Altair®

PBS Professional®

18.2.3

Reference Guide
You are reading the Altair PBS Professional 18.2.3

Reference Guide (RG)

Updated 12/18/18

Copyright © 2003-2018 Altair Engineering, Inc. All rights reserved.

ALTAIR ENGINEERING INC. Proprietary and Confidential. Contains Trade Secret Information. Not for use or disclosure outside of Licensee’s organization. The software and information contained herein may only be used internally and are provided on a non-exclusive, non-transferable basis. Licensee may not sublicense, sell, lend, assign, rent, distribute, publicly display or publicly perform the software or other information provided herein, nor is Licensee permitted to decompile, reverse engineer, or disassemble the software. Usage of the software and other information provided by Altair (or its resellers) is only as explicitly stated in the applicable end user license agreement between Altair and Licensee. In the absence of such agreement, the Altair standard end user license agreement terms shall govern.

Use of Altair’s trademarks, including but not limited to “PBS™”, “PBS Professional®”, “PBS Works™”, “PBS Control™”, “PBS Access™”, “PBS Analytics™”, “PBScloud.io™”, and Altair’s logos is subject to Altair’s trademark licensing policies. For additional information, please contact Legal@altair.com and use the wording “PBS Trademarks” in the subject line.

For a copy of the end user license agreement(s), log in to https://secure.altair.com/UserArea/agreement.html or contact the Altair Legal Department. For information on the terms and conditions governing third party codes included in the Altair Software, please see the Release Notes.

This document is proprietary information of Altair Engineering, Inc.

Contact Us

For the most recent information, go to the PBS Works website, www.pbsworks.com, select "My PBS", and log in with your site ID and password.

Altair

Altair Engineering, Inc., 1820 E. Big Beaver Road, Troy, MI 48083-2031 USA www.pbsworks.com

Sales

pbssales@altair.com 248.614.2400

Please send any questions or suggestions for improvements to agu@altair.com.
# Technical Support

Need technical support? We are available from 8am to 5pm local times:

<table>
<thead>
<tr>
<th>Location</th>
<th>Telephone</th>
<th>e-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>+1 800 174 396</td>
<td><a href="mailto:anz-pbssupport@india.altair.com">anz-pbssupport@india.altair.com</a></td>
</tr>
<tr>
<td>China</td>
<td>+86 (0)21 6117 1666</td>
<td><a href="mailto:pbs@altair.com.cn">pbs@altair.com.cn</a></td>
</tr>
<tr>
<td>France</td>
<td>+33 (0)1 4133 0992</td>
<td><a href="mailto:pbssupport@europe.altair.com">pbssupport@europe.altair.com</a></td>
</tr>
<tr>
<td>Germany</td>
<td>+49 (0)7031 6208 22</td>
<td><a href="mailto:pbssupport@europe.altair.com">pbssupport@europe.altair.com</a></td>
</tr>
<tr>
<td>India</td>
<td>+91 80 66 29 4500</td>
<td><a href="mailto:pbs-support@india.altair.com">pbs-support@india.altair.com</a></td>
</tr>
<tr>
<td></td>
<td>+1 800 425 0234 (Toll Free)</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>+39 800 905595</td>
<td><a href="mailto:pbssupport@europe.altair.com">pbssupport@europe.altair.com</a></td>
</tr>
<tr>
<td>Japan</td>
<td>+81 3 6225 5821</td>
<td><a href="mailto:pbs@altairjp.co.jp">pbs@altairjp.co.jp</a></td>
</tr>
<tr>
<td>Korea</td>
<td>+82 70 4050 9200</td>
<td><a href="mailto:support@altair.co.kr">support@altair.co.kr</a></td>
</tr>
<tr>
<td>Malaysia</td>
<td>+91 80 66 29 4500</td>
<td><a href="mailto:pbs-support@india.altair.com">pbs-support@india.altair.com</a></td>
</tr>
<tr>
<td></td>
<td>+1 800 425 0234 (Toll Free)</td>
<td></td>
</tr>
<tr>
<td>North America</td>
<td>+1 248 614 2425</td>
<td><a href="mailto:pbssupport@altair.com">pbssupport@altair.com</a></td>
</tr>
<tr>
<td>Russia</td>
<td>+49 7031 6208 22</td>
<td><a href="mailto:pbssupport@europe.altair.com">pbssupport@europe.altair.com</a></td>
</tr>
<tr>
<td>Scandinavia</td>
<td>+46 (0)46 460 2828</td>
<td><a href="mailto:pbssupport@europe.altair.com">pbssupport@europe.altair.com</a></td>
</tr>
<tr>
<td>Singapore</td>
<td>+91 80 66 29 4500</td>
<td><a href="mailto:pbs-support@india.altair.com">pbs-support@india.altair.com</a></td>
</tr>
<tr>
<td></td>
<td>+1 800 425 0234 (Toll Free)</td>
<td></td>
</tr>
<tr>
<td>South Africa</td>
<td>+27 21 831 1500</td>
<td><a href="mailto:pbssupport@europe.altair.com">pbssupport@europe.altair.com</a></td>
</tr>
<tr>
<td>South America</td>
<td>+55 11 3884 0414</td>
<td><a href="mailto:br_support@altair.com">br_support@altair.com</a></td>
</tr>
<tr>
<td>UK</td>
<td>+44 (0)1926 468 600</td>
<td><a href="mailto:pbssupport@europe.altair.com">pbssupport@europe.altair.com</a></td>
</tr>
</tbody>
</table>
## Contents

About PBS Documentation \hspace{1cm} v

1 Glossary of Terms \hspace{1cm} 1

2 PBS Commands \hspace{1cm} 19

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Requirements for Commands</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>mpiexec</td>
<td>24</td>
</tr>
<tr>
<td>2.3</td>
<td>nqs2pbs</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>pbs</td>
<td>28</td>
</tr>
<tr>
<td>2.5</td>
<td>pbsdsh</td>
<td>29</td>
</tr>
<tr>
<td>2.6</td>
<td>pbsfs</td>
<td>31</td>
</tr>
<tr>
<td>2.7</td>
<td>pbsnodes</td>
<td>35</td>
</tr>
<tr>
<td>2.8</td>
<td>pbsrun</td>
<td>40</td>
</tr>
<tr>
<td>2.9</td>
<td>pbsrun_unwrap</td>
<td>50</td>
</tr>
<tr>
<td>2.10</td>
<td>pbsrun_wrap</td>
<td>51</td>
</tr>
<tr>
<td>2.11</td>
<td>pbs_account</td>
<td>53</td>
</tr>
<tr>
<td>2.12</td>
<td>pbs_attach</td>
<td>55</td>
</tr>
<tr>
<td>2.13</td>
<td>pbs_comm</td>
<td>57</td>
</tr>
<tr>
<td>2.14</td>
<td>pbs_dataservice</td>
<td>60</td>
</tr>
<tr>
<td>2.15</td>
<td>pbs_ds_password</td>
<td>61</td>
</tr>
<tr>
<td>2.16</td>
<td>pbs_hostn</td>
<td>63</td>
</tr>
<tr>
<td>2.17</td>
<td>pbs_idled</td>
<td>64</td>
</tr>
<tr>
<td>2.18</td>
<td>pbs_iff</td>
<td>66</td>
</tr>
<tr>
<td>2.19</td>
<td>pbs_interactive</td>
<td>67</td>
</tr>
<tr>
<td>2.20</td>
<td>pbs_lamboot</td>
<td>68</td>
</tr>
<tr>
<td>2.21</td>
<td>pbs_migrate_users</td>
<td>69</td>
</tr>
<tr>
<td>2.22</td>
<td>pbs_migration_assistant.bat</td>
<td>70</td>
</tr>
<tr>
<td>2.23</td>
<td>pbs_mkdirs</td>
<td>71</td>
</tr>
<tr>
<td>2.24</td>
<td>pbs_mom</td>
<td>72</td>
</tr>
<tr>
<td>2.25</td>
<td>pbs_mpihp</td>
<td>77</td>
</tr>
<tr>
<td>2.26</td>
<td>pbs_mpiam</td>
<td>79</td>
</tr>
<tr>
<td>2.27</td>
<td>pbs_mpirun</td>
<td>80</td>
</tr>
<tr>
<td>2.28</td>
<td>pbs_password</td>
<td>82</td>
</tr>
<tr>
<td>2.29</td>
<td>pbs_probe</td>
<td>84</td>
</tr>
<tr>
<td>2.30</td>
<td>pbs_python</td>
<td>86</td>
</tr>
<tr>
<td>2.31</td>
<td>pbs_ralter</td>
<td>89</td>
</tr>
<tr>
<td>2.32</td>
<td>pbs_rdel</td>
<td>92</td>
</tr>
<tr>
<td>2.33</td>
<td>pbs_release_nodes</td>
<td>93</td>
</tr>
<tr>
<td>2.34</td>
<td>pbs_rstat</td>
<td>95</td>
</tr>
<tr>
<td>2.35</td>
<td>pbs_rsub</td>
<td>97</td>
</tr>
<tr>
<td>2.36</td>
<td>pbs_sched</td>
<td>104</td>
</tr>
<tr>
<td>2.37</td>
<td>pbs_server</td>
<td>106</td>
</tr>
<tr>
<td>2.38</td>
<td>pbs_snapshot</td>
<td>110</td>
</tr>
<tr>
<td>2.39</td>
<td>pbs_tclsh</td>
<td>116</td>
</tr>
<tr>
<td>2.40</td>
<td>pbs_tmrsh</td>
<td>117</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.41</td>
<td>pbs_topologyinfo</td>
<td>119</td>
</tr>
<tr>
<td>2.42</td>
<td>pbs_wish</td>
<td>121</td>
</tr>
<tr>
<td>2.43</td>
<td>printjob</td>
<td>122</td>
</tr>
<tr>
<td>2.44</td>
<td>qalter</td>
<td>124</td>
</tr>
<tr>
<td>2.45</td>
<td>qdel</td>
<td>137</td>
</tr>
<tr>
<td>2.46</td>
<td>qdisable</td>
<td>140</td>
</tr>
<tr>
<td>2.47</td>
<td>qenable</td>
<td>142</td>
</tr>
<tr>
<td>2.48</td>
<td>qhold</td>
<td>144</td>
</tr>
<tr>
<td>2.49</td>
<td>qmgr</td>
<td>146</td>
</tr>
<tr>
<td>2.50</td>
<td>qmove</td>
<td>167</td>
</tr>
<tr>
<td>2.51</td>
<td>qmsg</td>
<td>169</td>
</tr>
<tr>
<td>2.52</td>
<td>qorder</td>
<td>171</td>
</tr>
<tr>
<td>2.53</td>
<td>qrun</td>
<td>173</td>
</tr>
<tr>
<td>2.54</td>
<td>qrls</td>
<td>175</td>
</tr>
<tr>
<td>2.55</td>
<td>qrun</td>
<td>177</td>
</tr>
<tr>
<td>2.56</td>
<td>qselect</td>
<td>181</td>
</tr>
<tr>
<td>2.57</td>
<td>qsig</td>
<td>187</td>
</tr>
<tr>
<td>2.58</td>
<td>qstart</td>
<td>190</td>
</tr>
<tr>
<td>2.59</td>
<td>qstat</td>
<td>192</td>
</tr>
<tr>
<td>2.60</td>
<td>qstop</td>
<td>205</td>
</tr>
<tr>
<td>2.61</td>
<td>qsub</td>
<td>207</td>
</tr>
<tr>
<td>2.62</td>
<td>qterm</td>
<td>226</td>
</tr>
<tr>
<td>2.63</td>
<td>tracejob</td>
<td>228</td>
</tr>
<tr>
<td>2.64</td>
<td>win_postinstall.py</td>
<td>231</td>
</tr>
<tr>
<td>2.65</td>
<td>xpbs</td>
<td>232</td>
</tr>
<tr>
<td>2.66</td>
<td>xpbsmon</td>
<td>243</td>
</tr>
<tr>
<td>3</td>
<td>MoM Parameters</td>
<td>255</td>
</tr>
<tr>
<td>3.1</td>
<td>Syntax of MoM Configuration File</td>
<td>255</td>
</tr>
<tr>
<td>3.2</td>
<td>Contents of MoM Configuration File</td>
<td>256</td>
</tr>
<tr>
<td>4</td>
<td>Scheduler Parameters</td>
<td>265</td>
</tr>
<tr>
<td>4.1</td>
<td>Format of Scheduler Configuration File</td>
<td>265</td>
</tr>
<tr>
<td>4.2</td>
<td>Configuration Parameters</td>
<td>266</td>
</tr>
<tr>
<td>5</td>
<td>List of Built-in Resources</td>
<td>277</td>
</tr>
<tr>
<td>5.1</td>
<td>Resource Data Types</td>
<td>277</td>
</tr>
<tr>
<td>5.2</td>
<td>Viewing Resource Information</td>
<td>277</td>
</tr>
<tr>
<td>5.3</td>
<td>Attributes where Resources Are Tracked</td>
<td>278</td>
</tr>
<tr>
<td>5.4</td>
<td>Resource Table Format</td>
<td>278</td>
</tr>
<tr>
<td>5.5</td>
<td>Resources Built Into PBS</td>
<td>280</td>
</tr>
<tr>
<td>6</td>
<td>Attributes</td>
<td>291</td>
</tr>
<tr>
<td>6.1</td>
<td>Attribute Behavior</td>
<td>291</td>
</tr>
<tr>
<td>6.2</td>
<td>How To Set Attributes</td>
<td>291</td>
</tr>
<tr>
<td>6.3</td>
<td>Viewing Attribute Values</td>
<td>292</td>
</tr>
<tr>
<td>6.4</td>
<td>Attribute Table Format</td>
<td>293</td>
</tr>
<tr>
<td>6.5</td>
<td>Caveats</td>
<td>294</td>
</tr>
<tr>
<td>6.6</td>
<td>Server Attributes</td>
<td>295</td>
</tr>
<tr>
<td>6.7</td>
<td>Scheduler Attributes</td>
<td>314</td>
</tr>
<tr>
<td>6.8</td>
<td>Reservation Attributes</td>
<td>317</td>
</tr>
<tr>
<td>6.9</td>
<td>Queue Attributes</td>
<td>324</td>
</tr>
</tbody>
</table>
About PBS Documentation

The PBS Professional guides and release notes apply to the commercial releases of PBS Professional.

Document Conventions

**Abbreviation**

The shortest acceptable abbreviation of a command or subcommand is underlined.

**Attribute**

Attributes, parameters, objects, variable names, resources, types

**Command**

Commands such as `qmgr` and `scp`

**Definition**

Terms being defined

**File name**

File and path names

**Input**

Command-line instructions

**Method**

Method or member of a class

**Output**

Output, example code, or file contents

**Syntax**

Syntax, template, synopsis

**Utility**

Name of utility, such as a program

**Value**

Keywords, instances, states, values, labels

Notation

**Optional arguments** are enclosed in square brackets. For example:

```
qstat [-E]
```

**Variables** are enclosed in angle brackets. A variable is something the user must fill in with the correct value. In the following example, the user replaces `vnode name` with the name of the vnode:

```
pbsnodes -v <vnode name>
```
About PBS Documentation

Optional variables are enclosed in angle brackets inside square brackets. For example:

```
qstat [<job ID>]
```

Literal terms appear exactly as they should be used. For example, to get the version of the `qstat` command, type the following exactly:

```
qstat --version
```

List of PBS Professional Documentation

**PBS Professional Release Notes**
Supported platforms, what’s new and/or unexpected in this release, deprecations and interface changes, open and closed bugs, late-breaking information. For administrators and job submitters.

**PBS Professional Installation & Upgrade Guide:**
How to install and upgrade PBS Professional. For the administrator.

**PBS Professional Administrator’s Guide:**
How to configure and manage PBS Professional. For the PBS administrator.

**PBS Professional Hooks Guide:**
How to write and use hooks for PBS Professional. For the administrator.

**PBS Professional User’s Guide:**
How to submit, monitor, track, delete, and manipulate jobs. For the job submitter.

**PBS Professional Reference Guide:**
Covers PBS reference material.

**PBS Professional Programmer’s Guide:**
Discusses the PBS application programming interface (API). For integrators.

**PBS Manual Pages:**
PBS commands, resources, attributes, APIs.

**PBS Professional Quick Start Guide:**
Quick overview of PBS Professional installation and license file generation.

Where to Keep the Documentation

To make cross-references work, put all of the PBS guides in the same directory.

Ordering Software and Licenses

To purchase software packages or additional software licenses, contact your Altair sales representative at pbssales@altair.com.
This chapter describes the terms used in PBS Professional documentation.

Accept an action (Hooks)

A hook accepts an action when the hook allows the action to take place.

