **program options**

 Is a shortcut button for **Save File**

4.2.2 The Window menu

For opening new job and variable list windows, changing the layout of windows and formatting their contents.

New Job Window opens a new window containing a Job List.

New Variable Window opens a new window containing a Variable List.

b>Window Options opens a dialog for selecting which jobs and variables to display.

Cascade Re-arranges the job and variable windows so that they overlap showing the title bar of each window.

Tile Horizontally Re-arranges the job and variable windows so that they do not overlap. Each sub-window stretches the full width of the main window.

Tile Vertically Re-arranges the job and variable windows so that they do not overlap. Each sub-window stretches the full height of the main window.

Arrange Icons Re-arranges any icons at the bottom of the main window into a neat row.

Job display Format Opens a dialog for specifying the contents and layout of the currently selected Job List.

Job Colours Opens a menu of options for displaying jobs in various states in different colours.

Var display fmt Opens a dialog for specifying the contents and layout of the currently selected Variable List.

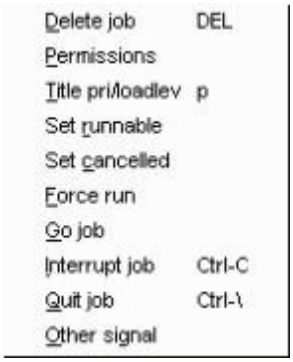
Save widths as default Saves the widths set by moving the borders on the title bar as default widths for future runs.

The following shortcut button is provided on the toolbar.

 Is a shortcut button for **Window Options**

4.2.3 4.2.3 The Action Menu & Buttons

For high level actions: starting and stopping batch jobs.



Delete job removes the selected job from the queue. This is a destructive operation, with no way to re-queue the job. A new copy will have to be submitted if required.

Permissions brings up the dialog to set the access modes for the selected job.

Title, pri/loadlev opens the dialog for setting the Title, Priority, Command Interpreter, and load level for the job. This is also available using the shortcut button.

Set runnable will change a job from the *Cancelled*, *Error* or *Abort* state to the *Ready* or *Run* state. This option is also available using the shortcut button.

Set cancelled puts a job on held (i.e. not able to run). This option is also available using the 'Set cancelled' shortcut button.

Force run sets a job runnable and overrides any time specification to allow the job to run as soon as any Variable Conditions are satisfied. This option is also available using the 'Force run' shortcut button.

Go job sets a job runnable overriding any time specification to allow the job to run as soon as any Variable Conditions are satisfied. The repeat time on the job is advanced to the next repetition. This option is also available using the 'Go Job' shortcut button.

Interrupt job attempts to terminate a running job by sending it an Interrupt Signal. This option is also available using the 'Interrupt job' shortcut button and by clicking the right mouse button over the job to select it and selecting "kill" from the pop-up menu.

Quit job tries to terminate a running job with a Quit Signal. This option is also available using the 'Quit job' shortcut button.

Other signal attempts to terminate a running job by sending it a specified Signal. This option opens a selection dialog.

The following shortcut buttons are provided on the toolbar.

 Is a shortcut for **Delete job**

 Is a shortcut for **Set runnable**

 Is a shortcut for **Set cancelled**

 Is a shortcut for **Title pri/loadlev**

 Is a shortcut for **Force run**

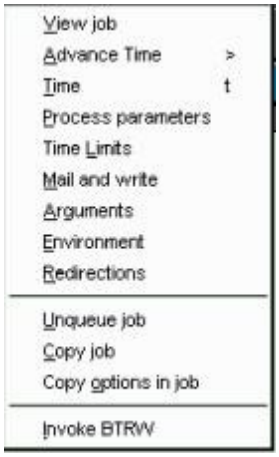
 Is a shortcut for **Go job**

 Is a shortcut for **Interrupt job**

 Is a shortcut for **Quit job**

4.2.4 The Jobs Menu & Buttons

Provides options for inspecting and managing batch jobs.



View job opens a text browser showing the job script to be run. This can also be selected by using the 'View job' shortcut button and by right-clicking the mouse over the job on the job list and selecting the option from the pop-up menu.

Advance Time skips the next scheduled execution of a job by advancing to the next repetition. This option is also available using the 'Advance Time' shortcut button.

Time brings up a dialog for setting the start time, retention options, repetition details and list of days to avoid. This option is also available using the 'Time' shortcut button.

Process parameters brings up the dialog to select the process parameters: working directory, ulimit, umask, network scope and which exit codes represent an error.

Time limits opens the dialog for specifying time restrictions to terminate a runaway job.

Mail and write opens the dialog to specify what notification is required when a job finishes.

Arguments opens a dialog for adding, modifying and deleting arguments that are passed to the job on its command line.

Environment opens a dialog for adding, editing and deleting the environment variables in the jobs run time environment.

Redirections opens the dialog for specifying I/O redirections.

Unqueue removes the job from the queue, putting a copy on the PC hard disk for editing and re-submission using btrw.

Copy job takes a copy of the job from the queue, putting the copy on the PC hard disk for editing and re-submission using btrw. The original job is left on the queue.

Copy options in job copies the options from the job to the defaults used in btrw.

Invoke BTRW opens program btrw.

The following shortcut buttons are provided on the toolbar.

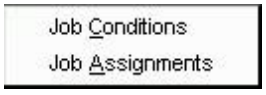
 Is a shortcut for **View Job**

 Is a shortcut for **Advance Time**

 Is a shortcut for **Time**.

4.2.5 The Conditions Menu

Provides options for setting up pre-conditions and assignments.



Job conditions brings up the dialog to add, modify and delete pre-conditions on the selected batch job.

Job assignments brings up the dialog to add, modify and delete assignments for the selected batch job.

Both these options are available using the toolbar buttons shown. Additionally a job may be selected and one these options set by right-clicking the mouse over the relevant job on the job list and selecting from the pop-up menu.

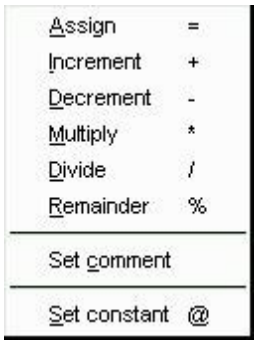
The shortcut buttons are:

 For **Job Conditions**

 For **Job Assignments**

4.2.6 The Variables Menu

Provides options for manipulating variables.



Assign brings up the dialog to modify the data held by the selected variable. This option is also available on the tool-bar button and on the right-click pop-up menu.

Increment increments the value of the selected variable by the currently set constant. This option is also available on the tool-bar button and on the right-click pop-up menu.

Decrement decrements the value of the selected variable by the currently set constant. This option is also available on the tool-bar button and on the right-click pop-up menu.

Multiply multiplies the value of the selected variable by the currently set constant.

Divide divides the value of the selected variable by the currently set constant.

Remainder performs a modulo on the value of the selected variable by the currently set constant.

Set comment brings up the dialog to modify the comment field of the selected variable. This option is also available on the right-click pop-up menu

Set constant for the increment, subtract, multiply, divide and remainder operations. This option is also available on the right-click pop-up menu.

The following shortcut buttons are provided on the toolbar.

Is a shortcut for **Assign**

Is a shortcut for **Increment**

Is a shortcut for **Decrement**

4.2.7 The Search Menu

Both the variable and job lists may be navigated by using search options to find an item of interest.

Search for a specified item or pattern.

Search forward from the current position

Search backward from the current position

4.2.8 The View Menu

Specifies what is visible in the main window of btqw.

Show Toolbar either shows or hides the Toolbar with the shortcut buttons. Selecting this option when the buttons are visible hides the tool bar and vice versa.

Show Status Bar shows or hides the bar at the bottom of the main screen which displays status information. Selecting this option when it is visible hides the status bar and vice versa.

1 Job List 1 etc selects the specified window and brings it to the top if it is obscured by any other windows. There will be an entry in this menu for each of the sub-windows or icons in btqw.

4.2.9 Help

Help for using btqw.

Index brings up the Contents page of the Help.

Using Help gives help on using the Help system.

About displays information, such as release number, about the version of btqw that is running.

Context Help changes the mouse pointer to show a question mark. Clicking on an item in btqw shows the help page for that item.

The following shortcut buttons are provided on the toolbar.

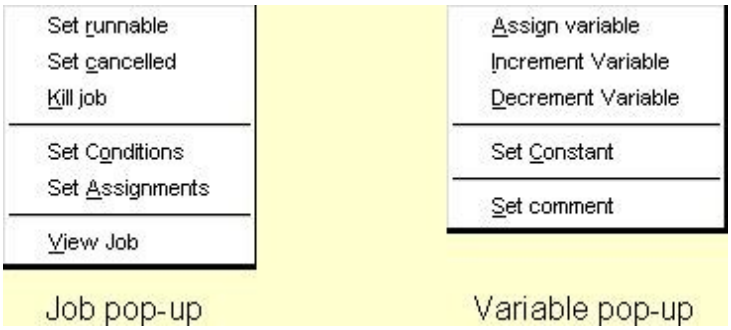
 Shortcut for **Help**

 Shortcut for **Context Help**

4.2.10 Job and variable pop-up menus

Jobs and variables may be selected and various commonly-selected options applied by right-clicking on the job or variable list. This will select the job or variable clicked over

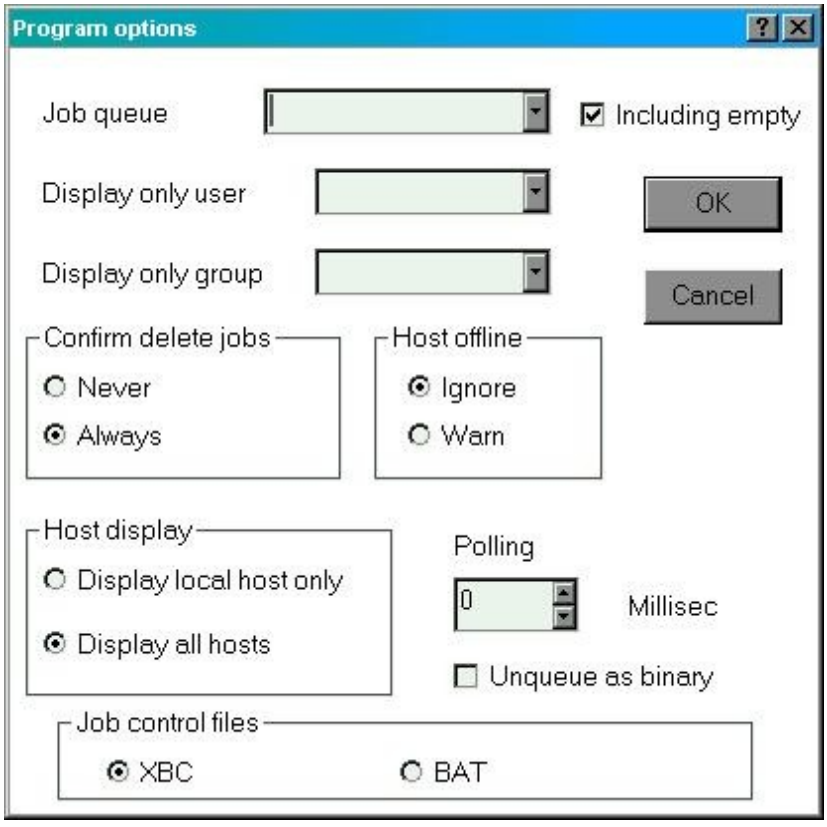
and bring up pop-up menus as shown:



4.3 Setting the Program Options

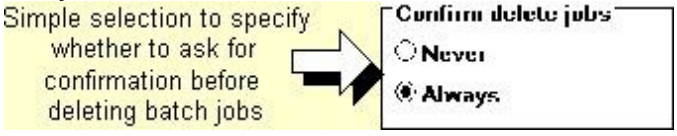
The content and format of information displayed by btqw can be customised via the **View options** item under the Options menu. Confirmation for the delete commands may also be set under this option.

Selecting this option opens the following dialog window.



4.3.1 Setting the Confirmation level

By default btqw asks for confirmation before deleting any job from the queue. This may be relaxed to allow jobs to be deleted without confirmation.

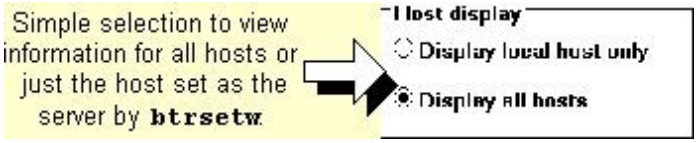


4.3.2 Restricting the display

The display may be restricted by effectively filtering to only show information for selected users, groups, job queues and local or all GNUbatch hosts.

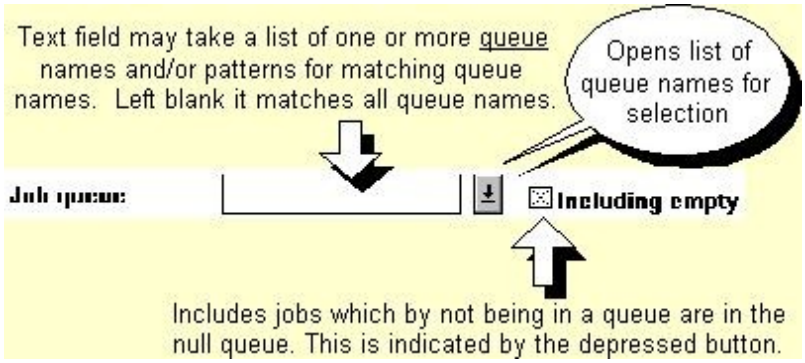
4.3.3 Restricting the display to the local host

All hosts running GNUbatch in the networked mode can be treated as a single system. By default btqw will show all of the externally visible jobs and variables. The view can be restricted to show just the local job queue and variables.



