

# ACRC How-To: Submitting Job Arrays to BlueCrystal Phase 3

## Table of Contents

ACRC How-To: Submitting Job Arrays to BlueCrystal Phase 3.....	1
Introduction.....	1
Examples.....	2

### Introduction

If you are exploring a parameter-space or perhaps running a set of monte-carlo simulations, it is not uncommon to submit a large number of jobs (hundreds or perhaps thousands) to the scheduler. Preparing, submitting, managing and tracking all these jobs can be an administrative headache. For these reasons, the scheduler offers you some support in the form of *job arrays*. If you avail yourself of this feature, you can:

- Submit all your jobs with a single command, and
- modify (or delete) all, or a range of your jobs, again with a single command.

Further more, the scheduler will set the environment variable *PBS\_ARRAYID* to a unique value for each of your jobs. You can use this value to automatically initialise your program, either via the command line or scripted insertion into parameter file, such as a Fortran namelist file. Where the initialisation could specify a region of parameter-space, or form part of a random seed for a monte-carlo simulation.

In the next section, I will provide an example submission script to launch a job array. You can copy this script and modify the details to suit your particular task.

## Examples

Below is an example submission script that will launch several jobs as part of a job array. Note that **-t** is the option which indicates a job array.

```
#!/bin/bash

#PBS -N my_job_array

## NB: The resource request is per job,
## and NOT the total resource required for the whole array of jobs.
#PBS -l nodes=1:ppn=1,walltime=00:02:00

## The -t option specifies how many jobs are in the array.
## For example -t 1-4 specifies 4 jobs. The IDs will be in the range [1,4] with a stride of 1.
## Other examples include:
## -t 1,5,7-9
## in this case IDs 1 and 5 and 7,8 and 9 will be used.
## -t 1-50%5
## in this case IDs 1 to 50 will be used and only 5 jobs can run concurrently.
#PBS -t 1,5,7-9%1

cd $PBS_O_WORKDIR
echo "The Array ID is: ${PBS_ARRAYID}"

## We can then use this variable to trigger the difference between
## jobs. For example we could provide a pre-prepared parameter file to an executable:
## ./my_prog.exe paramfile-${PBS_ARRAYID}.txt

## Or invoke some scripting based on the ARRAYID:
let START_VAL=${PBS_ARRAYID}*100
echo "START_VAL is set to: ${START_VAL}"
## and perhaps pass this value to your executable as a command-line argument:
## ./my_prog.exe ${START_VAL}
```

As stated in the introduction, once an array is submitted to the scheduler, you can apply subsequent commands to the whole or part array. For example, you can change aspects of the resource request for queued jobs using **qalter**. In the following example, we change the *walltime* request for the job array with the given job ID:

- **qalter 362415[] -l walltime=00:01:00**

Note the inclusion of the square brackets, and change the Job ID to that relevant to your job.

We can also delete either all or part of the array from the queue. For example:

- **qdel 362415[]** will delete an entire job array,
- **qdel 362415[] -t 7-9** will delete a portion of the array corresponding to the given range, and
- **qdel 362415[1]** will delete only the job with the specified Array ID from the array.

To see the status for all your jobs, including those in an array, type:

- **qstat -u <username> -t**