Access control list, ACL

An ACL, or Access Control List, is a list of users, groups, or hosts from which users or groups may be attempting to gain access. This list defines who or what is allowed or denied access to parts of PBS such as the server, queues, or reservations. A server ACL applies to access to the server, and therefore all of PBS. A queue’s ACL applies only to that particular queue. A reservation’s ACL applies only to that particular reservation. See "ACLs" on page 361 in the PBS Professional Administrator’s Guide.

Access to a queue

Applies to users, groups, and hosts. Being able to submit jobs to the queue, move jobs into the queue, being able to perform operations on jobs in the queue, and being able to get the status of the queue.

Access to a reservation

Applies to users, groups, and hosts. Being able to place jobs in the reservation, whether by submitting jobs to the reservation or moving jobs into the reservation. It also means being able to delete the reservation, and being able to operate on the jobs in the reservation.

Access to the server

Applies to users, groups, and hosts. Being able to run PBS commands to submit jobs and perform operations on them such as altering, selecting, and querying status. It also means being able to get the status of the server and queues.

Account

An account is an arbitrary character string, which may have meaning to one or more hosts in the batch system. Frequently, an account is used as a grouping for charging for the use of resources.

Action (Hooks)

A PBS operation or state transition. The actions that hooks can affect are submitting a job, altering a job, running a job, making a reservation, and moving a job to another queue.

Active (Failover)

A server daemon is active when it is managing user requests and communicating with a scheduler and MoMs.

Active Directory (Windows)

Active Directory is an implementation of LDAP directory services by Microsoft to use in Windows environments. It is a directory service used to store information about the network resources (e.g. user accounts and groups) across a domain.

Admin (Windows)

A user logged in from an account that is either:

1. A member of a group having full control over the local computer and the domain controller
2. Allowed to make domain and schema changes to the Active Directory.
Administrator, PBS Administrator
A person who administers PBS, performing functions such as downloading, installing, upgrading, configuring, or managing PBS. A PBS administrator must have an account with Manager privilege and an account with root privilege. Administrator is distinguished from “site administrator”, although often these are the same person.
The term PBS Administrator is used in PBS documentation to mean the above, rather than to indicate a user role recognized by PBS Professional.

Administrators (Windows)
A group that has built-in capabilities that give its members full control over the local system, or the domain controller host itself.

Advance reservation
A reservation for a specific set of resources for a specified start time and duration in the future. Advance reservations are created by users to reserve resources for jobs. The reservation is available only to the creator of the reservation and any users or groups specified by the creator.

AOE, Application operating environment
The environment on a vnode. This may be one that results from provisioning that vnode, or one that is already in place

API
PBS provides an Application Programming Interface, or API, which is used by the commands to communicate with the server. This API is described in the PBS Professional Programmer’s Guide. A site may make use of the API to implement new commands if so desired.

Application Checkpoint
The application performs its own checkpointing when it receives the appropriate signal etc.

Array job
See "Job array".

Attribute
An attribute is a data item belonging to an object. The attribute’s value affects the behavior of or provides information about the object. A job’s owner can set the attributes of a job, and the administrator can set attributes of queues and vnodes.

Backfilling
A scheduling policy where
1. High-priority jobs are scheduled for execution
2. Lower-priority jobs are run if the following conditions are true:
   Resources (that cannot be used by the high-priority jobs) are available
   The lower-priority jobs will not delay the higher-priority jobs
Lower-priority jobs selected for execution are those next in priority order that will fit in the available resources.

Batch, Batch processing
Allowing jobs to be run outside of the interactive login environment.

Borrowing vnode
The vnode where a shared vnode resource is available, but not managed.

Built-in hook
A hook that is supplied as part of PBS. These hooks cannot be created or deleted by administrators. See "Managing Built-in Hooks" on page 73 in the PBS Professional Hooks Guide.
Built-in resource

A resource that is defined in PBS Professional as shipped. Examples of built-in resources are ncpus, which tracks the number of CPUs, and mem, which tracks memory. See "Built-in vs. Custom Resources" on page 229 in the PBS Professional Administrator’s Guide.

Checkpoint/Restart

Allows jobs to be checkpointed and restarted. Uses OS-provided or third-party checkpoint/restart facility.

Checkpoint and Abort, checkpoint_abort

The checkpoint script or tool writes a restart file, then PBS kills and requeues the job. The job resumes from the start file when it is executed again.

Chunk

A set of resources allocated as a unit to a job. Specified inside a selection directive. All parts of a chunk come from the same host. In a typical MPI (Message-Passing Interface) job, there is one chunk per MPI process.

Chunk-level resource, host-level resource

A resource that is available at the host level, for example, CPUs or memory. Chunk resources are requested inside of a selection statement. The resources of a chunk are to be applied to the portion of the job running in that chunk.

Chunks resources are requested inside a select statement. A single chunk is requested using this form:
```
-l select=resource name=value:resource name=value
```
For example, one chunk might have 2 CPUs and 4GB of memory:
```
-l select=ncpus=2:mem=4gb
```
To request multiples of a chunk, prefix the chunk specification by the number of chunks:
```
-l select=number of chunks<chunk specification>
```
For example, to request six of the previous chunk:
```
-l select=6:ncpus=2:mem=4gb
```
To request different chunks, concatenate the chunks using the plus sign (“+”):
```
-l select=number of chunks<chunk specification>+number of chunks<chunk specification>
```
For example, to request two kinds of chunks, one with 2 CPUs per chunk, and one with 8 CPUs per chunk, both kinds with 4GB of memory:
```
-l select=6:ncpus=2:mem=4gb+3:ncpus=8:mem=4GB
```

Chunk set

An identical set of chunks requested in a select statement. The following is a chunk set:
```
4:ncpus=8:mem=4GB
```

Cluster

A relatively homogeneous set of systems that are used as if they are a single machine.

Commands

PBS supplies both command line programs that are POSIX 1003.2d conforming and a graphical interface. These are used to submit, monitor, modify, and delete jobs. These client commands can be installed on any system type supported by PBS and do not require the local presence of any of the other components of PBS.

There are three classifications of commands: user commands (which any authorized user can use), Operator commands, and Manager (or administrator) commands. Operator and Manager commands require specific access privileges.

Communication daemon, comm

The daemon which handles communication between the server, scheduler, and MoMs. Executable is pbs_comm.
Complex
A PBS complex consists of the machines running one primary server+scheduler (plus, optionally, a secondary backup server+scheduler) and all the machines on which the MoMs (attached to this server+scheduler) are running. A complex can be a heterogeneous mix of system architectures, and can include one or more clusters.

Consumable resource
A consumable resource is a resource that is reduced or taken up by being used. Examples of consumable resources are memory or CPUs. See "Consumable vs. Non-consumable Resources" on page 230 in the PBS Professional Administrator's Guide.

CPU
Has two meanings, one from a hardware viewpoint, and one from a software viewpoint:
1. A core. The part of a processor that carries out computational tasks. Some systems present virtual cores, for example in hyperthreading.
2. Resource required to execute a program thread. PBS schedules jobs according, in part, to the number of threads, giving each thread a core on which to execute. The resource used by PBS to track CPUs is called "ncpus". The number of CPUs available for use defaults to the number of cores reported by the OS. When a job requests one CPU, it is requesting one core on which to run.

Creating a hook
When you “create a hook” using qmgr, you’re telling PBS that you want it to make you an empty hook object that has no characteristics other than a name.

Custom resource
A resource that is not defined in PBS as shipped. Custom resources are created by the PBS administrator or by PBS for some systems. See “Built-in vs. Custom Resources” on page 229 in the PBS Professional Administrator’s Guide.

Degraded reservation
An advance reservation for which one or more associated vnodes are unavailable.
A standing reservation for which one or more vnodes associated with any occurrence are unavailable.

Delegation (Windows)
A capability provided by Active Directory that allows granular assignment of privileges to a domain account or group. So for instance, instead of adding an account to the “Account Operators” group which might give too much access, delegation allows giving the account read access only to all domain users and groups information. This is done via the Delegation wizard.

Destination, destination identifier, destination queue, destination server
String. One or more queues or a server. Jobs may be queried at or sent to a destination queue (for an example, see "qmove" on page 167). Commands may be directed to a destination queue or server (for an example, see "qenable" on page 142). A destination may be at the default PBS server or at another server.

Destination queue format:
<queue name>
Indicates specified queue at default server.

@<server name>
When moving a job, indicates default queue at that server.
When operating on queues, can indicate all queues at that server.

<queue name>@<server name>
Indicates specified queue at specified server.

Destination server format:
(no server name)
Indicates default server.

@<server name>
Indicates specified server.

@default
Indicates default server.

**Directive**
A means by which the user specifies to PBS the value of a job submission variable such as number of CPUs, the name of the job, etc. The default start of a directive is “#PBS”. PBS directives either specify resource requirements or attribute values. See page "Changing the Directive Prefix", on page 16 of the PBS Professional User’s Guide.

**Domain Admin Account (Windows)**
A domain account on Windows that is a member of the “Domain Admins” group.

**Domain Admins (Windows)**
A global group whose members are authorized to administer the domain. By default, the Domain Admins group is a member of the Administrators group on all computers that have joined a domain, including the domain controllers.

**Domain User Account (Windows)**
A domain account on Windows that is a member of the Domain Users group.

**Domain Users (Windows)**
A global group that, by default, includes all user accounts in a domain. When you create a user account in a domain, it is added to this group automatically.

**Endpoint**
A PBS server, scheduler, or MoM daemon.

**Enterprise Admins (Windows)**
A group that exists only in the root domain of an Active Directory forest of domains. The group is authorized to make forest-wide changes in Active Directory, such as adding child domains.

**Entity, PBS entity**
A user, group, or host.

**Entity share**
Setting job execution and/or preemption priority according to how much of the fairshare tree is assigned to each job’s owner.

**Event**
A PBS operation or state transition. Also called *action*. For a list of events, see "Event Types" on page 90 in the PBS Professional Hooks Guide.

**Execution event hook**
A hook that runs at an execution host. These hooks run after a job is received by MoM. Execution event hooks have names prefixed with “execjob_”.


Chapter 1

Glossary of Terms

Execution host
A computer which runs PBS jobs. An execution host is a system with a single operating system (OS) image, a unified virtual memory space, one or more CPUs and one or more IP addresses. Systems like Linux clusters, which contain separate computational units each with their own OS, are collections of hosts. Systems such as the HPE SGI 8600 are also collections of hosts.

An execution host can be comprised of one or more vnodes. On the HPE SGI 8600, each blade is treated as a vnode. See "Vnode".

Execution queue
A queue from which a job can be executed.

Failover
The PBS complex can run a backup server. If the primary server fails, the secondary takes over without an interruption in service.

Failure action
The action taken when a hook fails to execute. Specified in the fail_action hook attribute. See "Using the fail_action Hook Attribute" on page 34 in the PBS Professional Hooks Guide.

Fairshare
A scheduling policy that prioritizes jobs according to how much of a specified resource is being used by, and has recently been used by, job submitters. Job submitters can be organized into groups and subgroups, so that jobs can also be prioritized according to those groups' resource usage. Users and groups can each be allotted a percentage of total resource usage. See "Using Fairshare" on page 136 in the PBS Professional Administrator's Guide.

File staging
File staging is the transfer of files between a specified storage location and the execution host. See "Stage in" and "Stage out".

Finished jobs
Jobs whose execution is done, for any reason:
- Jobs which finished execution successfully and exited
- Jobs terminated by PBS while running
- Jobs whose execution failed because of system or network failure
- Jobs which were deleted before they could start execution

Floating license
A unit of license dynamically allocated (checked out) when a user begins using an application on some host (when the job starts), and deallocated (checked in) when a user finishes using the application (when the job ends).

Furnishing queue/complex
In peer scheduling, the queue/complex from which jobs are pulled to be run at another queue/complex

Generic group limit
A limit that applies separately to groups at the server or a queue. This is the limit for groups which have no individual limit specified. A limit for generic groups is applied to the usage across the entire group. A separate limit can be specified at the server and each queue.

Generic project limit
Applies separately to projects at the server or a queue. The limit for projects which have no individual limit specified. A limit for generic projects is applied to the usage across the entire project. A separate limit can be specified at the server and each queue.
Glossary of Terms

Generic user limit
A limit that applies separately to users at the server or a queue. This is the limit for users who have no individual limit specified. A separate limit for generic users can be specified at the server and at each queue.

Global resource
A global resource is defined in a resources_available attribute, at the server, a queue, or a host. Global resources can be operated on via the qmgr command and are visible via the qstat and pbsnodes commands. See "Global vs. Local Resources" on page 231 in the PBS Professional Administrator’s Guide.

Group
A collection of system users. A user must be a member of at least one group, and can be a member of more than one group.

Group access, Access by group
Refers to access to PBS objects, such as the server, queues, and reservations. A user in the specified group is allowed access at the server, queues, and reservations

Group ID (GID)
Unique numeric identifier assigned to each group. See "Group".

Group limit
Refers to configurable limits on resources and jobs. This is a limit applied to the total used by a group, whether the limit is a generic group limit or an individual group limit.

History jobs
Jobs which will no longer execute at this server:
- Moved jobs
- Finished jobs

Hold
A restriction which prevents a job from being executed. When a job has a hold applied to it, it is in the Held (H) state. See section 2.48, “qhold”, on page 144.

HTT
Intel’s Hyper-Threading Technology

Hook
Hooks are custom executables that can be run at specific points in the execution of PBS. They accept, reject, or modify the upcoming action. This provides job filtering, patches or workarounds, and extends the capabilities of PBS, without the need to modify source code.

Host
A machine running an operating system. A host can be made up of one or more vnodes. All vnodes of a host share the same value for resources_available.host.

Host access, Access by host
Refers to user access at the server, queues, and reservations from the specified host

Idle
A server daemon is idle when it is running, but only accepting handshake messages, not performing workload management.

Importing a hook
When you “import a hook” using qmgr, you’re telling PBS which Python script to run when the hook is triggered.
Importing a hook configuration file

When you “import a hook configuration file” using qmgr, you’re telling PBS which file should be stored as the configuration file for the specified hook.

Indirect resource

A shared vnode resource at vnode(s) where the resource is not defined, but which share the resource.

Individual group limit

Applies separately to groups at the server or a queue. This is the limit for a group which has its own individual limit specified. An individual group limit overrides the generic group limit, but only in the same context, for example, at a particular queue. The limit is applied to the usage across the entire group. A separate limit can be specified at the server and each queue.

Individual project limit

Applies separately to projects at the server or a queue. Limit for a project which has its own individual limit specified. An individual project limit overrides the generic project limit, but only in the same context, for example, at a particular queue. The limit is applied to the usage across the entire project. A separate limit can be specified at the server and each queue.

Individual user limit

Applies separately to users at the server or a queue. This is the limit for users who have their own individual limit specified. A limit for an individual user overrides the generic user limit, but only in the same context, for example, at a particular queue. A separate limit can be specified at the server and each queue.

Installation account

The account used by the administrator when installing PBS. Not the pbsadmin account used by PBS.

Interactive job

A job where standard input and output are connected to the terminal from which the job was submitted.

Job or Batch job

A unit of work managed by PBS. A job is a related set of tasks, created and submitted by the user. The user specifies the resources required by the job, and the processes that make up the job. When the user submits a job to PBS, the user is handing off these tasks to PBS to manage. PBS then schedules the job to be run, and manages the running of the job, treating the tasks as parts of a whole. A job is usually composed of a set of directives and a shell script.

Job array

A job array is a container for a collection of similar jobs submitted under a single job ID. It can be submitted, queried, modified and displayed as a unit. The jobs in the collection are called subjobs. For more on job arrays, see "Job Arrays", on page 145 of the PBS Professional User’s Guide.

Job array identifier

The identifier returned upon success when submitting a job array. The format is

<sequence number>[/]

or

<sequence number>[/].server.domain.com.

Note that some shells require you to enclose a job array identifier in double quotes.

Job array range

A specification for a set of subjobs within a job array. When specifying a range, indices used must be valid members of the job array's indices. Format:
Job ID, Job identifier

When a job is successfully submitted to PBS, PBS returns a unique identifier for the job. Format:

<sequence number>[.server][@new server]

Job state

A job exists in one of the possible states throughout its existence within the PBS system. For example, a job can be queued, running, or exiting. See "States" on page 371.

Job Submission Description Language (JSDL)

Language for describing the resource requirements of jobs.

Job-wide resource, server resource, queue resource

A job-wide resource, also called a server-level or queue-level resource, is a resource that is available to the entire job at the server or queue.

A job-wide resource is available to be consumed or matched at the server or queue if you set the server or queue resources_available.<resource name> attribute to the available or matching value. For example, you can define a custom resource called FloatingLicenses and set the server’s resources_available.FloatingLicenses attribute to the number of available floating licenses.

Examples of job-wide resources are shared scratch space, licenses, or walltime.

A job can request a job-wide resource for the entire job, but not for individual chunks. Job-wide resources are requested outside of a selection statement, in this form:

-l keyword=value[,keyword=value ...]

where keyword identifies either a consumable resource or a time-based resource such as walltime.