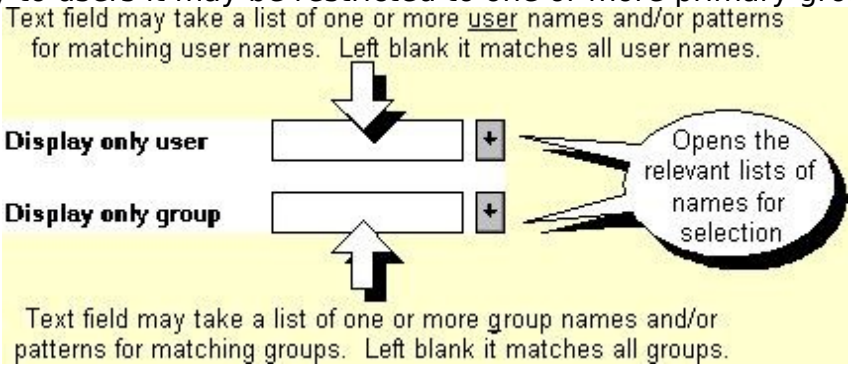
4.3.4 Restricting the display by job queue

The job display can be restricted by selecting a set of job queue names or patterns matching job queues.



4.3.5 Restricting the display by user & group

The display may be restricted to jobs and variables owned by a specified user or set of users. Similarly to users it may be restricted to one or more primary groups.



Sets of users or groups may contain just one name, a list of names or a list of patterns for matching names. The group and user names may be given as a comma-separated

list of alternatives, including the use of shell-style wild-cards. For example

```
fred
jmc,tony,ukops_jmc,ukops_wal
ukops*,ukadmin[1-5]
```

The wild-card options are:

`*`	Matches anything
`?`	Matches one character
`[a-m]`	Matches one character in list or range
`[!n-z]`	Matches one character not in list or range

4.4 Changing the fields displayed and their format.

There is far more information available for both jobs and variables than could be displayed easily in the main window of btqw. Different columns of information may be displayed as required. The field widths and handling of field overflow may also be adjusted, however field widths may more conveniently be adjusted using the title bars.

For example: If your batch jobs often have arguments and long titles, and you do not need the shell column in the job display, then you could remove the shell column and add an argument column:

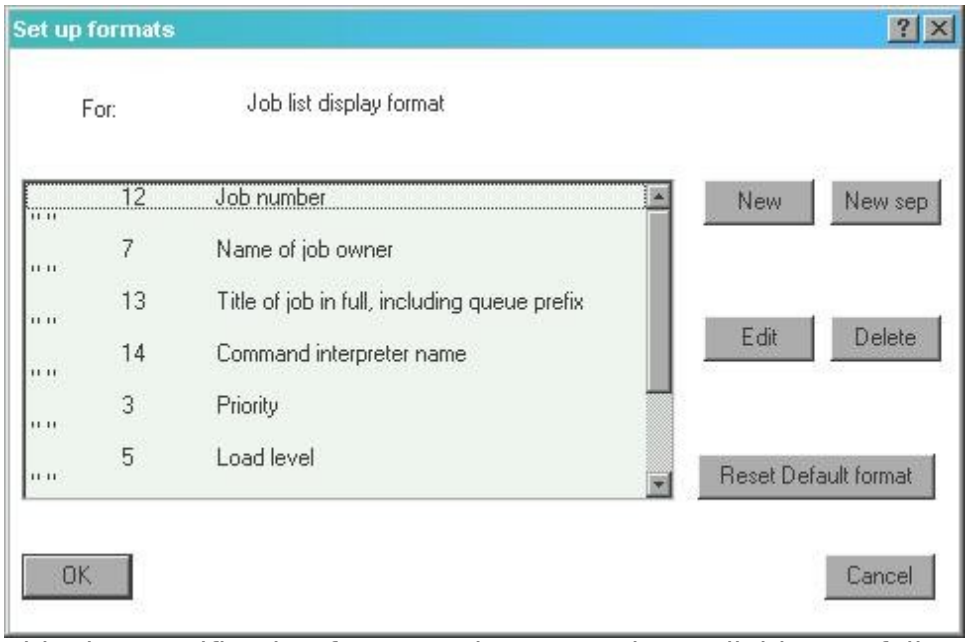
To change the job or variable fields, select the appropriate option from the **Window** menu.

4.4.1 Changing the Job Display

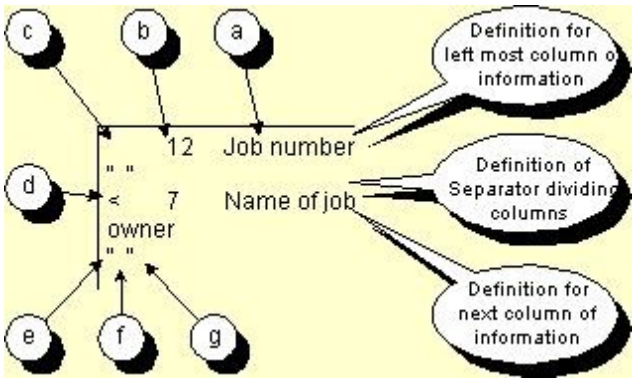
Selecting **Job Display Format** brings up the following window. The scrollable text window shows the display format with each row corresponding to a column of the job list display.

Each column may have a separator, usually one space, but any other character or characters may be chosen.

To change the order of items on the list, drag and drop entries in new positions as required.



Each line holds the specification for one column or column divider, as follows:

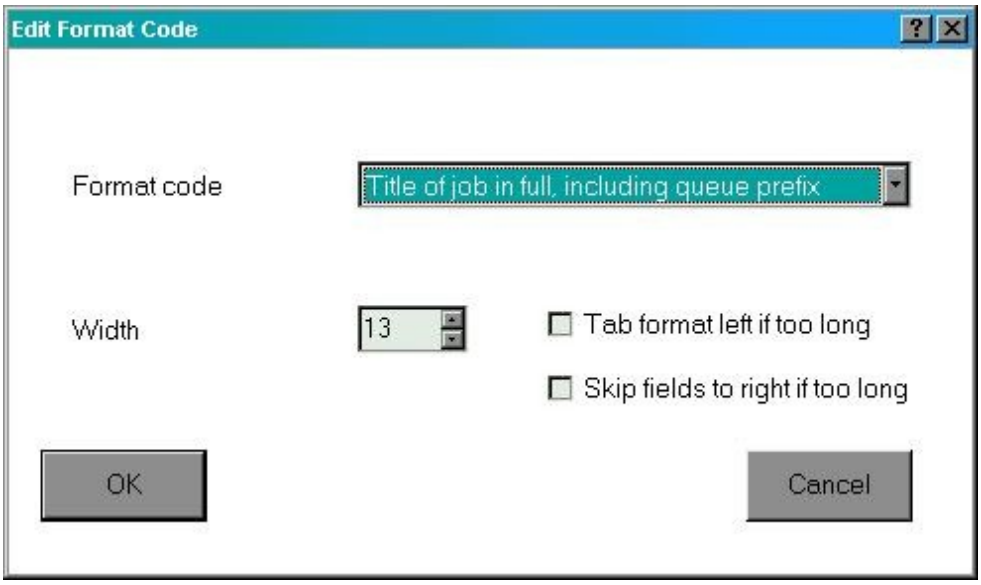


1. Field description / title
2. Width in characters
3. No action on field overflow
4. Overflow onto left hand field is permitted
5. Open quote before separator
6. Separator character(s)
7. Close quote after separator

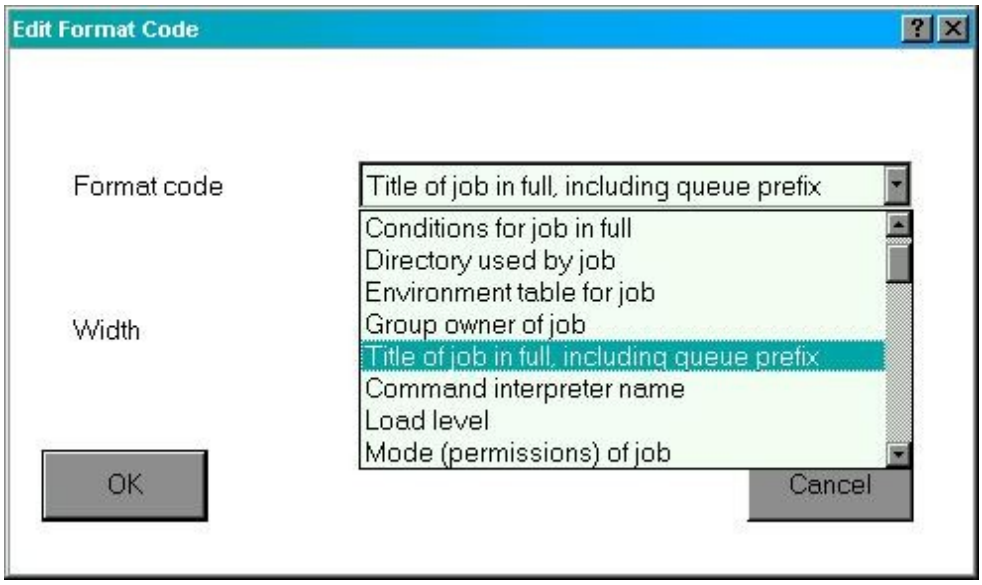
To edit an existing field select the line showing the specification for that field or separator and click on the **Edit** button. Alternatively double-click on the line.

To insert a new field select the line underneath the point at which you want to insert the new one and click on the **New** button, or the **New sep** button for a separator.

New and Edit will bring up a window looking something like this:



The Format code field specifies what information will be displayed in the column. Clicking on the button at the right gives a list of the available codes for selection..



The width is set to a suitable value for that field which can then be adjusted.

The two check boxes "tab format left" and "skip fields to right" provide for what happens if the field is too long to fit. In the first case the preceding column is "sacrificed" to accommodate it. In the second case all following columns are "sacrificed".

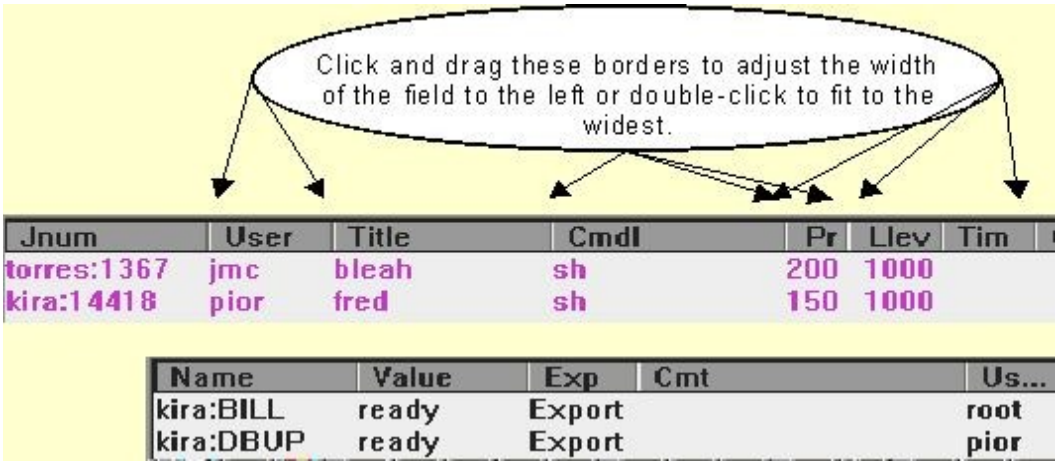
4.4.2 Changing the Variable Display

The variable list format may be adjusted in exactly the same way as for jobs by selecting the **Var Display Fmt** option in the **Window** menu.

4.4.3 Changing the widths from the header bars

If you just need to change the widths rather than the contents, it may be more convenient to use the header bars on the job and variable lists.

At the top of the job and variable lists are the titles, whose borders may be used to adjust the widths of the columns to the left.



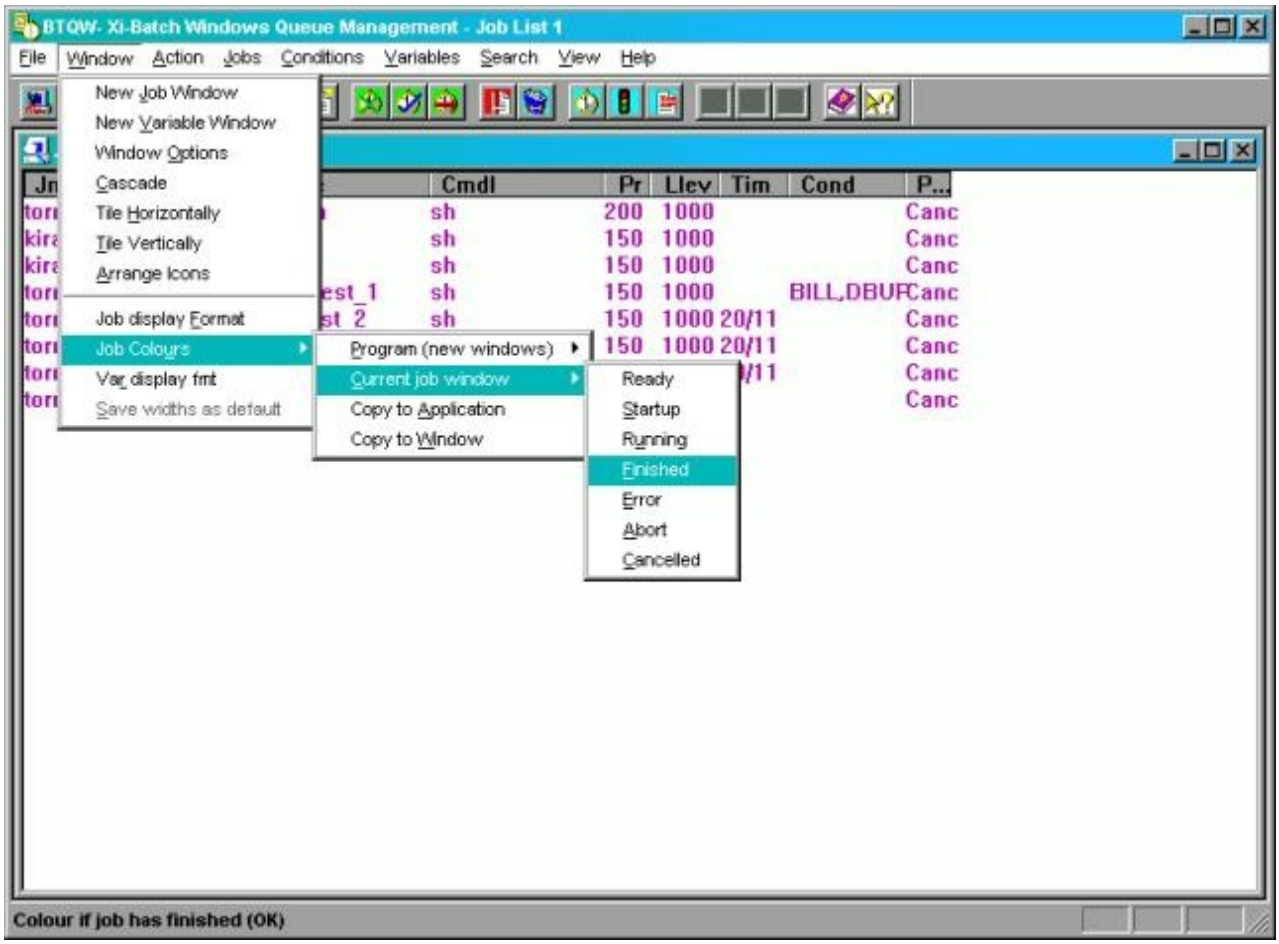
After this has been done, the widths as adjusted may be saved using the **Save widths as default** option on the **Window** menu.

