



## PBS Pro – Documentation

### PBS Pro Arrays – Advanced

#### Introduction

A limitation of arrays is that it can't process a multi-dimensional array of jobs. To overcome this limitation, a flat input file can be created with the required variables and this can be used to create an array of jobs or multi inputs to an executable.

The PBS command files below are examples of array jobs using a file to supply variables to an executable. The file was previously generated.

#### Submitting an Array Job via PBS using a file list

This first array example takes information, e.g. variable or file name, from a previously generated file and passes it to an executable.

```
#!/bin/bash -l
#
#####
# Name:      run_array.pbs
# Description: pbs file to run an array jobs
#
# set jobname
#PBS -N [name of array job]

# set job resources
# [change these resource settings as required]
#PBS -l select=1:ncpus=20
#PBS -l walltime=24:00:00
#PBS -l mem=64gb

# set job array size
# [change this setting as required]
#PBS -J 1-200

# set project code
# [change this to the correct project code]
#PBS -P [projectid]

# set default for email notifications
# [change this to the correct email address]
#PBS -m bea
#PBS -M username@usq.edu.au

# get system variables
#PBS -V
```



## PBS Pro – Documentation

```
# load software modules
module load [module name]

# change to working directory
cd $PBS_O_WORKDIR

# set job temporary working directory
TMPDIR=/sandisk1/${USER}
PBS_TMPDIR=${TMPDIR}

# get a particular line from file
parameters=`sed -n "${PBS_ARRAY_INDEX} p" array_file.list`
# "sed -n" - run sed in quiet mode
# "PBS_ARRAY_INDEX p" – get array index number line of file
# "array_file.list" – file containing list of file names or variables
# "parameters" – place the output from sed command into a variable

# get a particular field from file line from above
arrayinputfile=`echo $parameters | cut -d ' ' -f 2`
# cut the second field from the parameters variable
# and place into the arrayinputfile variable

# executable program
arrayexecutable.sh ${arrayinputfile}
```

### Submitting a Job via PBS using a file list with multi variables

This second array example takes information, e.g. variables, from a previously generated file and passes it to an executable.

```
#!/bin/bash
#
#####
# Name:      run_array.pbs
# Description: pbs file to run an array jobs
#
# set jobname
#PBS -N [name of array job]

# set job resources
# [change these resource settings as required]
#PBS -l select=1:ncpus=20
#PBS -l walltime=24:00:00
#PBS -l mem=64gb

# set job array size
# [change these resource settings as required]
#PBS -J 1-200

# set project code
```



## PBS Pro – Documentation

```
# [change this to the correct project code]
#PBS -P [projectid]

# set default for email notifications
# [change this to the correct email address]
#PBS -m bea
#PBS -M username@usq.edu.au

# get system variables
#PBS -V

# load software modules
module load [module name]

# change to working directory
cd $PBS_O_WORKDIR

# set job temporary working directory
TMPDIR=/sandisk1/${USER}
PBS_TMPDIR=${TMPDIR}

# get file list and submit array to cluster
parameters=`sed -n "${PBS_ARRAY_INDEX} p" array_file.list`
# "sed -n" - run sed in quiet mode
# "PBS_ARRAY_INDEX p" - get array index number line of file
# "array_file.list" - file containing list of file names or variables
# "parameters" - place the output from sed command into a variable

# create an array from file line from above
fileinputline =($parameters)

# get fields from array and place them into variables
var1=${fileinputline[0]}
var2=${fileinputline[1]}
var3=${fileinputline[2]}

# executable program with required variables
arrayexecutable.sh $var1 $var2 $var3
```

### References:

1. [PBS Professional 14 User Guide](#)
2. [PBS Professional 14 Administrator's Guide](#)
3. [PBS Professional - HPC Cluster Workload Manager](#)