A resource request “outside of a selection statement” means that the resource request comes after “-l”, but not after “-lselect=”. 

Kill a job

To terminate the execution of a job.

Leaf

An endpoint (a server, scheduler, or MoM daemon.)

License Manager Daemon (lmx-serv-altair)

Daemon that functions as the license server.

License server

Manages licenses for PBS jobs.

License Server List Configuration

One form of redundant license server configuration. A collection of license server files, or "<port number>@<hostname>" settings, pointing to license servers managing Altair licenses. Each server on the list is tried in turn. There could be X licenses on <server1>, Y licenses on <server2>, and Z licenses on <server3>, and the total licenses available would actually be X+Y+Z, but a request must be satisfied only by one server at a time. The first running server is the only server queried.
Glossary of Terms

Limit
A maximum that can be applied in various situations:
• The maximum number of jobs that can be queued
• The maximum number of jobs that can be running
• The maximum number of jobs that can be queued and running
• The maximum amount of a resource that can be allocated to queued jobs
• The maximum amount of a resource that can be consumed at any time by running jobs
• The maximum amount of a resource that can be allocated to queued and running jobs

Load balance
Scheduling policy wherein jobs are distributed across multiple hosts to even out the workload on each host.

Local resource
A local resource is defined in a Version 1 MoM configuration file. Local resources cannot be operated on via the qmgr command and are not visible via the qstat and pbsnodes commands. Local resources can be used by a scheduler. See "Global vs. Local Resources" on page 231 in the PBS Professional Administrator’s Guide.

Manager
A person who has been granted Manager privilege by being listed in the server’s managers attribute. A Manager is authorized to use all restricted capabilities of PBS. A PBS Manager may act upon the server, queues, or jobs. See "Manager" on page 359 in the PBS Professional Administrator’s Guide.

Managing vnode
The vnode where a shared vnode resource is defined, and which manages the resource.

Master provisioning script, Master script (Hooks)
The script that makes up the provisioning hook.

Memory-only vnode
Represents a node board that has only memory resources (no CPUs).

MoM
The daemon which runs on an execution host, managing the jobs on that host. MoM is the informal name for the process called pbs_mom. One MoM runs on each execution host.
MoM runs each job when it receives a copy of the job from the server. MoM creates a new session that is as identical to the user’s login session as possible. For example under Linux, if the user’s login shell is csh, MoM creates a session in which .login is run as well as .cshrc. MoM returns the job’s output to the user when directed to do so by the server.
MoM is a reverse-engineered acronym that stands for “Machine Oriented Mini-server”.

Monitoring
The act of tracking and reserving system resources and enforcing usage policy. This covers both user-level and system-level monitoring as well as monitoring running jobs. Tools are provided to aid human monitoring of the PBS system as well.

Mother Superior
Mother Superior is the MoM on the head or first host of a multihost job. Mother Superior controls the job, communicates with the server, and controls and consolidates resource usage information. When a job is to run on more than one execution host, the job is sent to the MoM on the primary execution host, which then starts the job.
Moved jobs
Jobs which were moved to another server

Node
No longer used. See "Execution host".

Non-consumable resource
A non-consumable resource is a resource that is not reduced or taken up by being used. Examples of non-consumable resources are Boolean resources and walltime. See "Consumable vs. Non-consumable Resources" on page 230 in the PBS Professional Administrator’s Guide.

Non-job event hook
A hook that is not directly related to a specific job. Non-job event hooks are periodic hooks, startup hooks, provisioning hooks, and reservation creation hooks.

nppcu
Number of Processors Per Compute Unit - Cray BASIL 1.3 attribute in the RESERVE XML for specifying how many processors of a compute unit should be used.

Object, PBS object
An element of PBS such as the server, a queue, or a reservation

Occurrence of a standing reservation
An instance of the standing reservation.

An occurrence of a standing reservation behaves like an advance reservation, with the following exceptions:
- While a job can be submitted to a specific advance reservation, it can only be submitted to the standing reservation as a whole, not to a specific occurrence. You can only specify when the job is eligible to run. See the qsub(1B) man page.
- When an advance reservation ends, it and all of its jobs, running or queued, are deleted, but when an occurrence ends, only its running jobs are deleted.

Each occurrence of a standing reservation has reserved resources which satisfy the resource request, but each occurrence may have its resources drawn from a different source. A query for the resources assigned to a standing reservation will return the resources assigned to the soonest occurrence, shown in the resv_nodes attribute reported by pbs_rstat.

Operator
This term means a person who has been granted Operator privilege by being listed in the server’s operators attribute. An Operator can use some but not all of the restricted capabilities of PBS. See "Operator" on page 358 in the PBS Professional Administrator’s Guide.

Overall limit
Limit on the total usage. In the context of server limits, this is the limit for usage at the PBS complex. In the context of queue limits, this is the limit for usage at the queue. An overall limit is applied to the total usage at the specified location. Separate overall limits can be specified at the server and each queue.

Owner, Job owner
The user who submitted a specific job to PBS.

Parameter
A parameter specifies an element of the behavior of a component of PBS. For example, MoMs have parameters specifying which events to log, or what the maximum load should be. Parameters are specified by editing the component’s configuration files.

pbshook
Keyword used by qmgr to operate on built-in hooks.
Chapter 1

Glossary of Terms

PBS Entity
A user, group, or host

pbs Module
The *pbs module* is an interface to PBS and the hook environment. The interface is made up of Python objects, which have attributes and methods. You can operate on these objects using Python code.

PBS Object
An element of PBS such as the server, a queue, or a reservation

pbsadmin (Windows)
The account that is used to execute the PBS daemons `pbs_server`, `pbs_mom`, `pbs_sched`, and `pbs_rshd` via the Service Control Manager on Windows. This must be “`pbsadmin`”.

PBS_HOME
The path containing PBS files. The path under which PBS files are installed on the local system.

PBS_EXEC
The path containing PBS executables. The path under which PBS executables are installed on the local system.

PBS Professional
A workload management system consisting of a server, a scheduler, and any number of execution hosts each managed by a MoM. PBS accepts batch jobs from users, and schedules them on execution hosts according to the policy chosen by the site. PBS manages the jobs and their output according to site-specified policy.

Peer scheduling
A feature allowing different PBS complexes to automatically run each others’ jobs. This way jobs can be dynamically load-balanced across the complexes. Each complex involved in peer scheduling is called a *peer*.

Placement set
A set of vnodes on which jobs can be run, selected so that the job will run as efficiently as possible. Placement sets are used to improve task placement (optimizing to provide a “good fit”) by exposing information on system configuration and topology. See “Placement Sets” on page 167 in the PBS Professional Administrator’s Guide.

Placement set series
The set of placement sets defined by a resource, where each set has the same value for the resource. If the resource takes on N values, there are N placement sets in the series. See “Placement Sets” on page 167 in the PBS Professional Administrator’s Guide.

Placement pool
All of the placement sets defined at a PBS object. Each queue can have its own placement pool, and the server can have its own placement pool. See "Placement Sets" on page 167 in the PBS Professional Administrator’s Guide.

Policy, Scheduling policy
The set of rules by which a scheduler selects jobs for execution.

POSIX
Refers to the various standards developed by the Technical Committee on Operating Systems and Application Environments of the IEEE Computer Society under standard P1003.

PPU
Physical Processing Unit. A grouping of one or more CPUs that share execution resources. May contain multiple processor units or logical processors.

Cray refers to these as “Compute Units”. Other vendors use other terms; AMD's Interlago chips do not contain HTT, however they do contain Modules consisting of two CPUs that share execution resources. A microproces-
sor with HTT also consists of pairs of CPUs that share a set of execution resources (for example 2 sets of registers share a floating point unit). At a high level these two features differ only in the degree of sharing (HTT-enabled Intel chips share more execution resources).

**Preempt**
Stop one or more running jobs in order to start a higher-priority job.

**Preemption level**
Job characteristic used to determine whether a job may preempt another or may be preempted, such as being in an express queue, starving, having an owner who is over a soft limit, being a normal job, or having an owner who is over a fairshare allotment.

**Preemption method**
The method by which a job is preempted. This can be checkpointing, suspension, or requeueing.

**Preemption target**
A preemption target is a job in a specified queue or a job that has requested a specified resource. The queue and/or resource is specified in another job’s `Resource_List.preempt_targets`.

**Pre-execution event hook**
A hook that runs before the job is accepted by MoM. These hooks do not run on execution hosts. Pre-execution event hooks are for job submission, moving a job, altering a job, or just before sending a job to an execution host.

**Primary Scheduler**
The PBS Professional scheduler daemon which is running during normal operation.

**Primary Execution Host**
The execution host where a job’s top task runs, and where the MoM that manages the job runs.

**Primary Server**
The PBS Professional server daemon which is running during normal operation.

**Project**
In PBS, a project is a way to group jobs independently of users and groups. A project is a tag that identifies a set of jobs. Each job’s `project` attribute specifies the job’s project.

**Project limit**
This is a limit applied to the total used by a project, whether the limit is a generic project limit or an individual project limit.

**Provision**
To install an OS or application, or to run a script which performs installation and/or setup

**Provisioned vnode**
A vnode which, through the process of provisioning, has an OS or application that was installed, or which has had a script run on it

**Provisioning hook**
The hook which performs the provisioning, either by calling other scripts or by running commands

**Provisioning tool**
A tool that performs the actual provisioning, e.g. HPE Performance Cluster Manager (HPCM).

**Pulling queue**
In peer scheduling, the queue into which jobs are pulled, and from which they are run
**Queue**

A *queue* is a named container for jobs at a server. There are two types of queues in PBS: routing queues and execution queues. A *routing queue* is a queue used to move jobs to other queues including those that exist on other PBS servers. A job must reside in an *execution queue* to be eligible to run and remains in an execution queue during the time it is running. In spite of the name, jobs in a queue need not be processed in queue order (first-come first-served or FIFO).

**Queuing**

The collecting together of work or tasks to be run on a computer. Users submit tasks or “jobs” to the resource management system where they are queued up until the system is ready to run them.

**Redundant License Server Configuration**

Allows licenses to continue to be available should one or more license servers fail. There are two types: 1) license server list configuration, and 2) three-server configuration.

**Reject an action (Hooks)**

An action is *rejected* when a hook prevents the action from taking place.

**Requeue**

The process of stopping a running job and putting it back into the *queued* ("Q") state.

**Rerunnable**

If a running PBS job can be terminated and then restarted from the beginning without harmful side effects, the job is rerunnable. The job's **Rerunnable** attribute must be set to *y* in order for PBS to consider a job to be rerunnable.

**Reservation Degradation**

PBS attempts to ensure that reservations run by finding usable vnodes when reservation vnodes become unavailable.

**Resource**

A *resource* can be something used by a job, such as CPUs, memory, high-speed switches, scratch space, licenses, or time, or it can be an arbitrary item defined for another purpose. PBS has built-in resources, and allows custom-defined resources. See "Using PBS Resources" on page 225 in the PBS Professional Administrator's Guide.

**Restart**

A job that was stopped after being checkpointed while previously executing is executed again, starting from the point where it was checkpointed.

**Restart File**

The job-specific file that is written by the checkpoint script or tool. This file contains any information needed to restart the job from where it was when it was checkpointed.

**Restart Script**

The script that MoM runs to restart a job. This script is common to all jobs, and so must use the information in a job’s restart file to restart the job.

**Route a job**

When PBS moves a job between queues. PBS provides a mechanism whereby a job is automatically moved from a routing queue to another queue. This is performed by PBS. The resource request for each job in a routing queue is examined, and the job is placed in a destination queue which matches the resource request. The destination queue can be an execution queue or another routing queue.
Routing queue
A queue that serves as a temporary holding place for jobs, before they are moved to another queue. Jobs cannot run from routing queues.

Scheduler
A scheduler is a daemon which implements some or all of the site’s job scheduling policy controlling when and where each job is run. A scheduler is a process called pbs_sched.

Scheduling
The process of selecting which jobs to run when and where, according to a predetermined policy. Sites balance competing needs and goals on the system(s) to maximize efficient use of resources (both computer time and people time).

Scheduling policy
Scheduling policy determines when each job runs, and how much of each resource it can use. Scheduling policy consists of a system for determining the priority of each job, combined with a set of limits on how many jobs can be run, and/or how much of each resource can be used.

Schema Admins (Windows)
A group that exists only in the root domain of an Active Directory forest of domains. The group is authorized to make schema changes in Active Directory.

Secondary Scheduler
The PBS Professional scheduler daemon which takes over when the primary scheduler is not available.

Secondary Server
The PBS Professional server daemon which takes over when the primary server fails.

Sequence_number
The numeric part of a job or job array identifier, e.g. 1234.

Server
The central PBS daemon, which does the following:
- Handles PBS commands
- Receives and creates batch jobs
- Sends jobs for execution

The server is the process called pbs_server.

Each PBS complex has one primary server, and if the complex is configured for failover, a secondary server.

The server contains a licensing client which communicates with the licensing server for licensing PBS jobs.

Shared resource
A vnode resource defined and managed at one vnode, but available for use at others.

Shrink-to-fit job
A job that requests the min_walltime resource. A shrink-to-fit job requests a running time in a specified range, where min_walltime is required, and max_walltime is not. PBS computes the actual walltime.

Sister
Any MoM that is not on the head or first host of a multihost job. A sister is directed by the Mother Superior. Also called a subordinate MoM.

Sisterhood
All of the MoMs involved in running a particular job.
Site
A location which for our purposes uses (or will use) PBS. A site can employ one or more PBS complexes, each made up of any combination of hardware and software PBS supports.

Snapshot Checkpoint
The checkpoint script or tool writes a restart file, and the job continues to execute. The job resumes from this start file if the system experiences a problem during the job’s subsequent execution.

Soonest occurrence of a standing reservation
The occurrence which is currently active, or if none is active, it is the next occurrence.

Stage in
The process of moving one or more job-related files from a storage location to the execution host before running the job.

Stage out
The process of moving one or more job-related files from the execution host to a storage location after running the job.

Staging and execution directory
The staging and execution directory is a directory on the execution host where the following happens:
• Files are staged into this directory before execution
• The job runs in this directory
• Files are staged out from this directory after execution
A job-specific staging and execution directory can be created for each job, or PBS can use a specified directory, or a default directory. See "Staging and Execution Directories for Job" on page 516 in the PBS Professional Administrator’s Guide.

Standing reservation
An advance reservation which recurs at specified times. For example, the user can reserve 8 CPUs and 10GB every Wednesday and Thursday from 5pm to 8pm, for the next three months.

State
The PBS server, vnodes, reservations, and jobs can be in various states, depending on what PBS is doing. For example the server can be idle or scheduling, vnodes can be busy or free, and jobs can be queued or running, among other states. For a complete description of states, see "States" on page 371.

Strict ordering
A scheduling policy where jobs are run according to policy order. If the site-specified policy dictates a particular priority ordering for jobs, that is the order in which they are run. Strict ordering can be modified by backfilling in order to increase throughput. See "Backfilling".

Subject
A process belonging to a job run by an authorized, unprivileged user (a job submitter.)

Subjob
One of the jobs in a job array, e.g. 1234[7], where 1234[] is the job array itself, and 7 is the index. Queued subjobs are not individually listed in the queue; only their job array is listed. Running subjobs are individually listed.

Subjob index
The unique index which differentiates one subjob from another. This must be a non-negative integer.
Subordinate MoM
Any MoM that is not on the head or first host of a multihost job. A subordinate MoM is directed by the Mother Superiors. Also called a sister.

Task
A process belonging to a job. A POSIX session started by MoM on behalf of a job.

Task placement
The process of choosing a set of vnodes to allocate to a job that will both satisfy the job's resource request (select and place specifications) and satisfy the configured scheduling policy.

Three-server Configuration
One form of redundant license server configuration. Means that if any 2 of the 3 license servers are up and running (referred to as a quorum), the system is functional, with 1 server acting as master who can issue licenses. If the master goes down, another server must take over as master. This is set up as a license file on each of the 3 redundant servers containing:

SERVER <server1> ... <port1>
SERVER <server2> ... <port2>
SERVER <server3> ... <port3>

PBS Professional can point to a license server host that has

Token
Also called “GridWorks Unit”, a unit of value which is checked out from the license server. The number of PBS tokens will be related to the number of CPUs requested by a job that is being executed.

TPP

User
Has two meanings:

1. A person who submits jobs to PBS, as differentiated from Operators, Managers and administrators. See "User" on page 358 in the PBS Professional Administrator’s Guide.
2. A system user, identified by a unique character string (the user name) and by a unique number (the user ID). Any person using the system has a username and user ID.

User access, Access by user
The specified user is allowed access at the server, queues, and reservations.

User ID, UID
A unique numeric identifier assigned to each user.