4.4.4 Selecting job status colours.

The job list display can be enhanced by selecting different colours to represent jobs in various states, for example green for running, blue for ready to run, red for error etc.

You can save the current scheme so that subsequent runs of btqw will use them again.

The following illustrates the sub-menus:



The first sub-menu has four items:

Program indicates selection of colours for all new windows. This will be saved when program options are saved from the **Program** menu, but will not affect the currently-displayed job window.

Current job window changes colours for the currently-selected job window without affecting other job windows or the program options.

Copy to application copies the colour selection from the currently-selected job window to the program options.

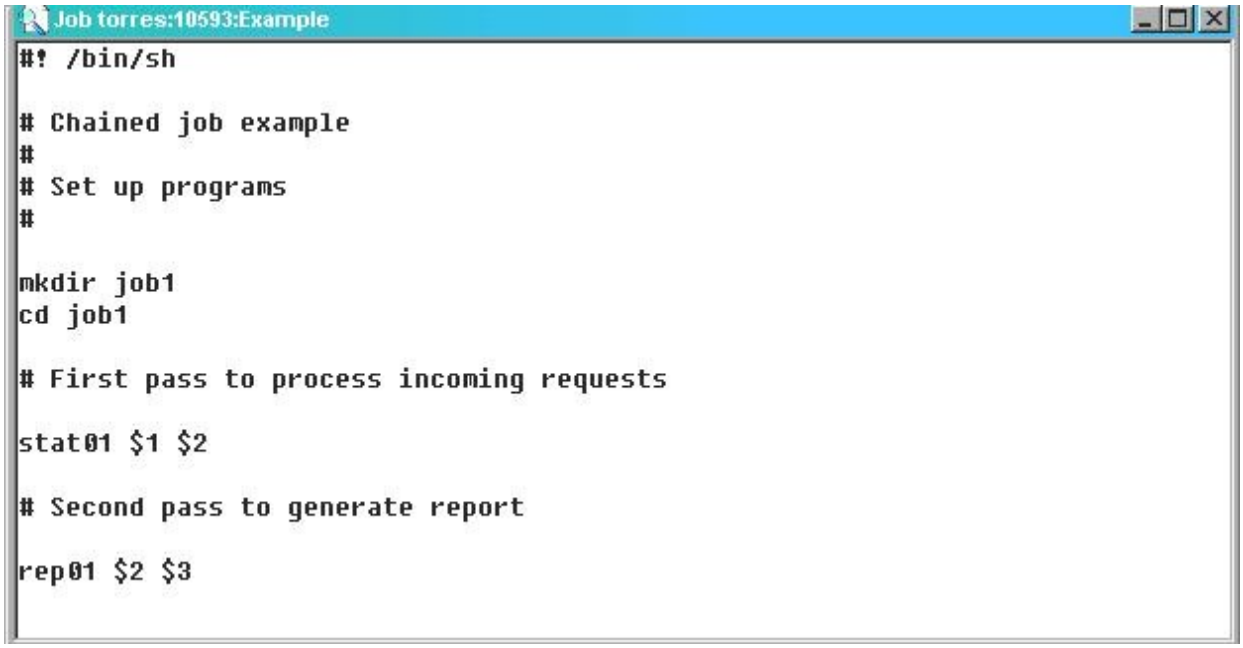
Copy to window restores the colour selection of the currently-selected job window from the program options.

Each entry in the final sub-menu brings up a Windows colour selection palette to choose the desired colour for the text of jobs in that state.

4.5 Viewing a Batch Job

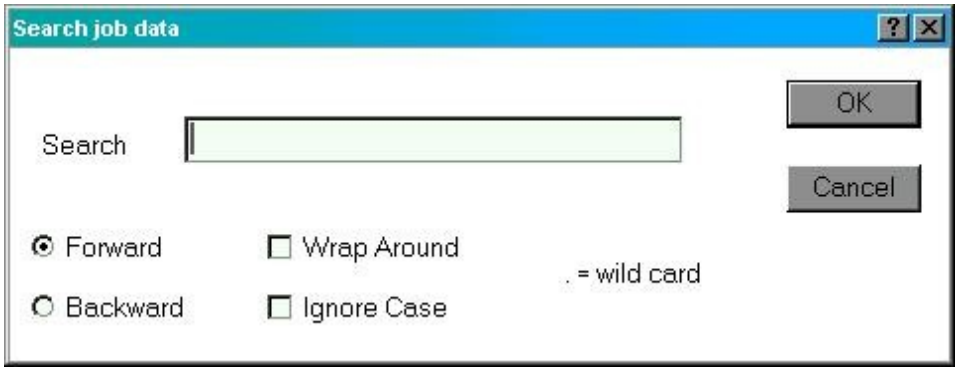
To view the script of a job either select it and the **View Job** option from the **Job** menu or click on the toolbar button.

Alternatively right-click on the job and select **View Job** from the pop-up menu.



If the job is longer or wider than can be seen in the Window, scroll bars will be provided.

Pressing the right mouse button will open the Search job data dialog. This provides options for specifying and finding text strings within the job script thus:



4.6 Changing Job and Variable parameters.

A job may be deleted, changed, reviewed by clicking on the line representing it in the job list and then selecting the required menu option or short cut button. Variables may be operated upon in exactly the same way. Some menu options and short cut buttons will have an immediate effect. Others will open a dialog window for additional information.

Many of the dialogs used by btqw and btrw are the same. They are described along with the parameters that they affect in the chapters on Jobs and Variables.

5 btrw - Job Submission & Editing Tool

Used for creating, editing and submitting batch jobs, btrw is a Windows alternative to the Unix command line program btr.

Btrw can create new batch jobs, submit them to the Unix host and save them on the PC hard disk. It can also edit existing jobs that have been unqueued using btqw.

Saved job files can be "dragged and dropped" into btrw and jobs submitted that way.

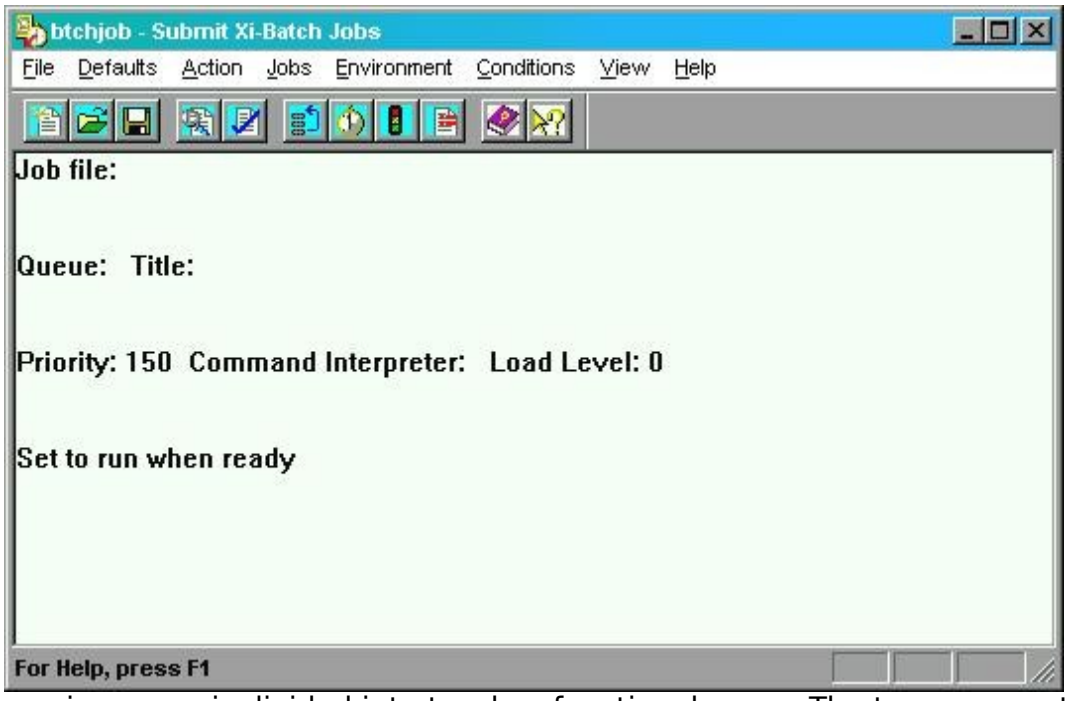
Both btqw and btrw save jobs on the local hard disk in the same "unqueued job" format. This uses two files, which are:

command file
: Which is like a shell script, that holds the specification for the job. This shell script contains statements to reproduce the job environment and a btr command. The btr command has options to set up all of the job parameters and references the *job file* by name. (Note that there is actually no btr command on Windows, but the format is kept consistent with the Unix version).

job file
: which contains the text, or script, of the job that is piped into the "command interpreter" by the scheduler.

5.1 The Main Window

When btrw is invoked the main window will be displayed. By default it will look something like this:



The main screen is divided into two key functional areas. The top area contains menus and short cut buttons for issuing commands. The bottom area displays part of the specification for the batch job which is being worked on.

Btrw uses similar windows and dialogs to btqw for specifying the job options.

To edit job scripts btrw invokes an editor of the user's choice. The editor is specified as a default parameter in the btrw options. On installation the default editor is notepad.

5.2 The Menus and Shortcut Buttons

All commands are performed by selecting a menu option or clicking on the equivalent shortcut button. Some of the menu options may also be selected using shortcut keys, which are indicated to the right of the relevant options in each menu.

5.2.1 The File Menu

For creating, opening and saving jobs. Also for configuring btrw and quitting.

New creates a new job, using the current default settings, ready for editing.

Delete job deletes the current job.

Open brings up a file selector dialog for opening a previously saved job.

Save writes a copy of the job to the PC hard disk.

Save as renames the file and writes a copy of the job to the PC hard disk.

Recent File may show some recently edited job files which may be re-selected by selecting the appropriate name.

Program options sets options like the job queue name and what program to use as an editor for the job file.

Save Defaults writes the default settings to disk for use the next time btrw is run.

User Permissions Shows the user's permissions on the Unix host. **Exit** terminates btrw.

The following shortcut buttons are provided on the toolbar.

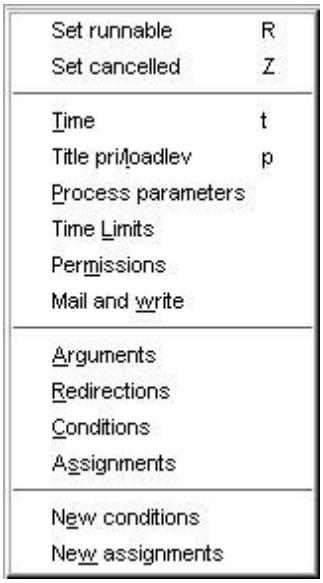
 Is a shortcut for **New**

 Is a shortcut for **Open**

 Is a shortcut for **Save**

5.2.2 The Defaults Menu

For specifying default options for all new jobs being created in this btrw session.



Set runnable state as the default for new jobs.

Set cancelled state as the default for new jobs.

Time Opens the standard job time and repeat specification dialog for setting default values.

Title pri/loadlev Opens the standard "title, priority, command interpreter and load level" dialog for setting default values.

Process params Opens the standard dialog for setting the process parameters: working directory, ulimit, umask, exit code ranges, advance time on error flag.

Time limits for detecting and stopping over-running jobs.

Mail and Write defaults for the job completion flags.

Permissions for job access modes.

Arguments Opens standard dialog for specifying job arguments as defaults.

Redirections Opens standard dialog for specifying job I/O redirections as defaults.

Conditions Opens standard dialog for specifying job conditions as defaults. Use this command to set a default list of conditions for new jobs.

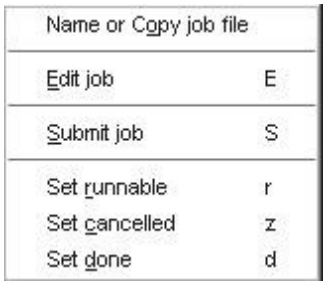
Assignments Opens standard dialog for specifying job assignments as defaults.

New conditions specifies default parameters to apply to any new conditions.

New assignments specifies default parameters to apply to any new assignments.

5.2.3 The Action Menu & Buttons

Provides options for Submitting jobs, creating, editing and deleting job files.



Name or Copy job file opens a file selector dialog for specifying the name of the file to contain the job script.

Edit job opens an editor for working on the job's script.

Submit job to the scheduler.

Set runnable specifies that the job is to be submitted in the runnable state.

Set cancelled specifies that the job is to be submitted in the cancelled (i.e. not runnable) state.

Set done specifies that the job is to be submitted in the done state.

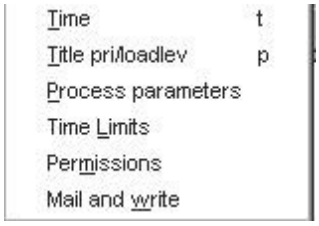
The following shortcut buttons are provided on the toolbar.

 Edit job

 Submit job

5.2.4 The Jobs Menu and Buttons

Menu options for specifying the various job parameters.



Time Opens the standard job time and repeat specification dialog.

Title, pri/loadlev brings up a dialog to specify the Job Title, Priority, Command Interpreter and Load Level.

Process Parameters Opens the standard dialog for setting the process parameters: working directory, ulimit, umask, exit code ranges, advance time on error flag.