User limit
Refers to configurable limits on resources and jobs. A limit placed on one or more users, whether generic or individual.

vchunk
The part of a chunk that is supplied by one vnode. If a chunk is broken up across multiple vnodes, each vnode supplies a vchunk.

Version 1 configuration file
Version 2 configuration file

Also called vnodedefs file. Vnode configuration file containing vnode attribute settings. Created using `pbs_mom -s insert` command.

Virtual processor, VP

PBS can treat a vnode as if it has more processors available than the number of physical processors. When `resources_available.ncpus` is set to a number higher than the actual number of physical processors, the vnode can be said to have virtual processors. Also called logical processors.

Vnode

A virtual node, or vnode, is an abstract object representing a set of resources which form a usable part of an execution host. This could be an entire host, or a nodeboard or a blade. A single host can be made up of multiple vnodes. Each vnode can be managed and scheduled independently. Each vnode in a complex must have a unique name. Vnodes can share resources, such as node-locked licenses.

vnodedefs file


vp

Virtual processor. The smallest unit of execution resources that can be specified to run a job. Cray refers to these as a CPU (aka a BASIL PE, an Intel thread or an AMD core).
In this chapter, we describe each PBS command, including any options, operands, etc.

2.1 Requirements for Commands

Some PBS commands require root privilege or PBS Operator or Manager privilege in order to run. Some can be executed by anyone, but the output depends upon the privilege of the user.

Most PBS commands require that the server be running; some require that MoMs be running.

The following table lists the commands, and indicates the permissions required to use each, and whether the server or MoM must be running.

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
<th>Permission Required</th>
<th>Server Must Be Running?</th>
<th>MoM Must Be Running?</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpiexec</td>
<td>Runs MPI programs under PBS on Linux</td>
<td>Any</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>nq2pbs</td>
<td>Converts NQS job scripts to PBS format.</td>
<td>Any</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs</td>
<td>Start, stop, restart, or get the PIDs of PBS daemons</td>
<td>Root on Linux; Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbsdsh</td>
<td>Distributes tasks to vnodes under PBS</td>
<td>Any</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>pbsfs</td>
<td>Show or manipulate PBS fair-share usage data</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>pbsnodes</td>
<td>Query PBS host or vnode status, mark hosts free or offline, change the comment for a host, or output vnode information</td>
<td>Result depends on permission</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>pbsrun</td>
<td>General-purpose wrapper script for mpirun</td>
<td>Root or PBS adminis-trator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbsrun_unwrap</td>
<td>Unwraps mpirun, reversing pbsrun_wrap</td>
<td>Root only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbsrun_wrap</td>
<td>General-purpose script for wrapping mpirun in pbsrun</td>
<td>Root only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_account</td>
<td>For Windows. Manage PBS service account</td>
<td>Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
## Table 2-1: Permission and Daemon Requirements for Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
<th>Permission Required</th>
<th>Server Must Be Running?</th>
<th>MoM Must Be Running?</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbs_attach</td>
<td>Attaches a session ID to a PBS job</td>
<td>Any</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pbs_comm</td>
<td>Starts the PBS communication daemon</td>
<td>Root on Linux; Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_dataservice</td>
<td>Start, stop, or check the status of PBS data service</td>
<td>Root on Linux; Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_ds_password</td>
<td>Sets or changes data service user account or its password</td>
<td>Root on Linux; Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_hostn</td>
<td>Reports hostname and network address(es)</td>
<td>Any</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_idled</td>
<td>Runs PBS daemon that monitors the console and informs pbs_mom of idle time</td>
<td>Root or PBS administrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_iff</td>
<td>Tests authentication with the server</td>
<td>Any; useful only to root server</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>pbs_interactive</td>
<td>For Windows. Register, unregister, or get the version of PBS_INTERACTIVE service</td>
<td>Administrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_lamboot</td>
<td>PBS front end to LAM’s lamboot program</td>
<td>Any</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_migrate_users</td>
<td>Transfers per-user or per-server passwords between PBS servers during a migration upgrade</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>pbs_migration_assistant.bat</td>
<td>For Windows. Tells PBS which server configuration settings to use.</td>
<td>Administrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_mkdirs</td>
<td>For Windows. Create, or fix the permissions of, the directories and files used by PBS</td>
<td>PBS administrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_mom</td>
<td>Runs the PBS job monitoring and execution daemon</td>
<td>Root on Linux; Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>pbs_mpihp</td>
<td>Runs an MPI application in a PBS job with HP MPI</td>
<td>Any</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pbs_mpilam</td>
<td>Runs MPI programs under PBS with LAM MPI</td>
<td>Any</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pbs_mpirun</td>
<td>Runs MPI programs under PBS with MPICH</td>
<td>Any</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>pbs_password</td>
<td>Sets or updates password of a PBS user</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 2-1: Permission and Daemon Requirements for Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
<th>Permission Required</th>
<th>Server Must Be Running?</th>
<th>MoM Must Be Running?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pbs_probe</code></td>
<td><strong>Deprecated.</strong> Reports PBS diagnostic information and fixes permission errors</td>
<td>Root or PBS administrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_python</code></td>
<td>Python interpreter for debugging a hook script from the command line</td>
<td>Any</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_ralter</code></td>
<td>Modify an existing advance or standing reservation</td>
<td>Job owner or PBS administrator</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_rdel</code></td>
<td>Deletes a PBS advance or standing reservation</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_release_nodes</code></td>
<td>Releases vnodes assigned to a PBS job</td>
<td>Job owner, PBS Manager, Operator, administrator, root on Linux, Admin on Windows</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>pbs_rstat</code></td>
<td>Shows status of PBS advance or standing reservations</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_rsub</code></td>
<td>Creates a PBS advance or standing reservation</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_sched</code></td>
<td>Runs a PBS scheduler</td>
<td>Root on Linux; Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_server</code></td>
<td>Starts a PBS batch server</td>
<td>Root on Linux; Admin on Windows</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_snapshot</code></td>
<td>Captures PBS data to be used for diagnostics</td>
<td>Root on Linux</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_tclsh</code></td>
<td>TCL shell with TCL-wrapped PBS API</td>
<td>Any</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_tmrsh</code></td>
<td>TM-enabled replacement for <code>rsh/ssh</code> for use by MPI implementations</td>
<td>Any</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><code>pbs_topologyinfo</code></td>
<td>Reports topological information used for licensing purposes</td>
<td>Root or Windows administrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>pbs_wish</code></td>
<td>TK window shell with TCL-wrapped PBS API</td>
<td>Any</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>printjob</code></td>
<td>Prints job information</td>
<td>Root or Windows Administrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>qalter</code></td>
<td>Alters a PBS job</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qdel</code></td>
<td>Deletes PBS jobs</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qdisable</code></td>
<td>Prevents a queue from accepting jobs</td>
<td>Manager or Operator only</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
## Table 2-1: Permission and Daemon Requirements for Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
<th>Permission Required</th>
<th>Server Must Be Running?</th>
<th>MoM Must Be Running?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>qenable</code></td>
<td>Allows a queue to accept jobs</td>
<td>Manager or Operator only</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qhold</code></td>
<td>Holds PBS batch jobs</td>
<td>Some holds can be set by Operator, Manager, root, or administrator only</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qmgr</code></td>
<td>Administrator’s command interface for managing PBS</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qmove</code></td>
<td>Moves a PBS job from one queue to another</td>
<td>Any; managers and operators can move jobs in some cases where unprivileged users cannot</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qmsg</code></td>
<td>Writes message string into one or more job output files</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qorder</code></td>
<td>Swaps queue positions of two PBS jobs</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qrerun</code></td>
<td>Requeues a PBS job</td>
<td>Manager or Operator only</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qrls</code></td>
<td>Releases holds on PBS jobs</td>
<td>Some holds can be released by Operator, Manager, root, or administrator only</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qrun</code></td>
<td>Runs a PBS job immediately</td>
<td>Operator or Manager only</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qselect</code></td>
<td>Selects specified PBS jobs</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qsig</code></td>
<td>Send signal to PBS job</td>
<td>Operator or Manager required to send <code>admin-suspend</code>, <code>admin-resume</code>, <code>suspend</code>, and <code>resume</code>. Any privilege for other signals.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>qstart</code></td>
<td>Turns on scheduling or routing for the jobs in a PBS queue</td>
<td>Operator or Manager only</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qstat</code></td>
<td>Displays status of PBS jobs, queues, or servers</td>
<td>Result depends on permission</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qstop</code></td>
<td>Prevents PBS jobs in the specified queue from being scheduled or routed</td>
<td>Operator or Manager only</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
### Table 2-1: Permission and Daemon Requirements for Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
<th>Permission Required</th>
<th>Server Must Be Running?</th>
<th>MoM Must Be Running?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>qsub</code></td>
<td>Submits a job to PBS</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>qterm</code></td>
<td>Terminates one or both PBS servers, and optionally terminates scheduler and/or MoMs</td>
<td>Operator or Manager only</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>tracejob</code></td>
<td>Extracts and prints log messages for a PBS job</td>
<td>Root or PBS adminstrator only</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>win_postinstall.py</code></td>
<td>For Windows. Configures PBS services</td>
<td>Administrator</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>xpbs</code></td>
<td><strong>Deprecated.</strong> GUI front end to PBS commands</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>xbssmon</code></td>
<td><strong>Deprecated.</strong> PBS GUI for displaying and monitoring execution hosts</td>
<td>Any; more information available to root</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

### 2.1.1 Windows Requirements

Under Windows, use double quotes when specifying arguments to PBS commands.
2.2 mpiexec

Runs MPI programs under PBS on Linux

2.2.1 Synopsis

`mpiexec`

`mpiexec --version`

2.2.2 Description

The PBS `mpiexec` command provides the standard `mpiexec` interface on a system running supported versions of HPE MPI. If executed on a different system, it will assume it was invoked by mistake. In this case it will use the value of `PBS_O_PATH` to search for the correct `mpiexec`. If one is found, the PBS `mpiexec` will exec it.

The PBS `mpiexec` calls the HPE `mpirun(1)`. The name of the array to use when invoking `mpirun` is user-specifiable via the `PBS_MPI_SGIARRAY` environment variable.

It is transparent to the user; MPI jobs submitted outside of PBS run as they would normally. MPI jobs can be launched across multiple HPE SGI systems. PBS will manage, track, and cleanly terminate multi-host MPI jobs. PBS users can run MPI jobs within specific partitions.

If CSA has been configured and enabled, PBS will collect accounting information on all tasks launched by an MPI job. CSA information will be associated with the PBS job ID that invoked it, on each execution host.

If the `PBS_MPI_DEBUG` environment variable’s value has a nonzero length, PBS writes debugging information to standard output.

2.2.3 Usage

The PBS `mpiexec` command presents the `mpiexec` interface described in section “4.1 Portable MPI Process Startup” of the “MPI-2: Extensions to the Message-Passing Interface” document in [http://www.mpiforum.org/docs/mpi-20-html/node42.htm](http://www.mpiforum.org/docs/mpi-20-html/node42.htm)

2.2.4 Options

`--version`

The `mpiexec` command returns its PBS version information and exits. This option can only be used alone.

2.2.5 Requirements

- System running a supported version of HPE MPI.
- PBS uses HPE SGI’s `mpirun(1)` command to launch MPI jobs. HPE SGI’s `mpirun` must be in the standard location.
- The location of `pbs_attach()` on each vnode of a multi-vnode MPI job must be the same as it is on the mother superior vnode.
- In order to run multihost jobs, the HPE SGI Array Services must be correctly configured. HPE SGI systems communicating via HPE SGI’s Array Services must all use the same version of the `sgi-arraysvcs` package. HPE SGI systems communicating via HPE SGI’s Array Services must have been configured to interoperate with each other using the default array. See HPE SGI’s `array_services(5)` man page.
2.2.6 Environment Variables

PBS_CPUSET_DEDICATED
The PBS mpiexec script sets the PBS_CPUSET_DEDICATED environment variable to assert exclusive use of the resources in the assigned cpuset.

PBS_ENVIRONMENT
The PBS_ENVIRONMENT environment variable is used to determine whether mpiexec is being called from within a PBS job.

PBS_MPI_DEBUG
The PBS mpiexec checks the PBS_MPI_DEBUG environment variable. If this variable has a nonzero length, debugging information is written.

PBS_MPI_SGIARRAY
If the PBS_MPI_SGIARRAY environment variable is present, the PBS mpiexec will use its value as the name of the array to use when invoking mpirun.

PBS_O_PATH
The PBS mpiexec uses the value of PBS_O_PATH to search for the correct mpiexec if it was invoked by mistake.

2.2.7 Path

PBS' mpiexec is located in PBS_EXEC/bin/mpiexec.

2.2.8 See Also

The PBS Professional Administrator's Guide, "pbs_attach" on page 55
HPE SGI man pages: HPE SGI's mpirun(1), HPE SGI's mpiexec_mpt(1), HPE SGI's array_services(5)
2.3  nqs2pbs

Converts NQS job scripts to PBS format

2.3.1  Synopsis

nqs2pbs <NQS script> [PBS script]
nqs2pbs --version

2.3.2  Description

This utility converts an existing NQS job script to work with PBS and NQS. The existing script is copied and PBS directives using #PBS are inserted prior to each NQS directive (#QSUB or #@$) in the original script.

2.3.2.1  Specifying Time or Date

PBS will interpret a time specification without a date in the following way:

- If the time specified has not yet been reached, the job will become eligible to run at that time today.
- If the specified time has already passed when the job is submitted, the job will become eligible to run at that time tomorrow.

This command does not support time zone identifiers. All times are taken as local time.

Converting NQS date specifications to the PBS form may result in a warning message and an incompletely converted date. PBS does not support date specifications of “today”, “tomorrow”, or the name of the days of the week such as “Monday”. If any of these are encountered in a script, the PBS specification will contain only the time portion of the NQS specification, i.e. #PBS -a <hhmm[.ss]>.

It is suggested that you specify the execution time on the qsub command line rather than in the script.

Certain NQS date specifications and options are not supported by PBS. A warning message is displayed indicating the problem and the line of the script on which it occurred.

2.3.3  Options

--version

The nqs2pbs command returns its PBS version information and exits. This option must be used alone.

2.3.4  Operands

NQS script

Specifies the file name of the NQS script to convert. This file is not changed.

PBS script

If specified, this is the name of the new PBS script. If not specified, the new file name is nqs_script.new

2.3.5  Errors

If any unrecognizable NQS directives are encountered, an error message is displayed. The new PBS script is deleted if any errors occur.
2.3.6 See Also

"qsub" on page 207
2.4  

**pbs**

Start, stop, restart, or get the PIDs of PBS daemons

### 2.4.1  Synopsis

```bash
pbs [start | stop | restart | status]
```

### 2.4.2  Description

The `pbs` command starts, stops or restarts all PBS daemons on the local machine, or reports the PIDs of all daemons when given the `status` argument. Does not affect other hosts.

You can start, stop, restart, or status the PBS daemons using the `systemctl` command; see “Starting & Stopping PBS” on page 129 in the PBS Professional Installation & Upgrade Guide.

#### 2.4.2.1  Caveats

This command operates only on daemons that are marked as active in `pbs.conf`. For example, if `PBS_START_MOM` is set to 0 in the local `pbs.conf`, this command will not operate on `pbs_mom`, and will not start, stop, or restart `pbs_mom`.

This command is typically placed in `/etc/init.d` so that PBS starts up automatically.

#### 2.4.2.2  Required Privilege

Root privilege is required to use this command.

### 2.4.3  Arguments

- **restart**
  
  All daemons on the local machine are stopped, then they are restarted. PBS reports the name of the license server and the number and type of licenses available.

- **start**
  
  Each daemon on the local machine is started. PBS reports the number and type of licenses available, as well as the name of the license server. Any running jobs are killed.

- **status**
  
  PBS reports the PID of each daemon on the local machine.

- **stop**
  
  Each daemon on the local machine is stopped, and its PID is reported.

### 2.4.4  See Also

The PBS Professional Administrator's Guide, "pbs_comm" on page 57, "pbs_mom" on page 72, "pbs_sched" on page 104, "pbs_server" on page 106
2.5  pbsdsh

Distributes tasks to vnodes under PBS

2.5.1  Synopsis

```bash
pbsdsh [-c <copies>] [-s] [-v] [-o] -- <program> [<program args>]

pbsdsh [-n <vnode index>] [-s] [-v] [-o] -- <program> [<program args>]

pbsdsh --version
```

2.5.2  Description of pbsdsh Command

The `pbsdsh` command allows you to distribute and execute a task on each of the vnodes assigned to your job by executing (spawning) the application on each vnode. The `pbsdsh` command uses the PBS Task Manager, or TM, to distribute the program on the allocated vnodes.

When run without the `-c` or the `-n` option, `pbsdsh` will spawn the program on all vnodes allocated to the PBS job. The spawns take place concurrently; all execute at (about) the same time.

Note that the double dash must come after the options and before the program and arguments. The double dash is only required for Linux.

The `pbsdsh` command runs one task for each line in the `$PBS_NODEFILE`. Each MPI rank gets a single line in the `$PBS_NODEFILE`, so if you are running multiple MPI ranks on the same host, you still get multiple `pbsdsh` tasks on that host.

2.5.2.1  Example

The following example shows the `pbsdsh` command inside of a PBS batch job. The options indicate that the user wants `pbsdsh` to run the `myapp` program with one argument (`app-arg1`) on all four vnodes allocated to the job (i.e. the default behavior).

```bash
#!/bin/sh
#PBS -l select=4:ncpus=1
#PBS -l walltime=1:00:00
pbsdsh ./myapp app-arg1
```

2.5.3  Options to pbsdsh Command

- `-c <copies>`
  
  The program is spawned `copies` times on the vnodes allocated, one per vnode, unless `copies` is greater than the number of vnodes. If `copies` is greater than the number of vnodes, it wraps around, running multiple instances on some vnodes. This option is mutually exclusive with `-n`.

- `-n <vnode index>`
  
  The program is spawned only on a single vnode, which is the `vnode index`-th vnode allocated. This option is mutually exclusive with `-c`.

- `-o`
  
  No obit request is made for spawned tasks. The program does not wait for the tasks to finish.
The program is run in turn on each vnode, one after the other.

- Produces verbose output about error conditions and task exit status.

--version
The `pbsdsh` command returns its PBS version information and exits. This option can only be used alone

### 2.5.4 Operands

- **program**
  The first operand, `program`, is the program to execute. The double dash must precede `program` under Linux.

- **program args**
  Additional operands, `program args`, are passed as arguments to the program.

### 2.5.5 Standard Error

The `pbsdsh` command writes a diagnostic message to standard error for each error occurrence.

### 2.5.6 See Also

2.6  pbsfs

Show or manipulate PBS fairshare usage data

2.6.1  Synopsis

Showing usage data:

pbsfs [-c <entity1> <entity2>] [-g <entity>] [-I <scheduler name>] [-p] [-t]

Manipulating usage data:

pbsfs [-d] [-e] [-I <scheduler name>] [-s <entity> <usage value>]

Printing version:

pbsfs --version

2.6.2  Description

You can use the pbsfs command to print or manipulate a PBS scheduler’s fairshare usage data. You can print the usage data in various formats, described below. Changes made using pbsfs take effect in the next scheduling cycle; you do not need to restart or HUP a scheduler for changes to take effect.

We recommend that if you use the options that manipulate usage data, you should do this when a scheduler is not scheduling jobs, because scheduling while changing fairshare usage data may give unwanted results.

2.6.2.1  Prerequisites

The server must be running in order to use the pbsfs command.

2.6.2.2  Permissions

You must be root to run the pbsfs command; if not, it will print the error message, “Unable to access fairshare data”.

2.6.3  Options to pbsfs

You can safely use the following options while jobs are being scheduled:

(no options)

Same as pbsfs -p.

-c <entity1> <entity2>

Compares two fairshare entities.

-g <entity>

Prints a detailed listing for the specified entity, including the path from the root of the tree to the entity.

-I <scheduler name>

Specifies name of scheduler whose data is to be manipulated or shown. Required for multischeds; optional for default scheduler. Name of default scheduler is “default”. If not specified, pbsfs operates on default scheduler.
PBS Commands

Chapter 2

- p
  Prints the fairshare tree as a table, showing for each internal and leaf vertex the group ID of the vertex’s parent, group ID of the vertex, vertex shares, vertex usage, and percent of shares allotted to the vertex.

- t
  Prints the fairshare tree in a hierarchical format.

--version
  The pbsfs command returns its PBS version information and exits. This option can only be used alone.

It is not recommended to be scheduling jobs when you use the following options:

- d
  Decays the fairshare tree by the amount specified in the fairshare_decay_factor scheduler parameter.

- e
  Trims fairshare tree to just the entities in the resource_group file. Unknown entities and their usage are deleted; as a result the unknown group has no usage and no children.

- s <entity> <usage value>
  Sets entity’s usage value to usage value. Editing a non-leaf entity is ignored. All non-leaf entity usage values are calculated each time you use the pbsfs command to make changes.

2.6.3.1 Output Formats for pbsfs

The pbsfs command can print output in three different formats:

**pbsfs -g <entity>**

Prints a detailed listing for the specified entity. Example:

```
pbsfs -g pbsuser3
fairshare entity: pbsuser3
  Resgroup: 20
  cresgroup: 22
  Shares: 40
  Percentage: 24.000000%
  fairshare_tree_usage: 0.832973
  usage: 1000 (cput)
  usage/perc: 4167
Path from root:
TREEROOT : 0 1201 / 1.000 = 1201
  group2 : 20 1001 / 0.600 = 1668
  pbsuser3 : 22 1000 / 0.240 = 4167
```
**pbsfs.**

**pbsfs -p**

Prints the entire tree as a table, with data in columns. Example:

```
pbsfs
```

Fairshare usage units are in: cput

```
TREEROOT : Grp: -1 cgrp: 0 Shares: -1 Usage: 1201 Perc: 100.000%
group2 : Grp: 0 cgrp: 20 Shares: 60 Usage: 1001 Perc: 60.000%
pbsuser3 : Grp: 20 cgrp: 22 Shares: 40 Usage: 1000 Perc: 24.000%
pbsuser2 : Grp: 20 cgrp: 21 Shares: 40 Usage: 1 Perc: 36.000%
group1 : Grp: 0 cgrp: 10 Shares: 40 Usage: 201 Perc: 40.000%
pbsuser1 : Grp: 10 cgrp: 12 Shares: 50 Usage: 100 Perc: 20.000%
pbsuser : Grp: 10 cgrp: 11 Shares: 50 Usage: 100 Perc: 20.000%
unknown : Grp: 0 cgrp: 1 Shares: 0 Usage: 1 Perc: 0.000%
```

**pbsfs -t**

Prints the entire tree as a tree, showing group-child relationships. Example:

```
pbsfs -t
```

```
TREEROOT(0)
group2(20)
   pbsuser3(22)
   pbsuser2(21)
group1(10)
   pbsuser1(12)
   pbsuser(11)
unknown(1)
```

### 2.6.3.2 Data Output by pbsfs

**cresgroup, cgrp**

Group ID of the entity

**fairshare entity**

The specified fairshare tree entity

**fairshare usage units**

The resource for which a scheduler accumulates usage for fairshare calculations. This defaults to **cput** (CPU seconds) but can be set in a scheduler’s configuration file.

**fairshare_tree_usage**

The entity’s effective usage. See "Computing Effective Usage (fairshare_tree_usage)" on page 142 in the PBS Professional Administrator’s Guide.

**Path from root**

The path from the root of the tree to the entity. A scheduler follows this path when comparing priority between two entities.

**Percentage, perc**

The percentage of the shares in the tree allotted to the entity, computed as **fairshare_perc**. See "Computing Target Usage for Each Vertex (fairshare_perc)" on page 141 in the PBS Professional Administrator’s Guide.

**Resgroup, Grp**

Group ID of the entity’s parent group
Chapter 2

PBS Commands

Shares
The number of shares allotted to the entity

usage
The amount of usage by the entity

usage/perc
The value a scheduler uses to pick which entity has priority over another. The smaller the number the higher the priority.

2.6.4 See Also

"Using Fairshare" on page 136 in the PBS Professional Administrator’s Guide.
2.7  pbsnodes

Query PBS host or vnode status, mark hosts free or offline, change the comment for a host, or output vnode information.

2.7.1  Synopsis

```
pbsnodes [ -o | -r ] [-s <server name>] [-C <comment>] <hostname> <hostname> ...
pbsnodes [-l] [-s <server name>]
pbsnodes -v <vnode> <vnode> ...
pbsnodes -a [v] [-S [j][L]] [-F json|dsv [-D <delimiter>]] [-s <server name>]
pbsnodes -H [-S [j][L]] [-F json|dsv [-D <delimiter>]] <hostname> <hostname> ...
pbsnodes --version
```

2.7.2  Description

The `pbsnodes` command is used to query the status of hosts or vnodes, to mark hosts `FREE` or `OFFLINE`, to edit a host’s `comment` attribute, or to output vnode information. The `pbsnodes` command obtains host information by sending a request to the PBS server.

2.7.2.1  Using pbsnodes

To print the status of the specified host or hosts, run `pbsnodes` with no options (except the `-s` option) and with a list of hosts.

To print the command usage, run `pbsnodes` with no options and without a list of hosts.

To remove a vnode from the scheduling pool, mark it `OFFLINE`. If it is marked `DOWN`, when the server next queries the MoM, and can connect, the vnode will be marked `FREE`.

To offline a single vnode in a multi-vnoded system, use:

```
qmgr -c "set node <vnode name> state=offline"
```

2.7.2.2  Output

The order in which hosts or vnodes are listed in the output of the `pbsnodes` command is undefined. Do not rely on output being ordered.

If you print attributes, `pbsnodes` prints out only those attributes which are not at default values.

2.7.2.3  Permissions

PBS Manager or Operator privilege is required to execute `pbsnodes` with the `-o` or `-r` options.

Operator or Manager privilege is required to view custom resources which have been created to be invisible to users, and to see some output such as PBS version.

2.7.3  Options to pbsnodes

(no options)

If neither options nor a host list is given, the `pbsnodes` command prints usage syntax.
-a
Lists all hosts and all their attributes (available and used.)
When listing a host with multiple vnodes:
The output for the jobs attribute lists all the jobs on all the vnodes on that host. Jobs that run on more than
one vnode will appear once for each vnode they run on.
For consumable resources, the output for each resource is the sum of that resource across all vnodes on that
host.
For all other resources, e.g. string and Boolean, if the value of that resource is the same on all vnodes on
that host, the value is returned. Otherwise the output is the literal string “<various>”.

-C <comment>
Sets the comment attribute for the specified host(s) to the value of comment. Comments containing spaces
must be quoted. The comment string is limited to 80 characters. Usage:

\texttt{pbsnodes -C \textit{comment} \textit{hostname} [\textit{hostname} ...]}
To set the comment for a vnode:

\texttt{qmgr -c "s n \textit{vnode name} comment=\textit{comment}"}

-F dsv [-D <delimiter>]
Prints output in delimiter-separated value format. Optional delimiter specification. Default delimiter is vertical
bar (“|”).

-F json
Prints output in JSON format.

-H <hostname> [\textit{hostname} ...]
Prints all non-default-valued attributes for specified hosts and all vnodes on specified hosts.
-j

Displays the following job-related headers for specified vnodes:

### Table 2-2: Output for -j Option

<table>
<thead>
<tr>
<th>Header</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnode</td>
<td>15</td>
<td>Vnode name</td>
</tr>
<tr>
<td>state</td>
<td>15</td>
<td>Vnode state</td>
</tr>
<tr>
<td>njobs</td>
<td>6</td>
<td>Number of jobs on vnode</td>
</tr>
<tr>
<td>run</td>
<td>5</td>
<td>Number of running jobs at vnode</td>
</tr>
<tr>
<td>susp</td>
<td>6</td>
<td>Number of suspended jobs at vnode</td>
</tr>
<tr>
<td>mem f/t</td>
<td>12</td>
<td>Vnode memory free/total</td>
</tr>
<tr>
<td>ncpus f/t</td>
<td>7</td>
<td>Number of CPUs at vnode free/total</td>
</tr>
<tr>
<td>nmics f/t</td>
<td>7</td>
<td>Number of MICs free/total</td>
</tr>
<tr>
<td>ngpus f/t</td>
<td>7</td>
<td>Number of GPUs at vnode free/total</td>
</tr>
<tr>
<td>jobs</td>
<td>No restriction</td>
<td>List of job IDs on vnode</td>
</tr>
</tbody>
</table>

Note that nmics and ngpus are custom resources that must be created by the administrator if you want them displayed here.

Each subjob is treated as a unique job.

-L

Displays output with no restrictions on column width.

-l

Lists all hosts marked as DOWN or OFFLINE. Each such host’s state and comment attribute (if set) is listed. If a host also has state STATE-UNKNOWN, it is listed. For hosts with multiple vnodes, only hosts where all vnodes are marked as DOWN or OFFLINE are listed.

-o <hostname> [ <hostname> ...]

Marks listed hosts as OFFLINE even if currently in use. This is different from being marked DOWN. A host that is marked OFFLINE continues to execute the jobs already on it, but is removed from the scheduling pool (no more jobs are scheduled on it.)

Offlines ALL compute nodes on a Cray.

For non-Cray hosts with multiple vnodes, pbsnodes operates on a host and all of its vnodes, where the host-name is resources_available.host, which is the name of the natural vnode.

To offline all vnodes on a non-Cray multi-vnoded machine:

    pbsnodes -o <name of natural vnode>

For Cray hosts, note that login nodes share a name with compute nodes, so if you use this option on a Cray, you will offline ALL of your compute nodes.

To offline a single vnode on a Cray or another multi-vnoded system, use:

    Qmgr: qmgr -c "set node <vnode name> state=offline"

Requires PBS Manager or Operator privilege.
PBS Commands

-\ r\ <hostname> [\<hostname> ...]
  Clears OFFLINE from listed hosts.

-\ S
  Displays the following vnode information:

<table>
<thead>
<tr>
<th>Header</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>15</td>
<td>Vnode name</td>
</tr>
<tr>
<td>state</td>
<td>15</td>
<td>Vnode state</td>
</tr>
<tr>
<td>OS</td>
<td>8</td>
<td>Value of OS custom resource, if any</td>
</tr>
<tr>
<td>hardware</td>
<td>8</td>
<td>Value of hardware custom resource, if any</td>
</tr>
<tr>
<td>host</td>
<td>15</td>
<td>Hostname</td>
</tr>
<tr>
<td>queue</td>
<td>10</td>
<td>Value of vnode’s queue attribute</td>
</tr>
<tr>
<td>ncpus</td>
<td>7</td>
<td>Number of CPUs at vnode</td>
</tr>
<tr>
<td>nmics</td>
<td>7</td>
<td>Number of MICs at vnode</td>
</tr>
<tr>
<td>mem</td>
<td>8</td>
<td>Vnode memory</td>
</tr>
<tr>
<td>ngpus</td>
<td>7</td>
<td>Number of GPUs at vnode</td>
</tr>
<tr>
<td>comment</td>
<td>No restriction</td>
<td>Vnode comment</td>
</tr>
</tbody>
</table>

Note that nmics, ngpus and OS are custom resources that must be created by the administrator if you want their values displayed here.

\ s\ <server name>
  Specifies the PBS server to which to connect.

\ v\ [<vnode> [<vnode> ...]]
  Lists all non-default-valued attributes for each specified vnode.
  With no arguments, prints one entry for each vnode in the PBS complex.
  With one or more vnodes specified, prints one entry for each specified vnode.

--version
  The \texttt{pbsnodes} command returns its PBS version information and exits. This option can only be used alone.

2.7.4 Operands

<server name>
  Specifies the server to which to connect. Default: default server.

<hostname> [\<hostname> ...]
  Specifies the host(s) to be queried or operated on.

<vnode> [<vnode> ...]
  Specifies the vnode(s) to be queried or operated on.
2.7.5  Exit Status

Zero
   Success

Greater than zero
   • Incorrect operands are given
   • pbsnodes cannot connect to the server
   • There is an error querying the server for the vnodes

2.7.6  See Also

The PBS Professional Administrator’s Guide, "qmgr" on page 146
2.8 **pbsrun**

General-purpose wrapper script for `mpirun`

### 2.8.1 Synopsis

```bash
pbsrun
pbsrun --version
```

### 2.8.2 Description

`pbsrun` is a wrapper script for any of several versions of `mpirun`. This provides a user-transparent way for PBS to control jobs which call `mpirun` in their job scripts. The `pbsrun_wrap` script instantiates `pbsrun` so that the wrapper script for the specific version of `mpirun` being used has the same name as that version of `mpirun`.

If the `mpirun` wrapper script is run inside a PBS job, it translates any `mpirun` call of the form:

```
mpirun [<options>] <executable> [<args>]
```

into

```
mpirun [<options>] pbs_attach [<special options to pbs_attach>] <executable> [<args>]
```

where `special options` refers to any option needed by `pbs_attach` to do its job (e.g. `-j $PBS_JOBID`).

If the wrapper script is executed outside of PBS, a warning is issued about “not running under PBS”, but it proceeds as if the actual program had been called in standalone fashion.

The `pbsrun` wrapper script is not meant to be executed directly; instead it is instantiated by `pbsrun_wrap`. It is copied to the target directory and renamed “`pbsrun.<mpirun version/flavor>`” where `mpirun version/flavor` is a string that identifies the `mpirun` version being wrapped (e.g. `ch_gm`).