Time limits opens dialog to set parameters for detecting and stopping over running jobs.

Permissions for access modes. I.e. read, write, etc.

Mail and write opens dialog for setting job completion mail and write flags.

The following shortcut buttons are provided on the toolbar.

 Is a shortcut for **Title pri/loadlev**

 Is a shortcut for **Time**

5.2.5 The Environment Menu

Menu options for setting up environment variables, command line arguments and I/O redirections for the job.

Environment Default sets the environment of the job to the default values.

Clear Environment removes all environment variables from the job.

Edit Environment opens the dialog for changing the environment variables on the job.

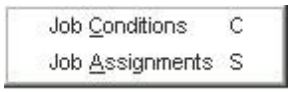
Squeeze Environment removes all items from the job's environment which are the same as the server's default.

Arguments opens the dialog for changing the command line arguments to the job.

Redirections opens the dialog for editing the I/O redirections for the job.

5.2.6 The Conditions Menu

Provides options for setting up pre-conditions and assignments.



Job conditions brings up the dialog to add, modify and delete pre-conditions on the job.

Job assignments brings up the dialog to add, modify and delete assignments for the job.

The following shortcut buttons are provided on the toolbar.

 Is a shortcut for **Job Conditions**

 Is a shortcut for **Job Assignments**

5.2.6.1 The View Menu

Specifies what is visible in the main window of btrw.

Show Toolbar either shows or hides the Toolbar with the shortcut buttons. Selecting this option when the buttons are visible hides the tool bar and vice versa.

Show Status Bar shows or hides the bar at the bottom of the main screen which displays status information. Selecting this option when it is visible hides the status bar and vice versa.

5.2.6.2 The Help Menu

Help for using btrw.

Index brings up the Contents page of the Help.

Using Help gives help on using the Help system.

About displays information, such as release number, about the version of btrw that is running.

Context Help changes the mouse pointer to show a question mark. Clicking on an item in btrw shows the help page for that item.

The following shortcut buttons are provided on the toolbar.

 For **Help**

 For **Context Help**

5.3 Creating a New Job

There are four essential operations required to create a new job. The first three are probably best done in sequence. This avoids btrw generating reminder messages later when it needs information from these operations. The first two operations are selected from the File menu, and the others from the Action menu as follows:

1. Select the **New** option, from the **File** menu. This deletes any existing job details and creates a new one from the default settings.
2. Use the **Save as** option, also from the **File** menu, to specify a name for the command file. A *Windows* file selection dialog will appear showing the contents of the current directory. Select a new file name. An alternative directory for the command file can also be specified in this dialog. All command files have `.XBC` as their extension so that btrw can recognise them.
3. Specify a name for the job file, by selecting the **Name or Copy job file** option, from the **Action** menu. A similar file selector will appear, which is used to enter the file (and possibly path) in the same way as for the command file name. The job file name must have `.XBJ` for an extension.
4. It is now possible to create and edit the job script using the **Edit job** option, from the **Action** menu, to invoke the text editor. The text editor is automatically loaded with the script for editing - in this case a blank file.

There are no constraints on when or how many times the script may be edited, once the first three steps have been done.

When a new job is created it will be given a specification based on whatever default options are currently in force. These can include: queue name, job title and initial state which will be shown on the job list entry. Other settings can only be seen by opening the relevant specification dialogs, for example **Time** from the **Jobs** menu.

5.4 Loading an Unqueued or Previously Saved Job

Use the **Open...** option from the **File** menu to open a previously saved or unqueued job. This opens the standard file selector dialog, in the currently set directory. The file list in the dialog is restricted to show only the Command files of each job. The dialog can be fooled by files which look like but are not valid command files. This is not dangerous, an error message will be displayed and another file can be selected.

Select the required job and click on OK, which loads the job specification into btrw.

5.4.1 Setting up or Editing the Job Specification

Select the job by clicking on its entry in the job list. Any of the parameters in the selected jobs specification can now be edited using the options under the **Jobs** and **Action** menus. These menus have options which are similar to those in btqw.

5.4.2 Editing the Job Script

Select the **Edit job** option from the **File** menu to invoke the text editor. The text editor is automatically loaded with the script for editing. Alternatively click on the **Edit job** short cut button instead of using the menu option.

Change the script, then save the changes and exit as appropriate for the editor. When this is done it is a good idea to save away the command file as well by selecting the **Save** option from the **File** Menu.

5.4.3 Submitting Jobs

Jobs can be submitted by selecting the **Submit job** option from the **File** menu or by clicking on the **Submit job** shortcut button.

Jobs can be submitted, then edited to produce other jobs and submitted again, as many times as required.

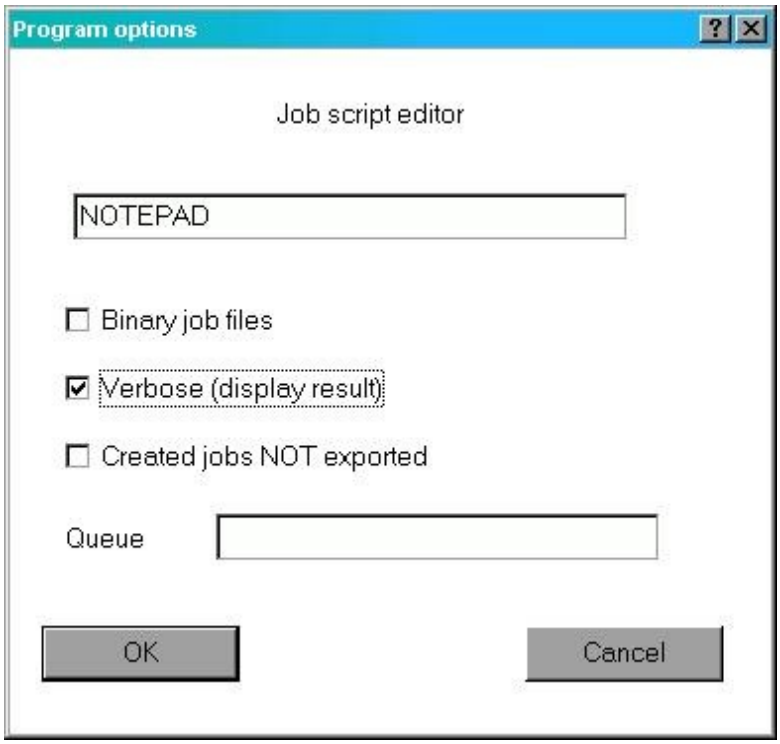
5.4.4 Saving Jobs

Jobs can be saved at any time, using the **Save** option from the **File** menu. This saves the current specification of the job and leaves it open for further work. A new job can be created by using the **Save as ...** option to create a new command file with a different name.

5.5 Selecting a different Text Editor

By default btrw is installed set up to run the notepad editor. This can be changed to any suitable editor of the user's choice.

Select **Program options** from the **File** menu to open the Program options dialog. The editor is specified in the top field of the dialog, which looks like this:



Type in the name of the desired editor, over the top of the existing name. If the new editor is not on the path, then specify the drive letter and full path as well.

5.6 Specifying Defaults

The options under the **Defaults** menu enable default values for the parameters and options to be specified. These provide the default settings for the options in the **Action**, **Jobs**, **Environment** and **Conditions** menu options.

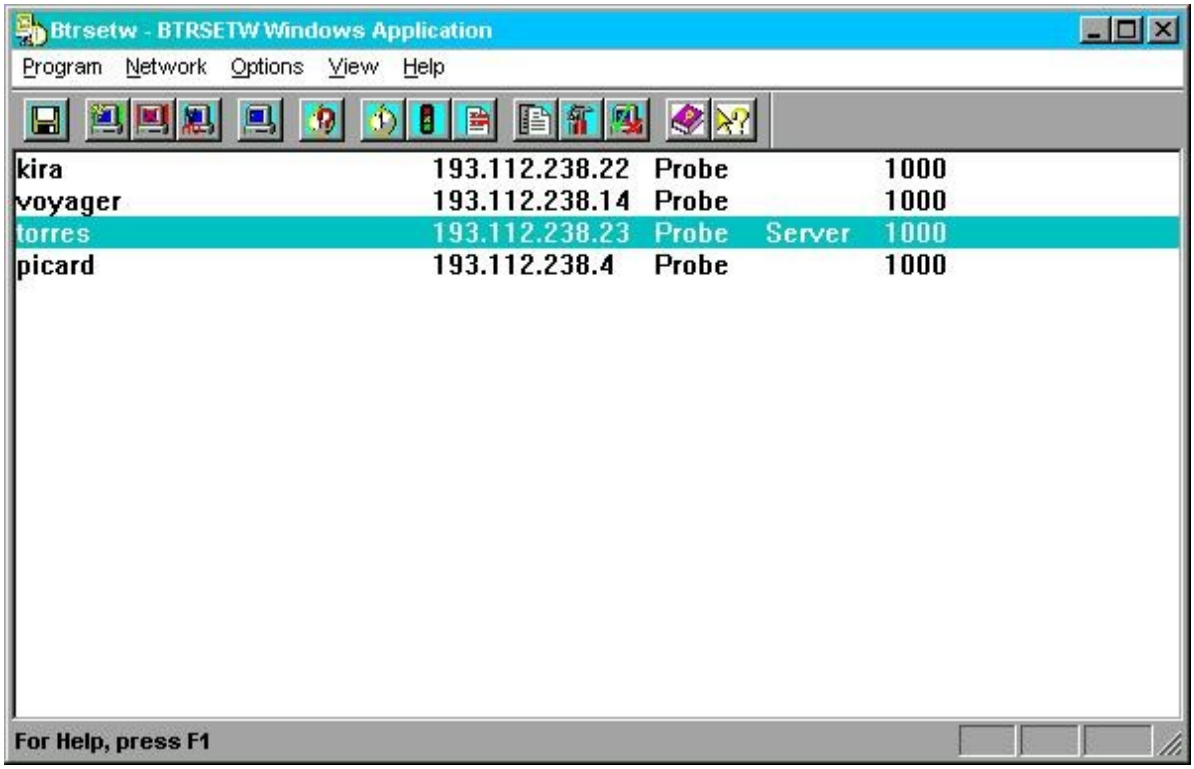
Changes to the default settings can be saved using the **Save Defaults** options from the **File** menu. They will then be loaded as the default settings next time btrw is run.

6 btrsetw - Set Up Tool

Btrsetw is a configuration tool for specifying which Unix hosts btqw will see, which host btrw will submit jobs to and what TCP sockets will be used. It is usually invoked from the Windows Program Manager, by clicking on the Btrsetw icon shown above.

6.1 The Main Window

When btrsetw is invoked the main window will be displayed, looking something like this:



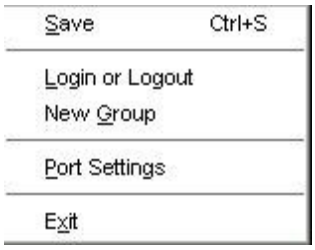
The main screen is divided into two key functional areas. The top area contains menus and short cut buttons for issuing commands. The bottom area displays the list of Unix hosts that can be managed using btqw and which host is the server for btrw.

6.2 The Menus and Shortcut Buttons

All commands are performed by selecting a menu option or clicking on the equivalent shortcut button. Some of the menu options may also be selected using shortcut keys, which are indicated to the right of the relevant options in each menu.

6.2.1 The Program Menu

For saving the host configuration, port settings and quitting from btrsetw.



Save the details of which hosts to connect to and which one is the server for btrw. Also saves the port settings.

Login or Logout optionally "logs in" as a different user to the default user with the server. This may be necessary in all cases if a password is required.

New Group selects a new primary group from the list of groups available to the user.

Port Settings opens a dialog for specifying the TCP socket numbers to be used for each port. This option is only needed if the default values have not been used on the Unix hosts.

Exit terminates btrsetw. If any the settings or hosts have been changed but not saved this option asks if it should save them before exiting.

The following shortcut button is provided on the toolbar.

 Is a shortcut for **Save**

6.2.2 The Network Menu

For configuring connections to one or more Unix hosts, which are running GNUbatch.

Add new host Opens the dialog for specifying a connection to a Unix host.

Delete host removes the selected Unix host from the list of machines to connect to.

Change host Opens the same dialog for specifying a connection to a Unix host as the **Add new host** option. In this case the details of the selected job are displayed for editing.

Set as Server nominates the selected Unix host as the server which btrw will submit jobs to. There must always be one host Set as Server.

The following shortcut buttons are provided on the toolbar.

 Is a shortcut for **Add new host**

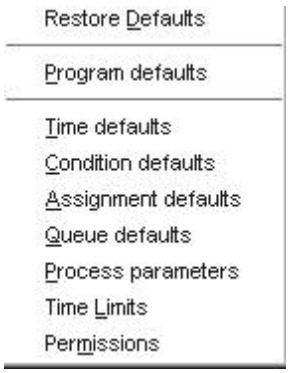
 Is a shortcut for **Delete host**

 Is a shortcut for **Change host**

 Is a shortcut for **Set Server**

6.2.3 The Options Menu

For specifying default options for all new jobs being created with btrw. These facilities are now incorporated in btrw for setting the defaults as well with extra facilities.



The following shortcut buttons are provided on the toolbar.

 Program defaults

 Time defaults

 Condition defaults

 Assignment defaults

 Title priority load level

 Process Parameters

 Permissions

6.2.4 The View Menu

Specifies what toolbars are visible in btrsetw.

Show Toolbar either shows or hides the Toolbar with the shortcut buttons. Selecting this option when the buttons are visible hides the tool bar and vice versa.

Show Status Bar shows or hides the bar at the bottom of the main screen which

displays status information. Selecting this option when it is visible hides the status bar and vice versa.

6.2.5 The Help Menu

Help for using btrsetw.

Index brings up the Contents page of the Help.

Using Help gives help on using the Help system.

About displays information, such as release number, about the version of btrsetw that is running.

Context Help changes the mouse pointer to show a question mark. Clicking on an item in btrsetw shows the help page for that item.

The following shortcut buttons are provided on the toolbar.

 For **Help**

 For **Context Help**

6.3 Setting up a Host and making it the Server

There are three operations required to create a new host and set it as the server for btrw, which should be done in sequence.

1. Select the **Add new host** option, from the **Network** menu. This opens a simple dialog for entry of the host parameters.
2. Enter the host name, make sure that the probe check box is "checked" and press the OK button.
3. Select the line with the new host definition and then select the **Set as Server** option, also from the **Network** menu.
4. Save the new definition using the **Save** option from the **Program** menu.

When setting up several hosts it is good practice to save between setting up each entry.

6.4 Making an existing host the Server

Perform steps 2 and 3 from the above example. Step 2 will deselect the existing server and mark the new host as the server.

6.5 Setting up a Host for management by btqw

Perform steps 1 and 3 from the above example.

7 Jobs and Related Entities

To execute a job GNUbatch invokes the specified command interpreter and pipes the text of the job to the standard input of the command interpreter. The most common types of batch job are shell scripts. Any program which will read instructions from standard input may be set up as a command interpreter for use by GNUbatch.

The text of a job is often referred to as the job, job file or commands. To avoid confusion the use of the word *script* is now being encouraged. The script for each batch job may invoke other programs, compiled or shell script, as it would if it was run from the command line. The term job file is still used to describe the file of an unqueued job which holds the job script.

The set of parameters held by GNUbatch governing what it should do with the job is often called the command file. This is now being referred to as the job specification. The term command file is used to refer to the file of an unqueued job which holds the specification.

Apart from variables, which are described in their own chapter, jobs also have relationships with three other entities. These are:

- The command interpreter under which the job actually runs. All jobs have a specified command interpreter.
- A queue, which provides a grouping mechanism for jobs. All jobs belong to a queue. This is not always obvious since jobs which do not specify a queue are associated with the null queue, which has no name.
- An optional relationship, for repeating jobs only, with the system wide holiday file.

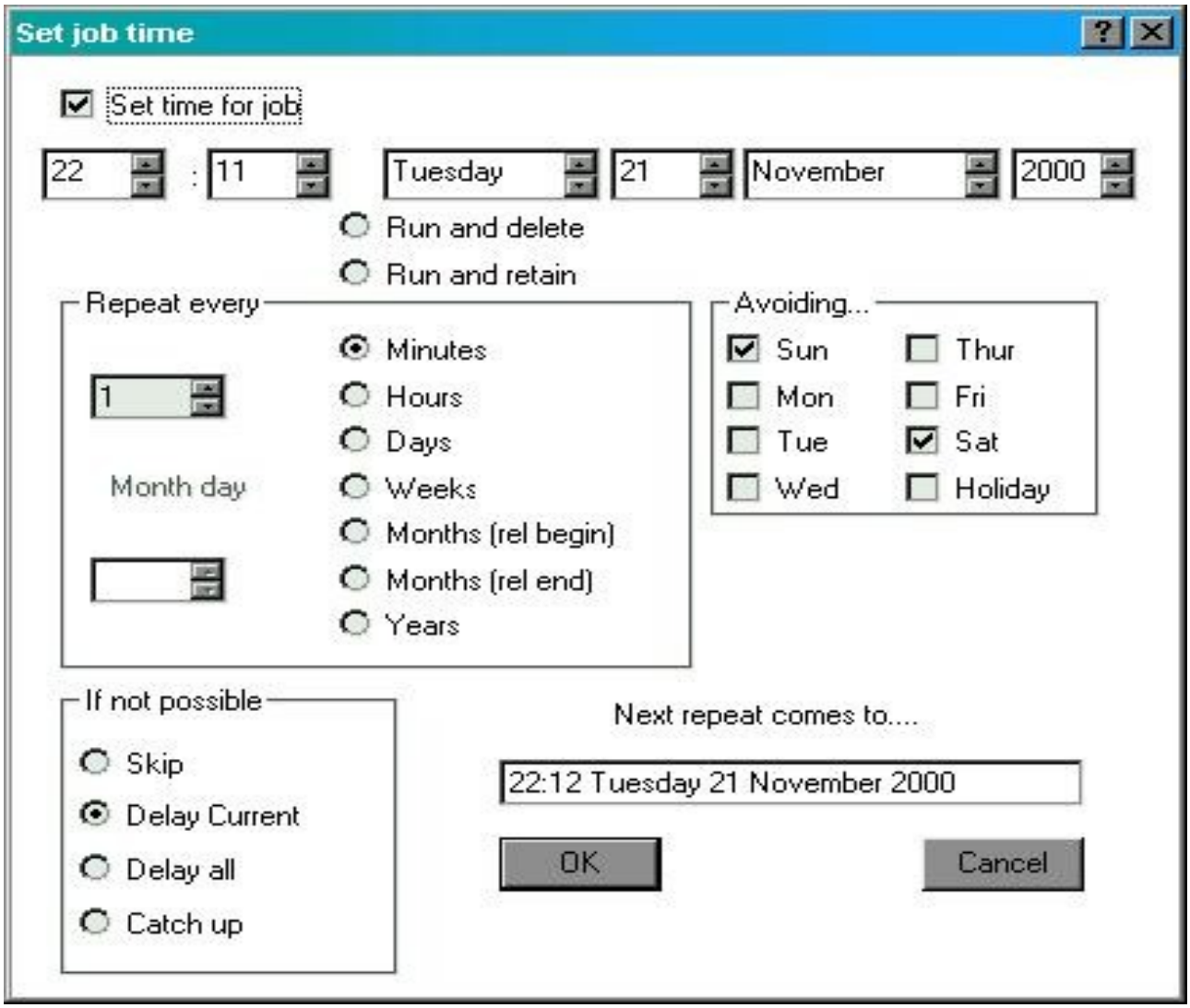
These entities are discussed at the end of this chapter, as well as in the sub-sections which describe how each job specifies relationships with them.

7.1 Time

Jobs can be specified without any scheduling time specifications. In this case they will run as soon as possible, just like jobs run under the standard Unix batch command. Once such a job has run it will be deleted from the queue.

Repeating jobs have additional options, such as a specification of any days to avoid, which do not have any impact on the first *Scheduled Run Start Time*.

Most of the time parameters are specified via a dialog which looks like this:



checked and the job has a time as shown above.

7.1.1 Scheduled Run Start Time

The time at which a job is scheduled to start can be specified by date and time to the nearest minute. GNUbatch starts jobs on the minute boundary, unless they are prevented from doing so by some condition. When a job has a time specified, it may be set up to be deleted, retained in a done state or repeated, automatically after the first run.

The time is specified using the line of boxes that look like this:

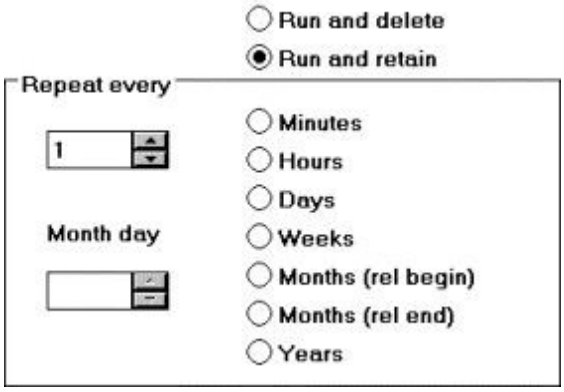
This example shows a time of 11:18 a.m. on Tuesday 3rd. September 1996. The hours box uses the 24 hour clock. To change a time click on the arrows next to each value. Pressing the up arrow brings the job forward and the down arrow moves it back to run later. The programs will not let a time that has already passed be specified.

If a job is due to run for the first time then it will always wait, if blocked by some condition, until that condition is satisfied. Once all conditions are satisfied the job will

run immediately.

When a job is blocked from repeating by some condition, there are a range of behaviours that it can follow. The options for these behaviours are described in the sub-section on Repetition.

7.1.2 Retention



Jobs may be set up to run once at a specified start time and then be retained on the queue after execution. Once the job has run its progress state is set to *Done*.

A job that is in the *Done* state may be set running at any time by a suitably authorised user or program. Similarly the specification of the job may be changed. For example a new run could be scheduled and possibly some repetition specified.

Retention and repetition options are specified using this part of the Time dialog.

The **Run and retain** option is selected in this example, as shown by the "filled in" button.

If the **Run and delete** option was selected the job would be deleted as soon as it finished.

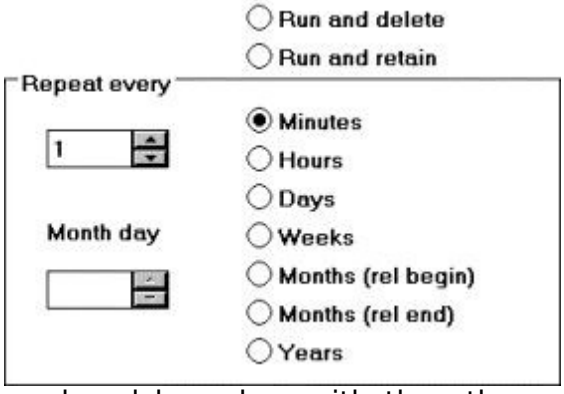
7.1.3 Auto delete after execution

Remember that you can set an auto delete time for jobs that have remained on the queue for some time after they last ran or were submitted. See below for how to set this.

7.1.4 Repetition

Jobs can be specified to repeat at regular intervals after the initial run. The interval is specified as an integer number of a particular unit of time.

Repeat times are specified in the same part of the time dialog as the retention and repetition options.

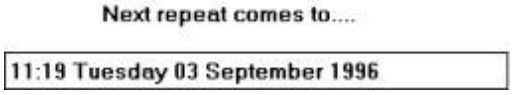


The repeat options are enclosed by a box with the other parameters for repeating jobs.

In this example the minutes option is selected and the **Repeat every** box has a value of 1. This specifies that the job is set to be run every minute.

If the repeat interval is specified in months the lower box will also contain a number. For **Months (rel begin)** the second value is the day of the month on which to run. For example a month day of 5 would indicate run on the 5th of the Month.

For **Months (rel end)** the second value is the day to run on with respect to the end of the month. For example if the number was 30 for a first run in March the job would next run on the 29th of April.

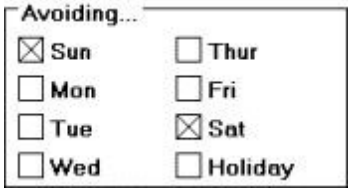


The time of this next run is shown at the bottom of the dialog. For the job that repeats every minute it looks like this:

Repeating jobs have additional options which can be set to indicate what to do if a job fails and to specify any days to avoid.

7.1.5 Days to Avoid

The repeat specification has options to specify one or more days of the week and holidays to be avoided when scheduling the next run of a job. The holidays are marked in the holiday table, which is set up on the Unix host. Up to 6 days of the week, possibly plus holidays, can be set to be avoided.



When the next repetition of a job is calculated the scheduler will step past any days to avoid. For example a job that runs at 3 minutes past the hour every hour, but avoiding Saturdays and Sundays will run at 23:03 on Friday night and then next at 3 minutes

past midnight on Monday morning.

The check boxes in the Avoiding box specify which days are to be avoided. A check mark in a box indicates that the day is to be avoided.

In this example the job will not run on Saturdays or Sundays.

The days to avoid parameter does not affect the initial run time. Hence a job can be submitted to run the first time on a Saturday, but avoid Saturday and Sunday thereafter.

7.1.6 Time adjustments on error

The time adjustment parameter specifies whether the job's scheduled start time should be left in the past or set to the next repetition in the event of the job failing. This parameter is set from the Process parameters dialog.

7.1.7 Action if Job Blocked

If a repeating job is prevented from running by an unsatisfied condition GNUbatch can follow one of four courses of action. These are:

1. **Skip** which bypasses this scheduled run of the job. The time will be advanced to the next repeat.
2. **Delay Current** waits until the relevant variable satisfies the condition, then start the job. If more than one scheduled start time is passed whilst waiting to start the job, all of the runs are delayed. For example a job that repeats every minute, that is blocked for 5 minutes, will be run 5 times in succession when unblocked.
3. **Delay all** is like Delay Current except that it only runs one copy of the job. It also resets the next repeat to be based upon the actual start time of the job. For example; if a job was run at 15 minutes past each hour but was then delayed for 10 minutes, the next run will be set for 25 minutes past the hour.
4. **Catch up** is like Delay Current but only one run is done if several are missed.

The required action is specified using the buttons in the **If not possible** box, which looks like this:

This refers to any reason for delay, which might include:

- Job does not meet conditions.

- Scheduler not running.
- Would exceed maximum load level.

7.2 Job Completion Messages

GNUbatch can send messages to the owner of a batch job, for example when it finishes or fails. These messages can be directed to e-mail, the users terminal session if logged in, or disabled as part of each jobs specification. The options are:

- Discard all messages.
- Write messages to the job owner's terminal, if they are logged in. Otherwise e-mail the messages back to them.
- E-mail messages to the job owner.

These can be set using the **Mail and Write** option under the Jobs menus of btqw or btrw. This option opens a dialog which looks like this.

The options use check boxes to select none, either or both.

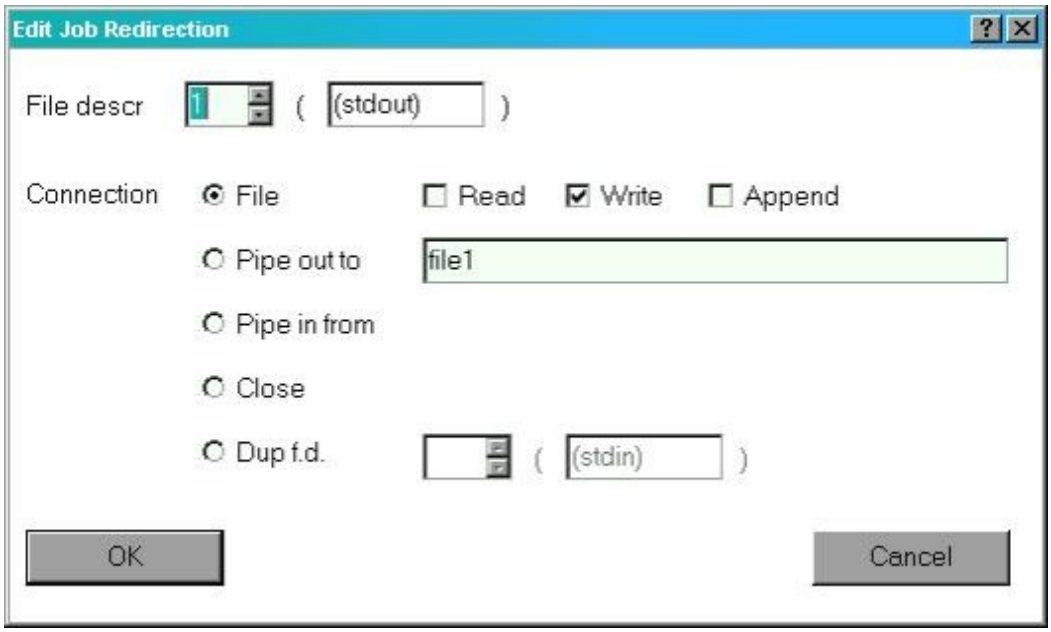
Do not confuse messages from the scheduler about a job with the output from the job. which is handled differently (although in many cases the output is appended unless redirected).