The `pbsrun` script, if executed inside a PBS job, runs an initialization script, named `$PBS_EXEC/lib/MPI/pbsrun.<mpirun version/flavor>.init`, then parses `mpirun`-like arguments from the command line, sorting which options and option values to retain, to ignore, or to transform, before calling the actual `mpirun` script with a “`pbs_attach`” prefixed to the executable. The actual `mpirun` to call is found by tracing the link pointed to by `$PBS_EXEC/lib/MPI/pbsrun.<mpirun version/flavor>.link`.

For all of the wrapped MPIs, the maximum number of ranks that can be launched is the number of entries in `$PBS_NODEFILE`.

The wrapped MPIs are:

- MPICH-GM’s `mpirun` (`mpirun.ch_gm`) with rsh/ssh (The wrapper is `deprecated` as of 14.2.1)
- MPICH-MX’s `mpirun` (`mpirun.ch_mx`) with rsh/ssh (The wrapper is `deprecated` as of 14.2.1)
- MPICH-GM’s `mpirun` (`mpirun.mpd`) with MPD (The wrapper is `deprecated` as of 14.2.1)
- MPICH-MX’s `mpirun` (`mpirun.mpd`) with MPD (The wrapper is `deprecated` as of 14.2.1)
- MPICH2’s `mpirun`
- Intel MPI’s `mpirun` (The wrapper is `deprecated` as of 13.0)
- MVAPICH1’s `mpirun` (The wrapper is `deprecated` as of 14.2.1)
- MVAPICH2’s `mpiexec`
- IBM’s `poe`
2.8.3 Options

--version

The pbsrun command returns its PBS version information and exits. This option can only be used alone.

2.8.4 Initialization Script

The initialization script, called $PBS_EXEC/lib/mpi/pbsrun.<mpirun version/flavor>.init, where mpirun version/flavor reflects the mpirun flavor/version being wrapped, can be modified by an administrator to customize against the local flavor/version of mpirun being wrapped.

Inside this sourced init script, 8 variables are set:

options_to_retain="-optA -optB <val> -optC <val1> val2> ..."
options_to_ignore="-optD -optE <n> -optF <val1> val2> ..."
options_to_transform="-optG -optH <val> -optI <val1> val2> ..."
options_to_fail="-optY -optZ ...

options_to_configfile="-optX <val> ...

options_with_another_form="-optW <val> ...
pbs_attach=pbs_attach

options_to_pbs_attach="-J $PBS_JOBID"

2.8.4.1 Initialization Script Options

options_to_retain

Space-separated list of options and values that pbsrun.<mpirun version/flavor> passes on to the actual mpirun call. Options must begin with “-” or “--”, and option arguments must be specified by some arbitrary name with left and right arrows, as in “<val>”.

options_to_ignore

Space-separated list of options and values that pbsrun.<mpirun version/flavor> does not pass on to the actual mpirun call. Options must begin with “-” or “--”, and option arguments must be specified by arbitrary names with left and right arrows, as in “<n>”.

options_to_transform

Space-separated list of options and values that pbsrun modifies before passing on to the actual mpirun call.

options_to_fail

Space-separated list of options that will cause pbsrun to exit upon encountering a match.

options_to_configfile

Single option and value that refers to the name of the configuration file containing command line segments found in certain versions of mpirun.

options_with_another_form

Space-separated list of options and values that can be found in options_to_retain, options_to_ignore, or options_to_transform, whose syntax has an alternate, unsupported form.

pbs_attach

Path to pbs_attach, which is called before the executable argument of mpirun.

options_to_pbs_attach

Special options to pass to the pbs_attach call. You may pass variable references (e.g. $PBS_JOBID) and they are substituted by pbsrun to actual values.
If `pbsrun` encounters any option not found in `options_to_retain`, `options_to_ignore`, and `options_to_transform`, it is flagged as an error.

These functions are created inside the `init` script. These can be modified by the PBS administrator.

```plaintext
transform_action() {
    # passed actual values of $options_to_transform
    args=$*
}

boot_action() {
    mpirun_location=$1
}

evaluate_options_action() {
    # passed actual values of transformed options
    args=$*
}

configfile_cmdline_action() {
    args=$*
}

default_action() {
    mpirun_location=$1
}

transform_action()
    The `pbsrun.<mpirun_version/flavor>` wrapper script invokes the function `transform_action()` (called once on each matched item and value) with actual options and values received matching one of the `options_to_transform`. The function returns a string to pass on to the actual `mpirun` call.

boot_action()
    Performs any initialization tasks needed before running the actual `mpirun` call. For instance, GM’s MPD requires the MPD daemons to be user-started first. This function is called by the `pbsrun.<mpirun_version/flavor>` script with the location of actual `mpirun` passed as the first argument. Also, the `pbsrun.<mpirun_version/flavor>` checks for the exit value of this function to determine whether or not to progress to the next step.

evaluate_options_action()
    Called with the actual options and values that resulted after consulting `options_to_retain`, `options_to_ignore`, `options_to_transform`, and executing `transform_action()`. This provides one more chance for the script writer to evaluate all the options and values in general, and make any necessary adjustments, before passing them on to the actual `mpirun` call. For instance, this function can specify what the default value is for a missing `-np` option.
configfile_cmdline_action()

Returns the actual options and values to be put in before the option_to_configfile parameter.

configfile_firstline_action()

Returns the item that is put in the first line of the configuration file specified in the option_to_configfile parameter.

eend_action()

Called by pbsrun.<mpirun version/flavor> at the end of execution. It undoes any action done by transform_action(), such as cleanup of temporary files. It is also called when pbsrun.<mpirun version/flavor> is prematurely killed. This function is called with the location of actual mpirun passed as first argument.

The actual mpirun program to call is the path pointed to by $PBS_EXEC/lib/MPI/pbsrun.<mpirun version/flavor>.link.

2.8.4.2 Modifying *.init Scripts

In order for administrators to modify *.init scripts without breaking package verification in RPM, master copies of the initialization scripts are named *.init.in. pbsrun_wrap instantiates the *.init.in files as *.init. For instance, $PBS_EXEC/lib/MPI/pbsrun.mpich2.init.in is the master copy, and pbsrun_wrap instantiates it as $PBS_EXEC/lib/MPI/pbsrun.mpich2.init. pbsrun_unwrap takes care of removing the *.init files.

2.8.5 Versions/Flavors of mpirun

2.8.5.1 MPICH-GM mpirun (mpirun.ch_gm) with rsh/ssh:

`pbsrun.ch_gm`

2.8.5.1.i Syntax

`pbsrun.ch_gm <options> <executable> <arg1> <arg2> ... <argn>`

Deprecated. The PBS wrapper script to MPICH-GM's mpirun (mpirun.ch_gm) with rsh/ssh process startup method is named pbsrun.ch_gm.

If executed inside a PBS job, this allows for PBS to track all MPICH-GM processes started by rsh/ssh so that PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard mpirun.ch_gm were used.

2.8.5.1.ii Options Handling

If executed inside a PBS job script, all mpirun.ch_gm options given are passed on to the actual mpirun call with these exceptions:

-`-machinefile <file>`
  
  The file argument contents are ignored and replaced by the contents of $PBS_NODEFILE.

-`-np`

  If not specified, the number of entries found in $PBS_NODEFILE is used.

-`-pg`

  The use of the -pg option, for having multiple executables on multiple hosts, is allowed but it is up to the user to make sure only PBS hosts are specified in the process group file; MPI processes spawned are not guaranteed to be under the control of PBS.
2.8.5.1.iii Wrap/Unwrap

To wrap MPICH-GM's mpirun script:

```bash
# pbsrun_wrap [MPICH-GM_BIN_PATH]/mpirun.ch_gm pbsrun.ch_gm
```

To unwrap MPICH-GM's mpirun script:

```bash
# pbsrun_unwrap pbsrun.ch_gm
```

2.8.5.2 MPICH-MX mpirun (mpirun.ch_mx) with rsh/ssh: pbsrun.ch_mx

2.8.5.2.i Syntax

```bash
pbsrun.ch_mx <options> <executable> <arg1> <arg2> ... <argn>
```

The wrapper is deprecated. The PBS wrapper script to MPICH-MX's mpirun (mpirun.ch_mx) with rsh/ssh process startup method is named pbsrun.ch_mx.

If executed inside a PBS job, this allows PBS to track all MPICH-MX processes started by rsh/ssh so that PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard mpirun.ch_mx were used.

2.8.5.2.ii Options HANDLING

If executed inside a PBS job script, all mpirun.ch_gm options given are passed on to the actual mpirun call with some exceptions:

- `-machinefile <file>`
  The `file` argument contents is ignored and replaced by the contents of `$PBS_NODEFILE`.

- `-np`
  If not specified, the number of entries found in `$PBS_NODEFILE` is used.

- `-pg`
  The use of the `-pg` option, for having multiple executables on multiple hosts, is allowed but it is up to the user to make sure only PBS hosts are specified in the process group file; MPI processes spawned are not guaranteed to be under the control of PBS.

2.8.5.2.iii Wrap/Unwrap

To wrap MPICH-MX's mpirun script:

```bash
# pbsrun_wrap [MPICH-MX_BIN_PATH]/mpirun.ch_mx pbsrun.ch_mx
```

To unwrap MPICH-MX's mpirun script:

```bash
# pbsrun_unwrap pbsrun.ch_mx
```

2.8.5.3 MPICH-GM mpirun (mpirun.mpd) with MPD: pbsrun.gm_mpd

2.8.5.3.i Syntax

```bash
pbsrun.gm_mpd <options> <executable> <arg1> <arg2> ... <argn>
```

The wrapper is deprecated. The PBS wrapper script to MPICH-GM's mpirun (mpirun.ch_gm) with MPD process startup method is called pbsrun.gm_mpd.

If executed inside a PBS job, this allows PBS to track all MPICH-GM processes started by the MPD daemons so that PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard mpirun.ch_gm with MPD were used.
2.8.5.3.ii  Options Handling

If executed inside a PBS job script, all mpirun.ch_gm with MPD options given are passed on to the actual mpirun call with these exceptions:

- **m <file>**
  The file argument contents are ignored and replaced by the contents of $PBS_NODEFILE.

- **np**
  If not specified, the number of entries found in $PBS_NODEFILE is used.

- **pg**
  The use of the -pg option, for having multiple executables on multiple hosts, is allowed but it is up to the user to make sure only PBS hosts are specified in the process group file; MPI processes spawned are not guaranteed to be under the control of PBS.

2.8.5.3.iii  Startup/Shutdown

The script starts MPD daemons on each of the unique hosts listed in $PBS_NODEFILE, using either rsh or ssh based on the value of the environment variable RSHCOMMAND. The default is rsh.

The script also takes care of shutting down the MPD daemons at the end of a run.

2.8.5.3.iv  Wrap/Unwrap

To wrap MPICH-GM's mpirun script with MPD:

```
# pbsrun_wrap [MPICH-GM_BIN_PATH]/mpirun.mpd pbsrun.gm_mpd
```

To unwrap MPICH-GM's mpirun script with MPD:

```
# pbsrun_unwrap pbsrun.gm_mpd
```

2.8.5.4  MPICH-MX mpirun (mpirun.mpd) with MPD: pbsrun.mx_mpd

2.8.5.4.i  Syntax

```
pbsrun.mx_mpd <options> <executable> <arg1> <arg2> ... <argn>
```

The wrapper is deprecated. The PBS wrapper script to MPICH-MX's mpirun (mpirun.ch_mx) with MPD process startup method is called pbsrun.mx_mpd.

If executed inside a PBS job, this allows PBS to track all MPICH-MX processes started by the MPD daemons so that PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard mpirun.ch_mx with MPD were used.

2.8.5.4.ii  Options Handling

If executed inside a PBS job script, all mpirun.mx_mpd with MPD options given are passed on to the actual mpirun call with these exceptions:

- **m <file>**
  The file argument contents are ignored and replaced by the contents of $PBS_NODEFILE.

- **np**
  If not specified, the number of entries found in $PBS_NODEFILE is used.

- **pg**
  The use of the -pg option, for having multiple executables on multiple hosts, is allowed but it is up to the user to make sure only PBS hosts are specified in the process group file; MPI processes spawned are not guaranteed to be under the control of PBS.
2.8.5.4.iii  Startup/Shutdown

The script starts MPD daemons on each of the unique hosts listed in $PBS_NODEFILE, using either \texttt{rsh} or \texttt{ssh}, based on the value of the environment variable \texttt{RSHCOMMAND}. The default is \texttt{rsh}.

The script also takes care of shutting down the MPD daemons at the end of a run.

2.8.5.4.iv  Wrap/Unwrap

To wrap MPICH-MX's \texttt{mpirun} script with MPD:

\begin{verbatim}
  # pbsrun_wrap [MPICH-MX_BIN_PATH]/mpirun.mpd pbsrun.mx_mpd
\end{verbatim}

To unwrap MPICH-MX's \texttt{mpirun} script with MPD:

\begin{verbatim}
  # pbsrun_unwrap pbsrun.mx_mpd
\end{verbatim}

2.8.5.5  MPICH2 \texttt{mpirun}: \texttt{pbsrun.mpich2}

2.8.5.5.i  Syntax

\texttt{pbsrun.mpich2} [<global args>] [<local args>] <executable> [<args>] [: [<local args>] <executable> [<args>]]

- or -

\texttt{pbsrun.mpich2 -configfile <configfile>}

where \texttt{configfile} contains command line segments as lines:

\begin{verbatim}
  [local args] executable1 [args]
  [local args] executable2 [args]
  [local args] executable3 [args]
\end{verbatim}

The PBS wrapper script to MPICH2's \texttt{mpirun} is called \texttt{pbsrun.mpich2}.

If executed inside a PBS job, this allows PBS to track all MPICH2 processes so that PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard MPICH2's \texttt{mpirun} were used.

2.8.5.5.ii  Options Handling

If executed inside a PBS job script, all MPICH2's \texttt{mpirun} options given are passed on to the actual \texttt{mpirun} call with these exceptions:

- \texttt{-host} and \texttt{-ghost}

  For specifying the execution host to run on. Not passed on to the actual \texttt{mpirun} call.

- \texttt{-machinefile <file>}

  The \texttt{file} argument contents are ignored and replaced by the contents of $PBS_NODEFILE.

MPICH2's \texttt{mpirun -localonly <num processes>}

For specifying number of processes to run locally. Not supported. The user is advised instead to use the equivalent arguments: \texttt{-np <num processes> -localonly}. The reason for this is that the \texttt{pbsrun} wrapper script cannot handle a variable number of arguments to an option (e.g. \texttt{"-localonly"} has one argument and \texttt{"-localonly <num processes>"} has two arguments).

- \texttt{-np}

  If the user does not specify the \texttt{-np} option, no default value is provided by the PBS wrapper scripts. It is up to the local \texttt{mpirun} to decide what the reasonable default value should be, which is usually 1.
2.8.5.5.iii  **Startup/Shutdown**

The script takes care of ensuring that the MPD daemons on each of the hosts listed in $PBS_NODEFILE are started. It also takes care of ensuring that the MPD daemons have been shut down at the end of MPI job execution.

2.8.5.5.iv  **Wrap/Unwrap**

To wrap MPICH2’s mpirun script:

```
# pbsrun_wrap [<MPICH2 BIN PATH>/]mpirun pbsrun.mpich2
```

To unwrap MPICH2’s mpirun script:

```
# pbsrun_unwrap pbsrun.mpich2
```

In the case where MPICH2 uses mpirun.py, run pbsrun_wrap on mpirun.py itself.

2.8.5.6  **Intel MPI mpirun: pbsrun.intelmpi**

Wrapping Intel MPI, and support for mpdboot, are deprecated.

2.8.5.6.i  **Syntax**