7.3 Redirection of Input and Output

Job redirections can be specified or edited using a dialog which looks like the following. The large box is a scrollable list of all the redirections currently specified.

The order of redirections can be changed by dragging and dropping the relevant lines.

To edit an existing redirection select the relevant line and click on the **Edit** button or double-click the line. This will open another dialog for specifying the details, which looks like this:



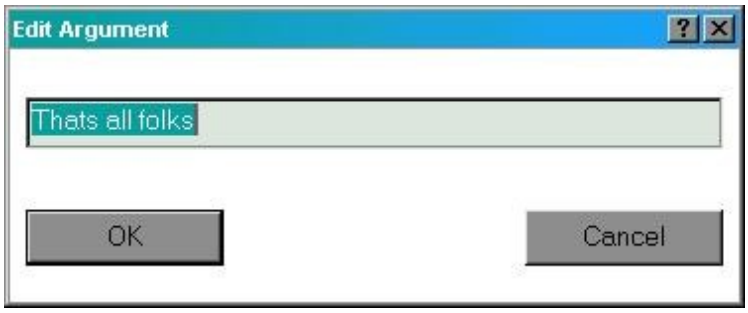
The **new** button opens the same dialog with default settings for editing.

7.4 Arguments

Job arguments can be edited, created and removed using dialogs similar to those for redirections. The main dialog will look like this:

The order of arguments can be changed by dragging and dropping the relevant lines.

The dialog for editing and creating arguments is much simpler than the one for redirections, since it just requires a plain text string. here is what it would look like when editing the first argument, `Thats all folks.`



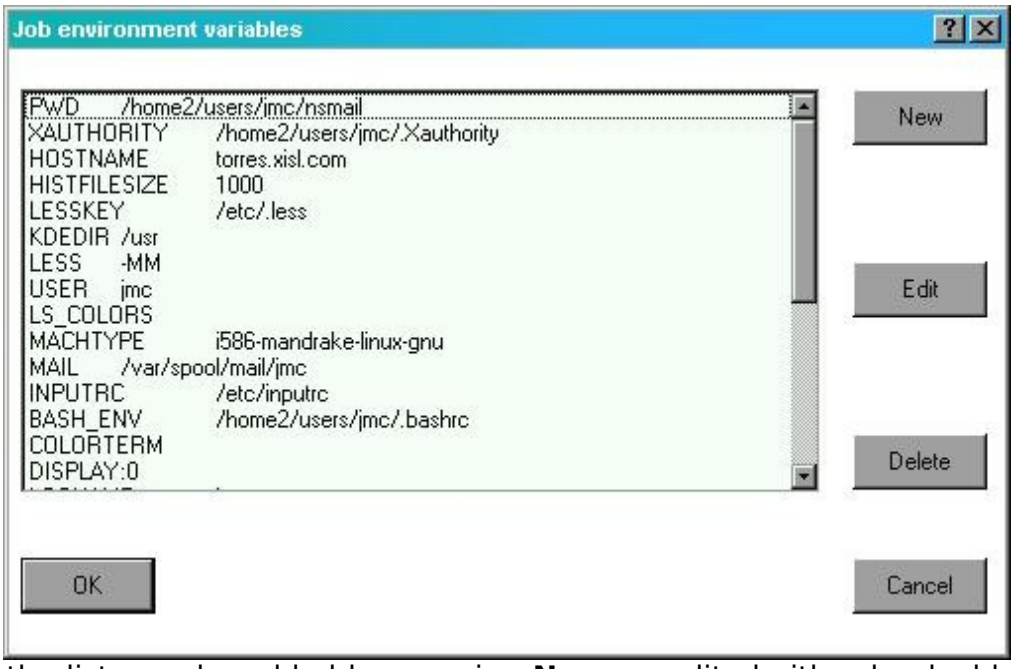
7.5 Environment Variables

The job specification holds a list of environment variables that are set up in the job's environment each time it is run. At run time the scheduler first sets up any environment variables that are specified in the `/etc/Xibatch-env` file. The variables from the job specification are then added to the environment.

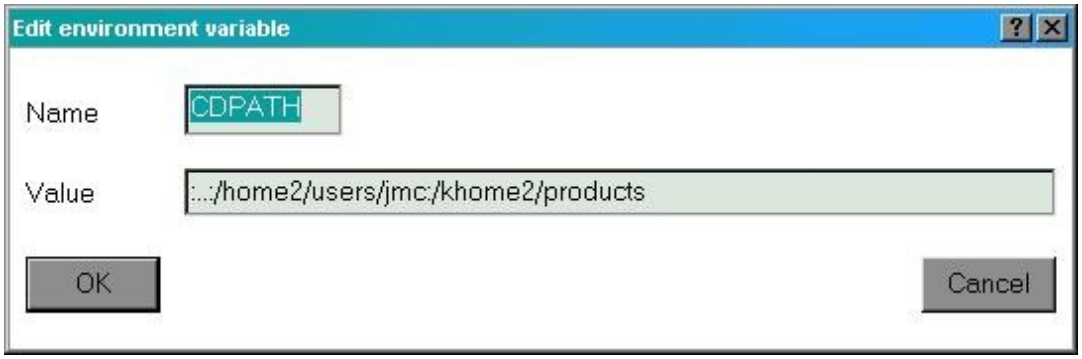
Jobs submitted from a PC running Windows, will be given a default environment. The default can be configured on each PC.

Symbols for meta-data can also be included in the values of environment variables. See the section on Meta-Data for a list of the available data.

The dialog for editing Environment Variables looks like this:



Items on the list may be added by pressing **New**, or edited either by double-clicking, or by selecting the item and pressing **Edit**. The resulting dialog looks like this:



The name and the value may be edited as required. Press **OK** to save.

7.6 Meta-Data

There are several useful parameters from the job specification that can be substituted into arguments, environment variables and I/O redirections. These are:

`%s` Command Interpreter name
`%t` Job title
`%U` User name
`%G` Group name
`%N0` Host name where job originated
`%d1` Job number, in decimal
`%d2` Priority, in decimal
`%d3` Load Level, in decimal
`%x1` Job number, in hexadecimal

%x2 Priority, in hexadecimal
%x3 Load Level, in hexadecimal
%% To insert a single % character put an extra % character in like this %%.

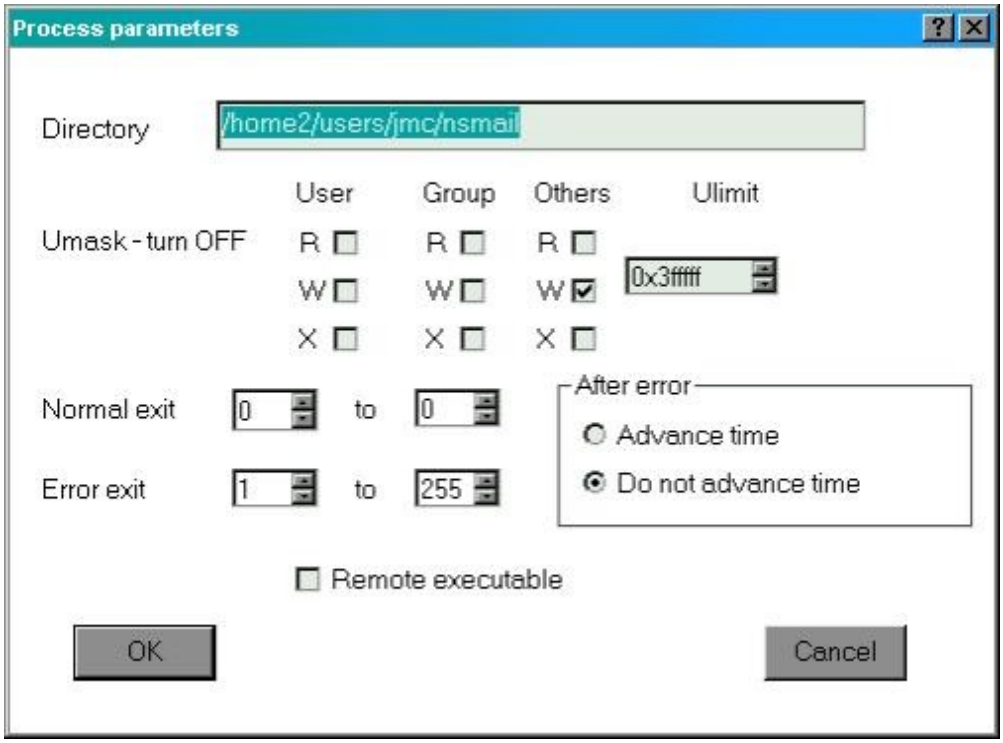
The substitution is performed at run time, making sure that the information is up to date.

7.7 Process Parameters

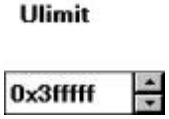
A copy of the complete environment in effect when jobs are submitted is saved as part of their specification. The job will be run using this environment. The various elements of this environment can be re-specified as required.

Jobs submitted from a PC running Windows, will be given a default environment. The default can be configured and overridden on each PC.

The process parameters dialog, contains a diverse set of parameters which are explained later in this section: It looks like this:



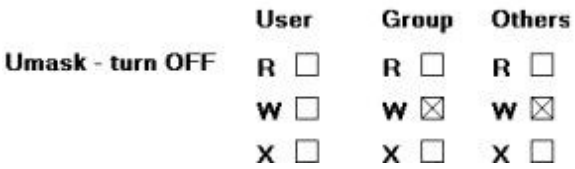
7.7.1 Ulimit and Umask



The *ulimit* and *umask* parameters may be set and passed to a batch job.

ulimit specifies the maximum file size, in blocks, that can be written by the job. It is

usually displayed as a hexadecimal number. For example if ulimit=3FFF he maximum file size would be 16K bytes.



umask affects the default permissions of files created by the job. It is usually represented as an octal number, the same as file permissions. For example if umask=022 write permission will be turned off for Group and Other on any files created by the job.

This dialog represents this more conveniently as check boxes. The boxes are filled to indicate the permissions to be turned off.

7.7.2 Working Directory

By default GNUbatch assumes that a job is to be run in the same directory as it was submitted from. This is held in the job specification, hence any alternative directory may be specified. Take care not to specify a directory which does not exist or for which the owner of the job has insufficient permissions.

Jobs submitted by btrw will use the user's home directory on the server by default, unless this is overridden.

7.7.3 Normal and Error Exit Codes

The ranges of exit code for normal and error exit may be set on the process parameter dialog thus:

for the normal exit range and

for the error exit range.

Remember that the ranges may overlap, and the smaller range will take priority in this case. Any unspecified exit codes will be taken as *abort*.

7.7.4 Network Scope

When two or more machines are running GNUbatch in co-operation with each other the scope of jobs becomes relevant. There are three alternatives, which are:

Local
: Specifies that the job is visible and accessible only on the machine to which it was submitted.

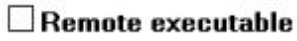
Export

Specifies that the job must run on the local host, but allows the job to be seen and managed from any networked GNUbatch host.

Full Export

Enables the job to run on any co-operating GNUbatch host as well as being visible and manageable by the remote hosts.

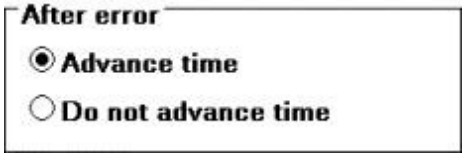
For jobs that are visible to btqw or submitted by btrw only the **Export** and **Full Export** options are relevant. This is because any local jobs can only be seen by the Unix host that owns them. A simple check box, labelled Remote executable, is used to specify which of these two options is selected:



Like this the job can be seen from any machine but must run on the host that owns it. If the box were checked the job could be run on any co-operating GNUbatch host.

7.7.5 Time adjustments on error

The time adjustment parameter specifies whether the job's scheduled start time should be left in the past or set to the next repetition in the event of the job failing.



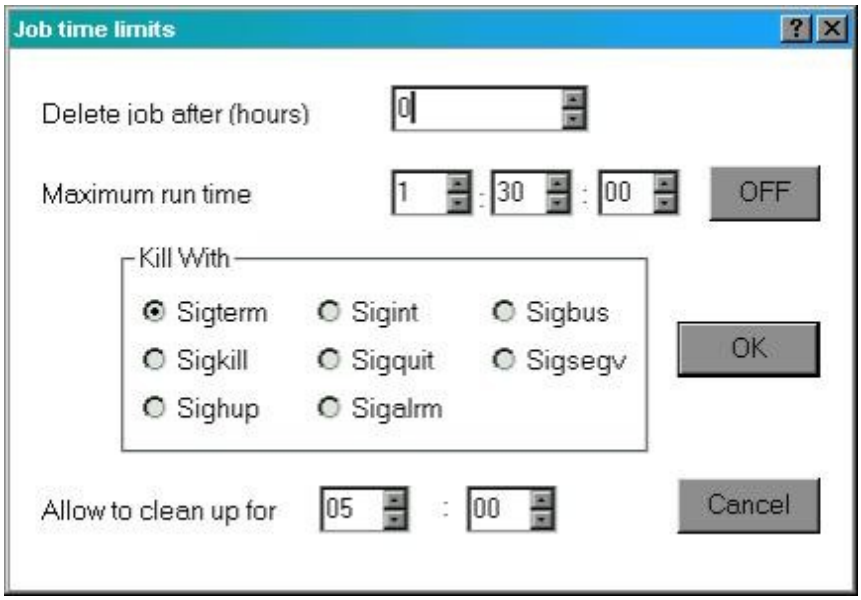
Specify the option by clicking on the relevant button in the **After error** box. Which looks like this:

Advance time is the original default option.

Specifying that the start time is not to be advanced to the next repetition, allows errors to be corrected and the job restarted. Select the advance time on error option, when a problem can or need not be rectified until the next repetition is due.

7.8 Time Limits

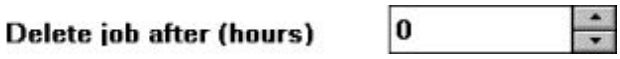
There are limits that can be specified for how long a job may run for and how long it should stay on the queue after running if no repeat is specified. These are set up using the Job time limits dialog, which looks like this:



7.8.1 Auto delete after execution

This is an option, which specifies an automatic delete time after the last run of a job if it has not been run again, either automatically or by manual setting. The default is zero hours, which will retain the job indefinitely.

The delete time is specified in the box at the top of the dialog which looks like this:

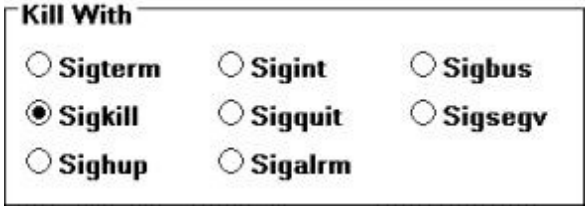


7.8.2 Time-out parameters for stopping Runaway Jobs

There are three parameters that specify how to identify and stop a runaway job. They are:

1. The maximum elapsed time (in hours, minutes and seconds) since starting that a job may run for until it is terminated by the scheduler. The default is 0 seconds which looks like this

2. What signal to send an over-running job in order to terminate it. The job should trap anything other than a SIGKILL and respond by tidying up and exiting cleanly.

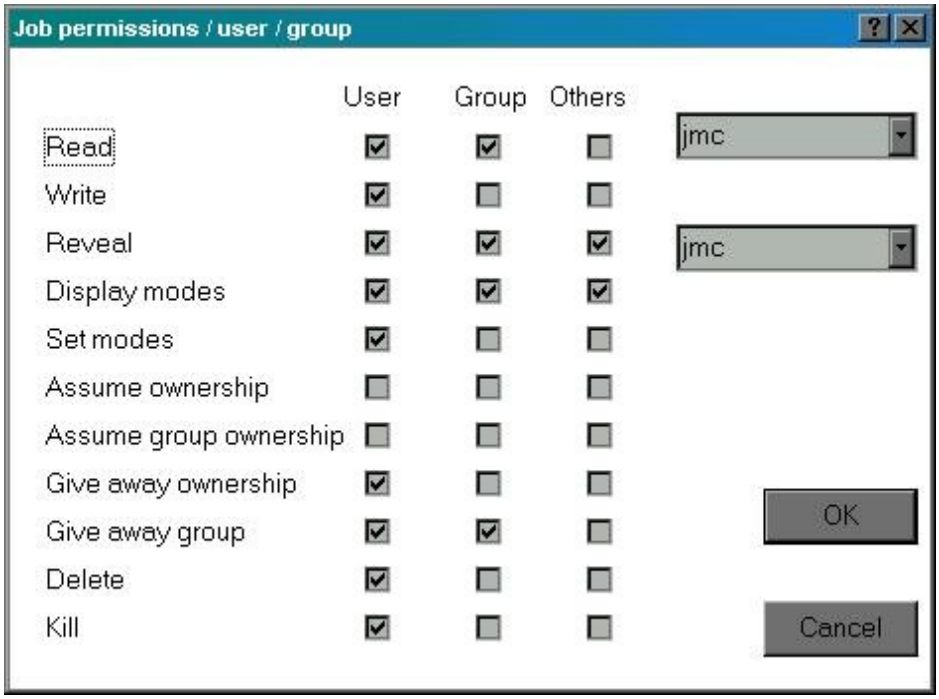
   The signal is specified by name in the **Kill With** box.

3. A grace time, in minutes and seconds, within which the job should terminate

after being sent a signal. If the job does not terminate itself within the specified grace time the scheduler will kill it with a SIGKILL. The default is 0 seconds which looks like this:

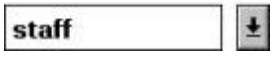
7.9 Owners, Groups and Modes

Each job belongs to a user and a Unix group. Access to jobs is controlled by a set of permissions, called modes, similar to those on ordinary Unix files. These can be seen and specified using the Job permissions dialog, which looks like this:



7.9.1 Owners and Groups

The job specification includes the user who owns the job and the Unix group that the job belongs to. The owner and the group are normally taken as those of the user who submitted the job. A different user and group may be specified when the job is submitted, but only if the submitting user has write admin file privilege.



Suitably authorised users may specify the owner and group of a job using this part of the permissions dialog.

An administrator may do this in one operation. Ordinary users, may be given sufficient privilege to change the specification. In this case the current owner has to specify who the job is to be given to and then the recipient must accept it.

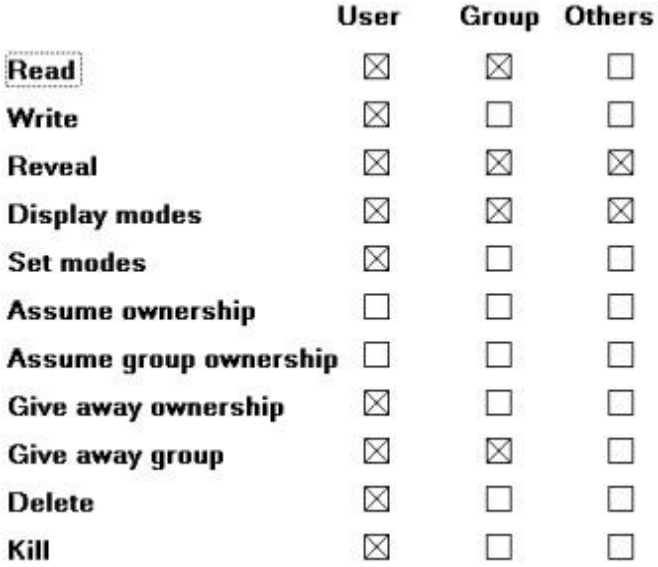
These security features prevent un-authorised transfer of jobs to and from more privileged owners and groups such as the root user.

Only the primary groups of users are considered for evaluating access permissions to jobs.

7.9.2 Modes

Access to jobs is controlled by the Modes which are similar to Unix file permissions, but with greater functionality. Permission to each access mode is granted to the owner of the variable (User), users in the same primary group (Group) or everyone (Others).

The modes take up most of the permissions dialog. These are what the default modes usually look like when GNUbatch is installed:

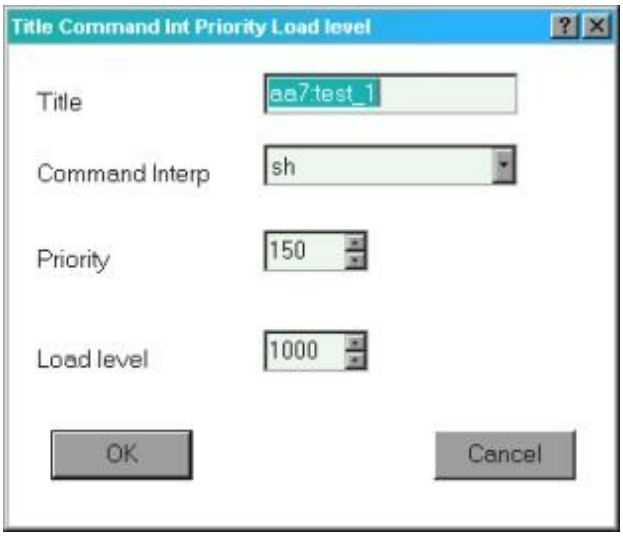


7.10 Job Identifiers - Queues, Titles and Job ID numbers

Each job has a unique job id number, also called the job number or jobno. This is an unsigned integer, generated by the scheduler when the job is submitted. The job number is used whenever jobs have to be identified unambiguously.

The job specification also includes a title, providing a more user friendly means of identifying jobs on the queue. This title is specified and editable by users and so cannot be guaranteed to be unique.

As part of the title specification, a job can be associated with a queue. Each job may only belong to one queue and a queue may hold many jobs. Queues and their uses are described later in this chapter.



To specify or edit a title (and queue name) use the **Title pri/loadlev** menu option.

In btqw this is available from the **Action** menu.

In btrw it is available under the **Jobs** menu.

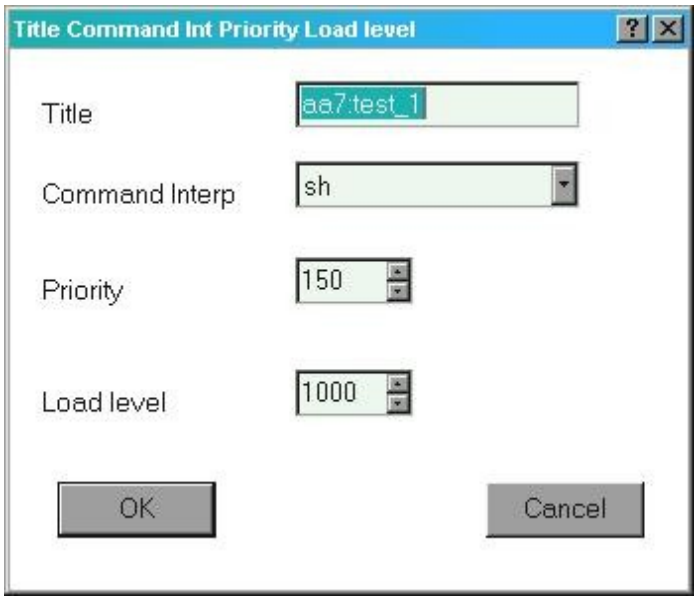
If the job has a queue name it will appear in front of the title, separated from it by a colon. For example if prog_a is in queue admin it will look like this:

admin:prog_a

To specify or change a queue name type it in the title string using that format.

7.11 Priority, Load Level and Command Interpreter

The priority, load level and command interpreter are loosely related in that they indicate the importance and impact on the system of a job.



All jobs are run under a command interpreter, which is referred to by name in the job specification. Command interpreters are separate entities which specify a default load

level for jobs submitted to run under them. See the section on command interpreters later in this chapter for more information.

The load level is held as an unsigned 16 bit integer. It specifies the relative impact that a job is likely to have on the loading of the host machine.

Priority is specified as an integer in the range 1 to 255, and controls how likely a job is to be run ahead of other jobs in the queue. If there were no conditions on jobs then they would all run as soon as their start time arrived.

There may be more jobs ready to start than the system will allow. In this case jobs with the higher priority get started ahead of lower priority ones. When the maximum number of jobs are running, those that did not get in have to wait until one or more of the higher priority jobs finish before being started.

To specify or change these parameters use the **Title pri/loadlev** menu option.

In btqw this is available from the **Action** menu.

In btrw it is available under the **Jobs** menu.

There may be operational limits set on the Priority and Load Level parameters. E.g. the priorities may have been restricted to values between 1 and 150 instead of 1 and 255.

7.12 Job Control Variables - Conditions and Assignments

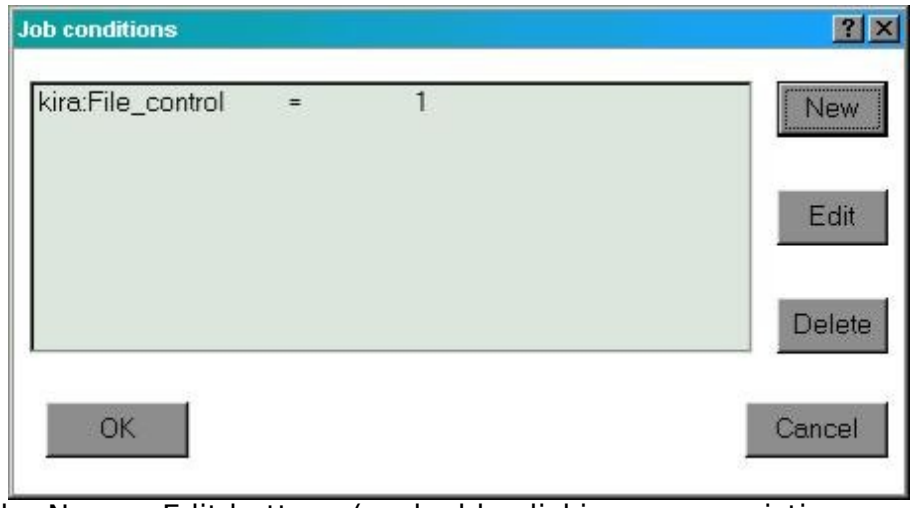
Dependencies between jobs, and other parts of the system, can be implemented using the Job Control Variables, often just called variables. The job specification holds two lists of relationships between a job and the variables. One list specifies conditions which must be true before GNUbatch will allow the job to start. The other list specifies assignments that GNUbatch will perform on the data held in variables when a job starts, stops or fails.

7.12.1 Conditions

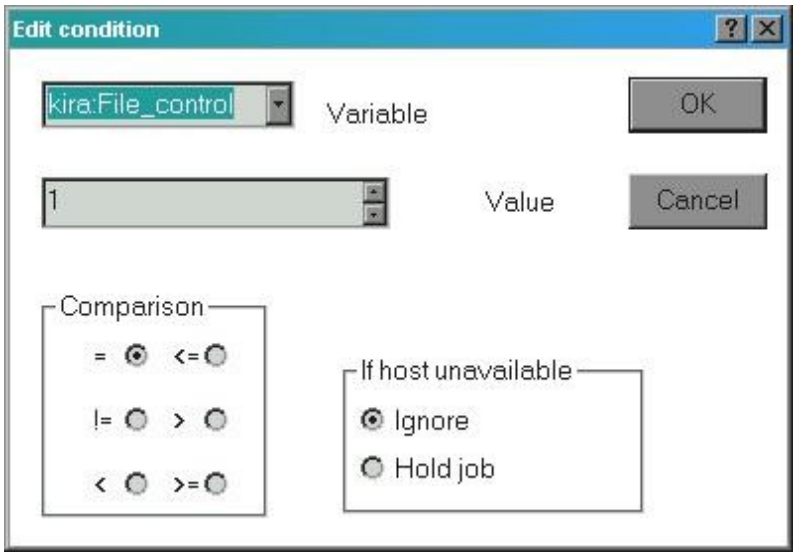
A condition is a simple expression that compares the value of a variable with a literal string or integer constant. The scheduler will not start a job unless all of the conditions are satisfied, i.e. the expressions return a value of true. Up to 10 Conditions may be specified for each job. The expression has the following three components, in this order:

1. A *variable name*, which may be any variable readable by the user. (This is always preceded by the host name and a colon).
2. A *comparator*, which may be any of =, !=, <, <=, > or >=.
3. A *constant*, which can be a string or an integer (negative or positive).