```
pbsrun.intelmpi [<mpdboot options>] [<mpiexec options>] <executable> [<prog args>] [: [<mpiexec options>] <executable> [<prog args>]]
- or -
pbsrun.intelmpi [<mpdboot options>] -f <configfile>
```

where **mpdboot options** are any options to pass to the mpdboot program, which is automatically called by Intel MPI’s mpirun to start MPDs, and **configfile** contains command line segments as lines.

The PBS wrapper script to Intel MPI’s mpirun is called pbsrun.intelmpi.

If executed inside a PBS job, this allows PBS to track all Intel MPI processes so that PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard Intel MPI’s mpirun were used.

2.8.5.6.ii  **Options Handling**

If executed inside a PBS job script, all of the options to the PBS interface to Intel MPI’s mpirun are passed to the actual mpirun call with these exceptions:

- **-host** and **-ghost**
  
  For specifying the execution host to run on. Not passed on to the actual mpirun call.

- **-machinefile <file>**
  
  The file argument contents are ignored and replaced by the contents of $PBS_NODEFILE.

**mpdboot options --totalnum=* and --file=***

Ignored and replaced by the number of unique entries in $PBS_NODEFILE and name of $PBS_NODEFILE respectively.

**arguments to mpdboot options --file=* and -f <mpd_hosts_file>**

Replaced by $PBS_NODEFILE.

- **-s**

  If pbsrun.intelmpi is called inside a PBS job, Intel MPI’s mpirun -s argument to mpdboot is not supported as this closely matches the mpirun option -s <spec>. The user can simply run a separate mpdboot -s before calling mpirun. A warning message is issued by pbsrun.intelmpi upon encountering a -s option telling users of the supported form.
-np
If the user does not specify the -np option, no default value is provided by the PBS wrap scripts. It is up to the local mpirun to decide what the reasonable default value should be, which is usually 1.

2.8.5.6.iii  Startup/Shutdown

Intel MPI’s mpirun itself takes care of starting/stopping the MPD daemons. pbsrun.intelmpi always passes the arguments -totalnum=<number of mpds to start> and -file=<mpd_hosts_file> to the actual mpirun, taking its input from unique entries in $PBS_NODEFILE.

2.8.5.6.iv  Wrap/Unwrap

To wrap Intel MPI’s mpirun script:

# pbsrun_wrap [INTEL_MPI_BIN_PATH]/mpirun pbsrun.intelmpi

To unwrap Intel MPI’s mpirun script:

# pbsrun_unwrap pbsrun.intelmpi

2.8.5.7  MVAPICH1 mpirun: pbsrun.mvapich1

2.8.5.7.i  Syntax

pbsrun.mvapich1 <mpirun options> <executable> <options>

The wrapper is deprecated. The PBS wrapper script to MVAPICH1’s mpirun is called pbsrun.mvapich1. Only one executable can be specified. MVAPICH1 allows the use of InfiniBand.

If executed inside a PBS job, this allows PBS to be aware of all MVAPICH1 ranks and to track their resources, so that PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard mpirun were used.

2.8.5.7.ii  Options Handling

If executed inside a PBS job script, all mpirun options given are passed on to the actual mpirun call with these exceptions:

-map <list>
The map option is ignored.

-exclude <list>
The exclude option is ignored.

-machinefile <file>
The machinefile option is ignored.

-np
If not specified, the number of entries found in $PBS_NODEFILE is used.

2.8.5.7.iii  Wrap/Unwrap

To wrap MVAPICH1’s mpirun script:

# pbsrun_wrap <path-to-actual-mpirun> pbsrun.mvapich1

To unwrap MVAPICH1’s mpirun script:

# pbsrun_unwrap pbsrun.mvapich1
2.8.5.8 MVAPICH2 mpiexec: pbsrun.mvapich2

2.8.5.8.i Syntax

```
pbsrun.mvapich2 <mpiexec args> <executable> <executable’s args> [: <mpiexec args> <executable> <executable’s args>]  
```

The PBS wrapper script to MVAPICH2's mpiexec is called pbsrun.mvapich2.

Multiple executables can be specified using the colon notation. MVAPICH2 allows the use of InfiniBand.

If executed inside a PBS job, this allows PBS to be aware of all MVAPICH2 ranks and to track their resources, so that
PBS can perform accounting and have complete job control.

If executed outside of a PBS job, it behaves exactly as if standard mpiexec were used.

2.8.5.8.ii Options Handling

If executed inside a PBS job script, all mpiexec options given are passed on to the actual mpiexec call with these exceptions:
- `--host <hostname>`
  
  The `hostname` argument contents are ignored.
- `--machinefile <file>`
  
  The `file` argument contents are ignored and replaced by the contents of the `$PBS_NODEFILE`.

2.8.5.8.iii Wrap/Unwrap

To wrap MVAPICH2’s mpiexec script:

```
# pbsrun_wrap <path-to-actual-mpiexec> pbsrun.mvapich2  
```

To unwrap MVAPICH2’s mpiexec script:

```
# pbsrun_unwrap pbsrun.mvapich2  
```

2.8.6 Requirements

The mpirun being wrapped must be installed and working on all the vnodes in the PBS cluster.

2.8.7 Errors

If pbsrun encounters any option not found in `options_to_retain`, `options_to_ignore`, and `options_to_transform`, it is flagged as an error.

2.8.8 See Also

The PBS Professional Administrator’s Guide, "pbs_attach" on page 55, "pbsrun_wrap" on page 51, "pbsrun_unwrap" on page 50
Chapter 2  PBS Commands

2.9 pbsrun_unwrap

Unwraps mpirun, reversing pbsrun_wrap

2.9.1 Synopsis

pbsrun_unwrap pbsrun.<mpirun version/flavor>
pbsrun_unwrap --version

2.9.2 Description

The pbsrun_unwrap script is used to reverse the actions of the pbsrun_wrap script.
Use pbsrun_wrap to wrap mpirun.
Using pbsrun_unwrap for Intel MPI is deprecated as of 13.0.

2.9.2.1 Syntax

pbsrun_unwrap pbsrun.<mpirun version/flavor>

For example, running the following:

    pbsrun_unwrap pbsrun.ch_gm

causes the following actions:

1. Checks for a link in $PBS_EXEC/lib/MPI/pbsrun.ch_gm.link; If one exists, get the pathname it points to, for example:
   /opt/mpich-gm/bin/mpirun.ch_gm.actual
2. rm $PBS_EXEC/lib/MPI/pbsrun.mpirun.ch_gm.link
3. rm /opt/mpich-gm/bin/mpirun.ch_gm
4. rm $PBS_EXEC/bin/pbsrun.ch_gm
5. mv /opt/mpich-gm/bin/mpirun.ch_gm.actual /opt/mpich-gm/bin/mpirun.ch_gm

2.9.3 Options

--version

    The pbsrun_unwrap command returns its PBS version information and exits. This option can only be used alone.

2.9.4 See Also

The PBS Professional Administrator’s Guide, "pbs_attach" on page 55, "pbsrun" on page 40, "pbsrun_wrap" on page 51
2.10 pbsrun_wrap

General-purpose script for wrapping mpirun in pbsrun

2.10.1 Synopsis

pbsrun_wrap [-s] <path to actual mpirun> pbsrun.<mpirun version/flavor>

pbsrun_wrap --version

2.10.2 Description

The pbsrun_wrap script is used to wrap any of several versions of mpirun in pbsrun. The pbsrun_wrap script creates a symbolic link with the same path and name as the mpirun being wrapped. This calls pbsrun, which uses pbs_attach to give MoM control of jobs. The result is transparent to the user; when mpirun is called from inside a PBS job, PBS can monitor and control the job, but when mpirun is called from outside of a PBS job, it behaves as it would normally. See “pbs_attach” on page 55 and “pbsrun” on page 40.

Use pbsrun_unwrap to reverse the process.

Using pbsrun_wrap for Intel MPI is deprecated as of 13.0.

2.10.2.1 Syntax

pbsrun_wrap [-s] <path to actual mpirun> pbsrun.<mpirun version/flavor>

Any mpirun version/flavor that can be wrapped has an initialization script ending in “.init”, found in $PBS_EXEC/lib/MPI:

$PBS_EXEC/lib/MPI/pbsrun.<mpirun version/flavor>.init

The pbsrun_wrap script instantiates the pbsrun wrapper script as pbsrun.<mpirun version/flavor> in the same directory where pbsrun is located, and sets up the link to actual mpirun call via the symbolic link:

$PBS_EXEC/lib/MPI/pbsrun.<mpirun version/flavor>.link

For example, running:

pbsrun_wrap /opt/mpich-gm/bin/mpirun.ch_gm pbsrun.ch_gm

causes the following actions:

1. Save original mpirun.ch_gm script:

   mv /opt/mpich-gm/bin/mpirun.ch_gm /opt/mpich-gm/bin/mpirun.ch_gm.actual

2. Instantiate pbsrun wrapper script as pbsrun.ch_gm:

   cp $PBS_EXEC/bin/pbsrun $PBS_EXEC/bin/pbsrun.ch_gm

3. Link “mpirun.ch_gm” to actually call “pbsrun.ch_gm”:

   ln -s $PBS_EXEC/bin/pbsrun.ch_gm /opt/mpich-gm/bin/mpirun.ch_gm

4. Create a link so that “pbsrun.ch_gm” calls “mpirun.ch_gm.actual”:

   ln -s /opt/mpich-gm/bin/mpirun.ch_gm.actual $PBS_EXEC/lib/MPI/pbsrun.ch_gm.link
2.10.3 Options

-\texttt{s}

Sets the “\texttt{strict\_pbs}” options in the various initialization scripts (e.g. \texttt{pbsrun.bgl.init}, \texttt{pbsrun.ch\_gm.init}, etc..) to 1 from the default 0. This means that the \texttt{mpirun} being wrapped by \texttt{pbsrun} will only be executed if inside a PBS environment. Otherwise, the user gets the error:

\texttt{Not running under PBS exiting since strict\_pbs is enabled; execute only in PBS}

--\texttt{version}

The \texttt{pbsrun\_wrap} command returns its PBS version information and exits. This option can only be used alone.

2.10.4 Requirements

The \texttt{mpirun} being wrapped must be installed and working on all the vnodes in the PBS complex.

2.10.5 See Also

The PBS Professional Administrator’s Guide, "\texttt{pbs\_attach} on page 55, "\texttt{pbsrun} on page 40, "\texttt{pbsrun\_unwrap} on page 50
2.11 pbs_account

For Windows. Manage PBS service account

2.11.1 Synopsis


2.11.2 Description

The `pbs_account` command is used to manage the PBS service account. It is used to create the account, set or validate the account password, add privileges to the account, and register or unregister the account with the SCM.

2.11.2.1 Permissions

This command can be run by administrators only.

2.11.2.2 Platforms

This command is available on Windows only.

2.11.3 Options

- `-a <account name>`
  Specifies service account name.
- `-c [<password>]`
  - If specified account does not exist, creates the account with the password.
  - If specified account exists, validates password against it.
  Gives necessary privileges to the specified account: `Create Token Object, Replace Process Level Token, Log on as a Service, and Act as Part of the Operating System`
  If password is not specified, user is prompted for password.
- `--ci`
  Informational only. Prints actions taken by `pbs_account` while creating PBS service account when operations are performed.
- `--instid <instance ID>`
  Specifies the instance ID when registering or unregistering multiple instances of a service. Example:
  ```
  pbs_account --reg "C:\Program Files (x86)\PBS Pro_2\exec\sbin\pbs_mom" --instid 2 -a <username> -p <password>
  pbs_account --unreg "C:\Program Files (x86)\PBS Pro_2\exec\sbin\pbs_mom" --instid 2
  ```
- `-o <output path>`
  Prints stdout and stderr messages in specified output path.
- `-p [<password>]`
  Updates the PBS service account password. If no password is specified, the user is prompted for a password.
--reg <path to service>
Registers the PBS service with the SCM, instructing it to run the services under the PBS service account. path to service must be in double quotes.

-s
Adds necessary privileges to the PBS service account. Grants the "Create Token Object", "Replace Process Level Token", "Log On as a Service", and "Act as Part of the Operating System" privileges to PBS service account.

--unreg <path to service>
Unregisters the PBS service with the SCM. path to service must be in double quotes.

(no options)
Prints name of PBS service account, if it exists. Exit value is 0.

2.11.4 Examples

Example 2-1: To create the PBS service account:

```bash
pbs_account -c -s -p <password>
```

Example 2-2: To change the PBS service account:

```bash
pbs_account --reg <service path> -a <PBS service account name>
```

Example 2-3: To register the server, scheduler, MoM, comm, and rshd services:

```bash
pbs_account --reg "\Program Files\PBS\exec\sbin\pbs_server.exe" -p <password>
pbs_account --reg "\Program Files\PBS\exec\sbin\pbs_mom.exe" -p <password>
pbs_account --reg "\Program Files\PBS\exec\sbin\pbs_sched.exe" -p <password>
pbs_account --reg "\Program Files\PBS\exec\sbin\pbs_comm.exe" -p <password>
pbs_account --reg "\Program Files\PBS\exec\sbin\pbs_rshd.exe" -p <password>
```

2.11.5 Exit Value

Zero
Upon success
2.12 **pbs_attach**

Attaches a session ID to a PBS job

### 2.12.1 Synopsis

**Linux**

```bash
pbs_attach [-j <job ID>] [-m <port number>] -p <PID>
pbs_attach [-j <job ID>] [-m <port number>] [-P] [-s] <cmd> [<arg> ...]
pbs_attach --version
```

**Windows**

```bash
pbs_attach [-c <path to script>] [-j <job ID>] [-m <port number>] -p <PID>
pbs_attach [-c <path to script>] [-j <job ID>] [-m <port number>] [-P] [-s] <cmd> [<arg> ...]
pbs_attach --version
```

### 2.12.2 Description

The `pbs_attach` command associates the processes in a session with a PBS job by attaching the session ID to the job. This allows PBS MoM to monitor and control those processes.

MoM uses process IDs to determine session IDs, which are put into MoM’s task list (attached to the job.) All process IDs in a session are then associated with the job.

When a command `cmd` is given as an operand, the `pbs_attach` process becomes the parent process of `cmd`, and the session ID of `pbs_attach` is attached to the job.

### 2.12.3 Options to `pbs_attach`

- `-c <path to script>`
  - Windows only. Specified command is invoked using a new command shell. In order to spawn and attach built-in DOS commands such as `set` or `echo`, it is necessary to open the task using a `cmd` shell. The new command shell, `cmd.exe`, is attached as a task to the PBS job. The `pbs_attach` command spawns a program using a new command shell when attaching a batch script, or when invoked with the `-c` option.

- `-j <job ID>`
  - The job ID to which the session ID is to be attached. If `job ID` is not specified, a best effort is made to determine the job to which to attach the session.

- `-m <port number>`
  - The port at which to contact MoM. Default: value of `$PBS_MANAGER_SERVICE_PORT` from `pbs.conf`.

- `-p <PID>`
  - Process ID whose session ID is to be attached to the job. Default: process ID of `pbs_attach`. Cannot be used with the `-p` or `-s` options or the `cmd` operand.

- `-P`
  - Attach sessions of both `pbs_attach` and the parent of `pbs_attach` to job. When used with `-s` option, the sessions of the new `fork()`ed `pbs_attach` and its parent, which is `pbs_attach`, are attached to the job. Cannot be used with the `-p` or `-s` options or the `cmd` operand.
PBS Commands

Chapter 2

2.12.3 PBS Commands

-s
Starts a new session and attaches it to the job; pbs_attach calls fork(), then the child pbs_attach first calls setsid() and then calls tm_attach to attach the new session to the job. The session ID of the new pbs_attach is attached to the job.

--version
The pbs_attach command returns its PBS version information and exits. This option can only be used alone.

2.12.4 Operands

cmd
Name of command whose process ID is to be associated with the job.

2.12.5 Exit Status

0
Success

1
Any error following successful command line processing. A message is printed to standard error.
If cmd is specified, pbs_attach waits for cmd to exit, then exits with the exit value of cmd.
If cmd is not specified, pbsAttach exits after attaching the session ID(s) to the job.

2.12.6 See Also

2.13 **pbs_comm**

Starts the PBS communication daemon

### 2.13.1 Synopsis

**Linux:**

```bash
pbs_comm [-N] [-r <other routers>] [-t <number of threads>]
pbs_comm --version
```

**Windows:**