The list of all conditions which are specified for a job can be seen and edited using the Job conditions dialog. A line of information is displayed for each condition. For example a job with one condition, that the variable `CHAIN_STATUS` must contain the value 1, will look like this:



Clicking on the New or Edit buttons (or double-clicking on an existing condition) opens another dialog, for specifying all of the parameters for a condition.



Where a condition refers to a variable on a remote machine, there is always the possibility that the machine is not running or disconnected. To handle this the condition has an option to specify whether the condition is critical or not.

If the condition is specified as *critical* the job has to wait until the machine is available, and the variable satisfies the expression, before running. This is set by the **Hold job** option in the **If host unavailable** box.

Alternatively if the condition is specified as *non-critical* and the machine is not available, the expression will be treated as satisfied. This is set by the **ignore** option in the **If host unavailable** box.

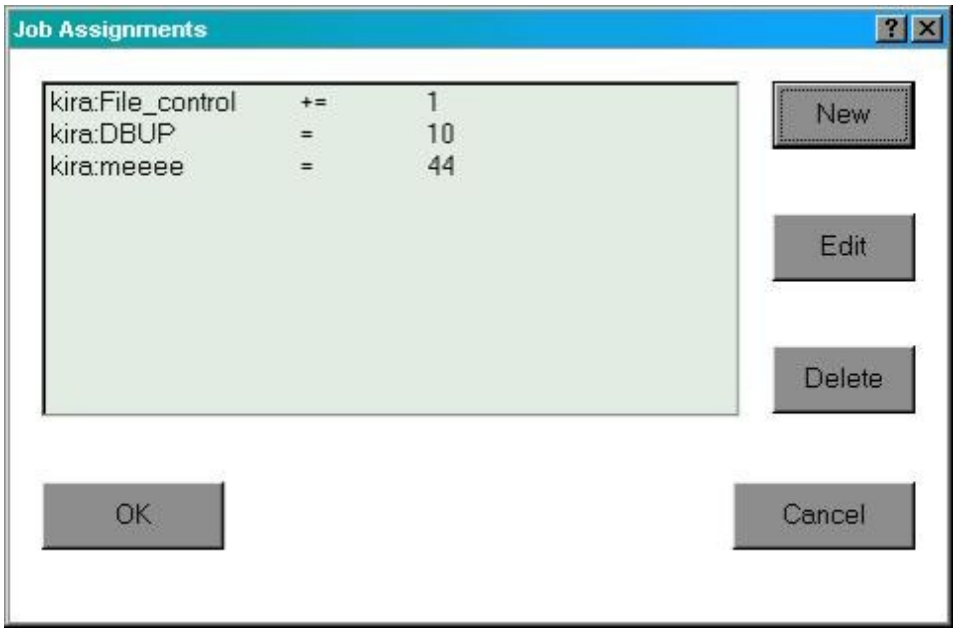
7.12.2 Assignments

Up to 8 assignments may be specified for a job. Each assignment specifies what operation to perform on a variable and under what circumstances to perform the operation. The operation is specified as a simple programming language like assignment statement, hence the name assignment. The circumstances are defined

by a set of flags; all, one or more of which may be set.

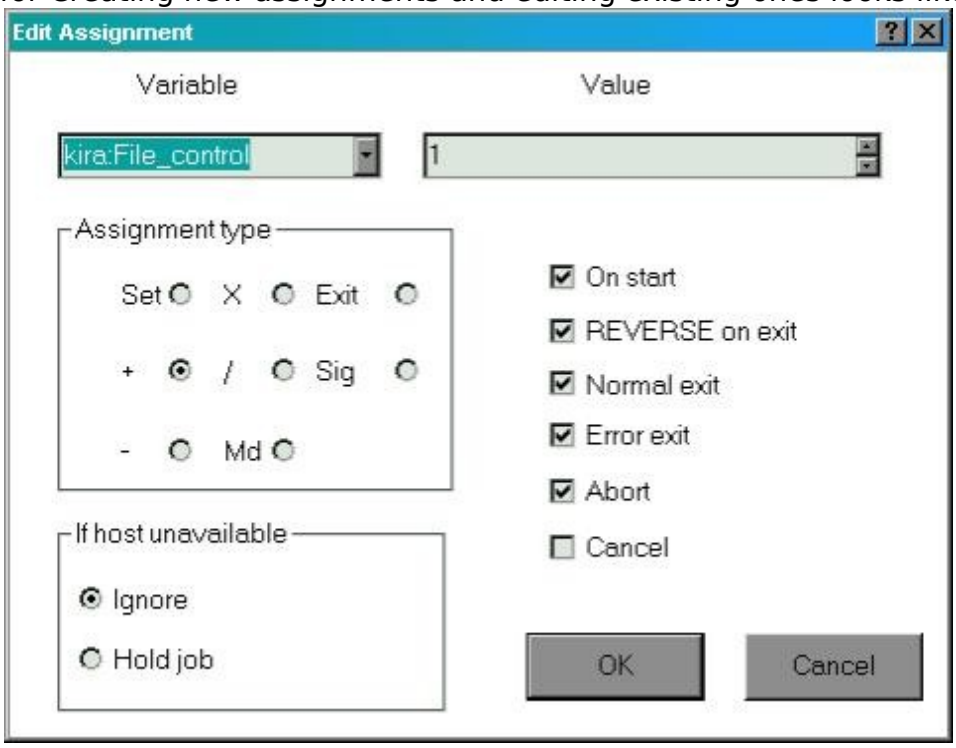
There are two special cases of the assignment. One performs a straight assignment of the exit code, with which the job terminated, to a variable. The other does the same thing with the signal number, if the job was killed.

Jobs Assignments can be seen, edited and created using this dialog:



This example shows 3 assignments.

The dialog for Creating new assignments and editing existing ones looks like this:



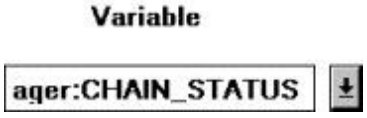
There are six flags to specify when an assignment should be performed. At least one flag must be set. They can be used in combination or all set as required. The flags are:

☐ On start	Start, the scheduler performs the assignment when it starts the job
☐ REVERSE	Reverse the start assignment for on exit specifications.
☒ Normal	Normal exit, performs the assignment on normal exit
☐ Error	Error exit, performs the assignment on error exit
☐ Abort	Aborted, performs assignment if job aborted (signal)
☐ Cancel	Cancellation, performs assignment if job cancelled.

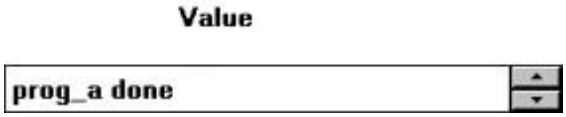
The **Reverse** flag is only relevant when the one or more of the exit flags is set. It causes the assignment to be reversed when the job started, when the job finishes. If all the flags are set then the assignment is performed on start up, and reset when the job finishes however it exited. If only the flags **Start Normal** and **Reverse** are set then the assignment is only reversed when the job finishes normally.

Each assignment statement has 3 components, which are:

- A *variable name*, which must already exist and be *writeable* by the user. This is always preceded by a host name and a colon. For example this is CHAIN_STATUS from machine voyager. The first three letters of the host name can not be fitted in the field.



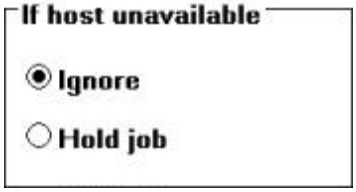
- An *assignment operator*, which can be just assignment, one of the arithmetic operators, assignment of signal and assignment of the exit code.
- A *constant*, which may be a string or numeric value. Only the = operator is valid for assignments with strings



If the **Reverse** flag is set, to reverse a start assignment, the assignments performed are to assign zero or the empty string in the case of plain assignment and to interchange add and subtract, multiply and divide for arithmetic operations. Modulus taking is irreversible and the reverse flag has no effect on this.

Exit code and signal setting only apply on exit, so the reverse flag has no effect on these.

Where an assignment operates upon a variable on a remote machine, there is a possibility that the remote machines copy of the scheduler is not running or disconnected. To handle this the assignment has an option to specify whether the operation is critical or not.



If the assignment is specified as *critical* the job has to wait until the machine is available, for the operation to be performed, before running. This option can be specified by clicking on the **Hold job** button in the **If host unavailable** box.

Alternatively if the assignment is specified as *non-critical* and the machine is not available, the job will be run without performing the specified operation. This option can be specified by clicking on the **Ignore** button in the **If host unavailable** box.

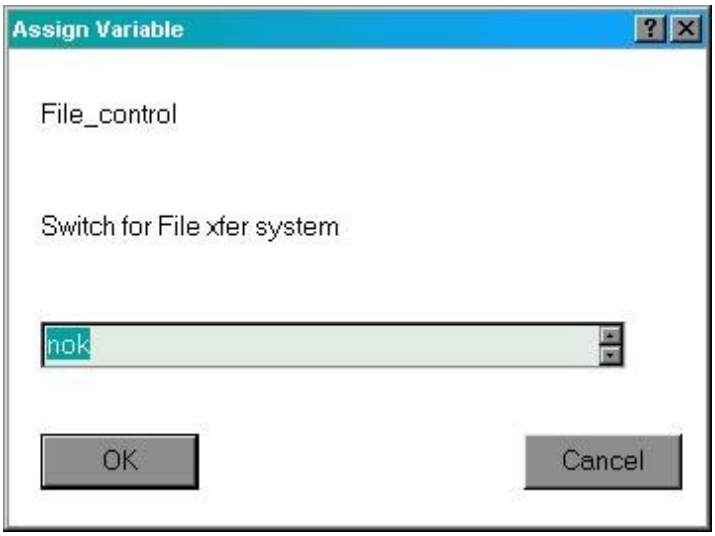
Once a job is running the *critical* specification has no effect. If a remote variable becomes unavailable during execution of a job, any critical job completion assignments to that variable are ignored.

8 Variables

GNUbatch variables may be edited from within btqw as follows:

8.1 Assignment

You may assign a new value to a variable to which you have access by clicking on the variable and selecting the assign entry from the variable menu, or selecting the assign variable from the pop-up value from right-clicking on the variable.



The dialog is as follows:

The name and comment are displayed and the current value in the box.

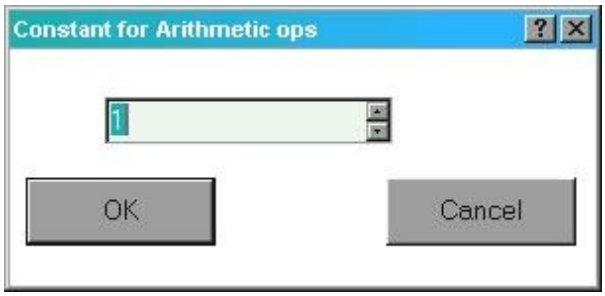
If the current value is numeric (not in this example), it may be incremented or decremented in steps using the buttons on the right of the box.

Press **OK** when done.

8.2 Quick arithmetic

You can set up a number (default 1) and perform arithmetic with that number on the value of a variable if it is numeric. All of the arithmetic operations have shortcut keys and increment and decrement are on the "right-click pop-up menu".

To set the constant, select **Set Constant** from the variables window to receive the dialog:



Use the buttons to adjust the value, or just type in a new value as required.

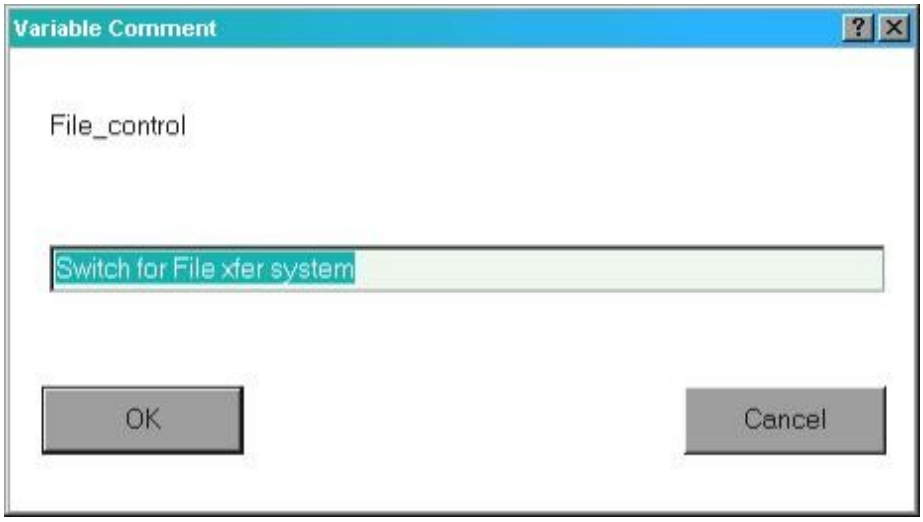
To apply the constant, select the variable and select the option - increment, decrement, multiply, divide or remainder.

Increment and decrement are the most common and are found on the toolbar and in the "right-click pop-up menu".

The arithmetic operations all apply the constant to the variable. Divide and remainder provide the quotient and remainder respectively when the current value of the variable is divided by the constant. Attempts to divide by zero are ignored.

8.3 Comment

The comment can be changed using the Variable Comment dialog which looks like:



9 Installation

The GNUbatch Windows interface is provided as standard *InstallShield* installation sets.

We suggest that you first set up the Unix hosts as explained in the Administration Manual according to whether the PCs have fixed IP addresses or are DHCP clients. Then run the *Windows* program btrsetw to set up the GNUbatch hosts file on the PC.