```bash
pbs_comm.exe [-R|-U|-N] [-r <other routers>] [-t <number of threads>]
pbs_comm --version
```

### 2.13.2 Description

The PBS communication daemon, `pbs_comm`, handles communication between daemons, except for scheduler-server and server-server communication, which uses TCP. The server, scheduler(s), and MoMs are connected by one or more `pbs_comm` daemons.

See “Communication” on page 47 in the PBS Professional Installation & Upgrade Guide.

### 2.13.3 Options to pbs_comm

- **-N**
  
  Runs the communication daemon in standalone mode.

- **-r <other routers>**
  
  List of other `pbs_comm` daemons to which this `pbs_comm` must connect. This is equivalent to the `pbs.conf` variable `PBS_COMM_ROUTERS`. The command line overrides the variable. Format:
  ```plaintext
  <hostname>[:<port number>],<hostname>[:<port number>]
  ```

- **-R**
  
  Registers the `pbs_comm` process. Available under Windows only.

- **-t <number of threads>**
  
  Number of threads the `pbs_comm` daemon uses. This is equivalent to the `pbs.conf` variable `PBS_COMM_THREADS`. The command line overrides the variable. Format:
  ```plaintext
  Integer
  ```

- **-U**
  
  Unregisters the `pbs_comm` process. Available under Windows only.

### 2.13.4 Configuration Parameters

**PBS_LEAF_ROUTERS**

Parameter in `/etc/pbs.conf`. Tells an endpoint where to find its communication daemon.

You can tell each endpoint which communication daemon it should talk to. Specifying the port is optional.

Format: `PBS_LEAF_ROUTERS=<hostname>[:<port number>],<hostname>[:<port number>]`
PBS Commands

**PBS_COMM_ROUTERS**
Parameter in `/etc/pbs.conf`. Tells `pbs_comm` where to find its fellow communication daemons.
When you add a communication daemon, you must tell it about the other `pbs_comm`s in the complex. When you inform communication daemons about each other, you only tell one of each pair about the other. Do not tell both about each other. We recommend that an easy way to do this is to tell each new `pbs_comm` about each existing `pbs_comm`, and leave it at that.

Format: `PBS_COMM_ROUTERS=<hostname>[:<port number>],<hostname>[:<port number>]`

**PBS_COMM_THREADS**
Parameter in `/etc/pbs.conf`. Tells `pbs_comm` how many threads to start.
By default, each `pbs_comm` process starts four threads. You can configure the number of threads that each `pbs_comm` uses. Usually, you want no more threads than the number of processors on the host.
Maximum allowed value: 100
Format: `Integer`
Example: `PBS_COMM_THREADS=8`

**PBS_COMM_LOG_EVENTS**
Parameter in `/etc/pbs.conf`. Tells `pbs_comm` which log mask to use.
By default, `pbs_comm` produces few log messages. You can choose more logging, usually for troubleshooting. See “Logging and Errors with TPP” on page 56 in the PBS Professional Installation & Upgrade Guide for logging details.
Format: `Integer`
Default: 511
Example: `PBS_COMM_LOG_EVENTS=<log level>`

**PBS_LEAF_NAME**
Parameter in `/etc/pbs.conf`. Tells endpoint what name to use for network. The value does not include a port, since that is usually set by the daemon.
By default, the name of the endpoint’s host is the hostname of the machine. You can set the name where an endpoint runs. This is useful when you have multiple networks configured, and you want PBS to use a particular network. TPP internally resolves the name to a set of IP addresses, so you do not affect how `pbs_comm` works.
Format: `String`
Example: `PBS_LEAF_NAME=host1`

**PBS_START_COMM**
Parameter in `/etc/pbs.conf`. Tells PBS init script whether to start a `pbs_comm` on this host if one is installed. When set to 1, `pbs_comm` is started.
Just as with the other PBS daemons, you can specify whether each host should start `pbs_comm`.
Format: `Boolean`
Default: 0
Example: `PBS_START_COMM=1`
2.13.5 Communication Daemon Logfiles

The pbs_comm daemon creates its log files under $PBS_HOME/comm_logs. This directory is automatically created by the PBS installer.

In a failover configuration, this directory is in the shared PBS_HOME, and is used by the pbs_comm daemons running on both the primary and secondary servers. This directory must never be shared across multiple pbs_comm daemons in any other case.

The log filename format is yyyyymmdd (the same as for other PBS daemons).

The log record format is the same as used by other pbs daemons, with the addition of the thread number and the daemon name in the log record. The log record format is as follows:

$date and time$;<event code>;<daemon name>(<thread number>);<object type>;<object name>;<message>

Example:

03/25/2014 15:13:39;0d86;host1.example.com;TPP;host1.example.com(Thread 2);Connection from leaf 192.168.184.156:19331, tfd=81 down

2.13.6 Signal Handling by Communication Daemon

The pbs_comm daemon handles the following signals:

- **HUP**
  
  Re-reads the value of $PBS_COMM_LOG_EVENTS from pbs.conf.

- **TERM**
  
  The pbs_comm daemon exits.
2.14  pbs_dataservice

Start, stop, or check the status of PBS data service

2.14.1  Synopsis

pbs_dataservice [start | stop | status]

2.14.2  Description

The pbs_dataservice command starts, stops or gets the status of the PBS data service.

2.14.2.1  Permission

On Linux, root privilege is required to use this command. On Windows, Admin privilege is required.

2.14.3  Arguments

start
   Starts the PBS data service.

stop
   Stops the PBS data service.
   Can be used only when the PBS server is not running.

status
   Displays the status of the PBS data service, as follows:
   •  Data service running
      PBS Data Service running
   •  Data service not running
      PBS Data Service not running

2.14.4  Exit Status

Zero
   Success
Non-zero
   Failure
## 2.15 pbs_ds_password

Sets or changes data service user account or its password

### 2.15.1 Synopsis

```
pbs_ds_password [-C <username>] [-r]
```

### 2.15.2 Description

You can use this command to change the user account or account password for the data service.

#### 2.15.2.1 Passwords

Blank passwords are not allowed.

If you type in a password, make sure it does not contain restricted characters. The `pbs_ds_password` command generates passwords containing the following characters:

```
0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^&*()_+
```

When creating a password manually, do not use `\` (backslash) or `‘` (backquote). This can prevent certain commands such as `pbs_server`, `pbs_ds_password`, and `printjob` from functioning properly, as they rely on connecting to the database. The format is also described in "PBS Password" on page 367.

#### 2.15.2.2 Permissions

On Linux, root privilege is required to use this command. On Windows, Admin privilege is required.

#### 2.15.2.3 Restrictions

Do not run this command if failover is configured. It is important not to inadvertently start two separate instances of the data service on two machines, thus potentially corrupting the database.

### 2.15.3 Options to pbs_ds_password

- `-C <username>`
  Changes user account for data service to specified account. Specified user account must already exist.
  On Linux-based systems, the specified user account must not be root.
  On Windows, the specified user account must match the PBS service account (which can be any user account.)
  This option cannot be used while the data service is running.
  Can be used with the `-r` option to automatically generate a password for the new account.

- `-r`
  Generates a random password. The data service is updated with the new password.
  Can be used with the `-C` option.

(no options)
  Asks the user to enter a new password twice. Entries must match. Updates data service with new password.
2.15.4 Exit Status

Zero
  Success
Non-zero
  Failure
2.16  pbs_hostn

Reports hostname and network address(es)

2.16.1  Synopsis

    pbs_hostn [-v] <hostname>
    pbs_hostn --version

2.16.2  Description

The pbs_hostn command takes a hostname, and reports the results of both the gethostbyname(3) and gethostbyaddr(3) system calls. Both forward and reverse lookup of hostname and network addresses need to succeed in order for PBS to authenticate a host.

Running this command can assist in troubleshooting problems related to incorrect or non-standard network configuration, especially within clusters.

2.16.3  Options

-**v**

    Turns on verbose mode.

--version

    The pbs_hostn command returns its PBS version information and exits. This option can only be used alone.

2.16.4  Operands

    hostname

    The pbs_hostn command accepts a hostname operand either in short name form, or in fully qualified domain name (FQDN) form.

2.16.5  Standard Error

The pbs_hostn command writes a diagnostic message to standard error for each error occurrence.

2.16.6  Exit Status

    Zero

    Upon successful processing of all the operands presented to the pbs_hostn command.

    Greater than zero

    If the pbs_hostn command fails to process any operand.
Chapter 2  

**PBS Commands**

2.17  **pbs_idled**

Runs PBS daemon that monitors the console and informs `pbs_mom` of idle time

2.17.1  **Linux Synopsis**

```
pbs_idled [-D <display>] [-r <reconnect delay>] [-w <wait time>]
```

```
pbs_idled --version
```

2.17.2  **Windows Synopsis**

```
pbs_idled [start | stop]
```

```
pbs_idled --version
```

2.17.3  **Linux Description**

On Linux, the `pbs_idled` program monitors an X windows display and communicates the idle time of the display back to PBS. If the mouse is moved or a key is touched, PBS is informed that the vnode is busy.

You should run this program from the system-wide Xsession file, in the background before the window manager is run. If this program is run outside of the Xsession, it needs to be able to make a connection to the X display. See the `xhost` or `xauth` man pages for a description of X security.

2.17.4  **Windows Description**

On Windows, `pbs_idled` reads its polling interval from a file called `idle_poll_time` which is created by MoM. The service monitors keyboard, mouse, and console activity, and updates a file called `idle_touch` when it finds user activity. The `idle_touch` file is created by MoM.

2.17.5  **Linux Options to pbs_idled**

- `-D <display>`
  The display to connect to and monitor
- `-r <reconnect delay>`
  Time to wait before we try to reconnect to the X display if the previous attempt was unsuccessful
- `-w <wait time>`
  Interval between times when the daemon checks for events or pointer movement
- `--version`
  The `pbs_idled` command returns its PBS version information and exits. This option can only be used alone.

2.17.6  **Windows Options to pbs_idled**

- `start`
  Starts the `pbs_idled` process.
stop
  Stops the pbs_idled process.
--version
  The pbs_idled process returns its PBS version information and exits. This option can only be used alone.

2.17.7  See Also

The PBS Professional Administrator's Guide
xhost(1), xauth(1)
2.18  

**pbs_iff**

Tests authentication with the server

### 2.18.1  Usage

```
pbs_iff [-t] <server host> <server port>
pbs_iff --version
```

### 2.18.2  Description

Called from the `pbs_connect()` IFL API to authenticate a connection with the PBS server. Designed to be called internally by PBS commands and components, to be used by our IFL layer to talk to the server.

If `pbs_iff` cannot authenticate, it returns an error message.

#### 2.18.2.1  Required Privilege

Can be run by any user.

It's a setuid root binary so it runs as the user who requests a connection to a server but it becomes root so that it can grab a privileged port.

### 2.18.3  Options to pbs_iff

- `-t`
  
  Test mode; means test whether `pbs_iff` can authenticate with the server

- `--version`
  
  Reports version and exits; can only be used alone

### 2.18.4  Arguments to pbs_iff

- `daemon host`
  
  Host where server is running

- `daemon port`
  
  Port on which server is listening; default is 15001

### 2.18.5  Exit Status

- **Zero**
  
  If `pbs_iff` is able to contact the server at the specified port

- **Non-zero**
  
  If `pbs_iff` is unable to contact the server at the specified port
2.19 pbs_interactive

Windows. Register, unregister, or get the version of PBS_INTERACTIVE service

2.19.1 Synopsis

   pbs_interactive [R | U]
   pbs_interactive --version

2.19.2 Description

The pbs_interactive command registers, unregisters, or gets the version of the Windows PBS_INTERACTIVE service. The service must be registered manually; the installer does not register it.

On Windows, the PBS_INTERACTIVE service itself monitors logging in and out by users, starts a pbs_idled process for each user logging in, and stops the pbs_idled process of each user logging out.

2.19.2.1 Required Privilege

Admin privilege is required to use this command.

2.19.3 Arguments

R
   Registers the PBS_INTERACTIVE service.

U
   Unregisters the PBS_INTERACTIVE service.

--version
   The pbs_interactive command returns its PBS version information and exits. This option can only be used alone.
2.20  **pbs_lamboot**

PBS front end to LAM’s `lamboot` program

### 2.20.1 Synopsis

```bash
pbs_lamboot
pbs_lamboot --version
```

### 2.20.2 Description

The PBS command `pbs_lamboot` replaces the standard `lamboot` command in a PBS LAM MPI job, for starting LAM software on each of the PBS execution hosts running Linux 2.4 or higher.

Usage is the same as for LAM’s `lamboot`. All arguments except for `bhost` are passed directly to `lamboot`. PBS will issue a warning saying that the `bhost` argument is ignored by PBS since input is taken automatically from `PBS_NODEFILE`. The `pbs_lamboot` program will not redundantly consult the `PBS_NODEFILE` if it has been instructed to boot the vnodes using the `tm` module. This instruction happens when an argument is passed to `pbs_lamboot` containing “-ssi boot tm” or when the `LAM_MPI_SSI_boot` environment variable exists with the value `tm`.

### 2.20.3 Options

- `--version`

  The `pbs_lamboot` command returns its PBS version information and exits. This option can only be used alone.

### 2.20.4 Operands

The operands for `pbs_lamboot` are the same as for `lamboot`.

### 2.20.5 Environment Variables and Path

The `PATH` on remote machines must contain `PBS_EXEC/bin`.

### 2.20.6 See Also

2.21 **pbs_migrate_users**

Transfers per-user or per-server passwords between PBS servers during a migration upgrade

2.21.1 **Synopsis**

```
pbs_migrate_users <old server> <new server>
pbs_migrate_users --version
```

2.21.2 **Description**

The `pbs_migrate_users` command is used to transfer the per-user or per-server password of a PBS user from one server to another during a migration upgrade. Users’ passwords on the old server are not deleted.

Available on Windows and supported Linux x86 and x86_64 platforms only.

2.21.3 **Options**

```
--version
```

The `pbs_migrate_users` command returns its PBS version information and exits. This option can only be used alone.

2.21.4 **Operands**

The format of *old server* and *new server* is

```
<hostname>[::<port number>]
```

2.21.5 **Exit Status**

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
</tr>
<tr>
<td>-1</td>
<td>Writing out passwords to files failed</td>
</tr>
<tr>
<td>-2</td>
<td>Communication failure between <em>old server</em> and <em>new server</em></td>
</tr>
<tr>
<td>-3</td>
<td><code>single_signon_password_enable</code> not set in either <em>old server</em> or <em>new server</em></td>
</tr>
<tr>
<td>-4</td>
<td>User running <code>pbs_migrate_users</code> not authorized to migrate users</td>
</tr>
</tbody>
</table>

2.21.6 **See Also**

"`pbs_password`" on page 82
2.22 pbs_migration_assistant.bat

For Windows. Tells PBS which server configuration settings to use.

2.22.1 Synopsis

`pbs_migration_assistant.bat /b`
`pbs_migration_assistant.bat /r`

2.22.2 Description

On Windows, you use `C:\Program Files x86\PBS\exec\bin\pbs_migration_assistant.bat` to set server configuration variables appropriately.

2.22.3 Options

`/b`
Tells PBS to use the old server configuration settings.

`/r`
Tells PBS to use the new server configuration settings.
2.23 **pbs_mkdirs**

For Windows. Create, or fix the permissions of, the directories and files used by PBS

### 2.23.1 Synopsis

```bash
pbs_mkdirs

pbs_mkdirs [ mom | sched | server ]
```

### 2.23.2 Description

Runs on Windows only. If the directories and files used by PBS exist, the `pbs_mkdirs` command fixes their permissions. If the directories and/or files do not exist, the `pbs_mkdirs` command creates them, with the correct permissions. The `pbs_mkdirs` command always examines the following directories and files:

- `pbs.conf`
- `PBS_EXEC`
- `PBS_HOME/spool`
- `PBS_HOME/undelivered`
- `PBS_HOME/pbs_environment`

#### 2.23.2.1 Required Privilege

You must have Administrator privilege to run this command.

### 2.23.3 Options

**mom**

The `pbs_mkdirs` command examines the following additional items:

- `PBS_HOME/mom_priv`
- `PBS_HOME/mom_logs`

**sched**

The `pbs_mkdirs` command examines the following additional items:

- `PBS_HOME/sched_priv`
- `PBS_HOME/sched_log`

**server**

The `pbs_mkdirs` command examines the following additional items:

- `PBS_HOME/server_priv`
- `PBS_HOME/server_logs`

*(no options)*

The `pbs_mkdirs` command examines all of the files and directories specified for each of the `mom`, `server`, and `sched` options.

### 2.23.4 See Also

The PBS Professional Administrator’s Guide, "[pbs_probe](#) on page 84"
2.24  \textbf{pbs\_mom}

Runs the PBS job monitoring and execution daemon

\textbf{2.24.1  Synopsis}

\texttt{pbs\_mom \[-a \langle\text{alarm timeout}\rangle\] \[-C\langle\text{checkpoint directory}\rangle\] \[-c\langle\text{config file}\rangle\] \[-d\langle\text{MoM home directory}\rangle\] \[-L\langle\text{logfile}\rangle\] \[-M\langle\text{MoM port}\rangle\] \[-N\] \[-n\langle\text{nice value}\rangle\] \[-p|-r\] \[-R\langle\text{inter-MoM communication port}\rangle\] \[-S\langle\text{server port}\rangle\] \[-s\langle\text{script options}\rangle\]}

\texttt{pbs\_mom --version}

\textbf{2.24.2  Description}

The \texttt{pbs\_mom} command starts the PBS job monitoring and execution daemon, called \textit{MoM}.

The standard MoM starts jobs on the execution host, monitors and reports resource usage, enforces resource usage limits, and notifies the server when the job is finished. The MoM also runs any prologue scripts before the job runs, and runs any epilogue scripts after the job runs.

The MoM performs any communication with job tasks and with other MoMs. The MoM on the first vnode on which a job is running manages communication with the MoMs on the remaining vnodes on which the job runs.

The MoM manages one or more vnodes. PBS may treat a host as a set of virtual nodes, in which case one MoM manages all of the host’s vnodes. See the PBS Professional Administrator’s Guide.

\textbf{2.24.2.1  Logging}

The MoM’s log file is in \texttt{PBS\_HOME/mom\_logs}. The MoM writes an error message in its log file when it encounters any error. If it cannot write to its log file, it writes to standard error. The MoM writes events to its log file. The MoM writes its PBS version and build information to the logfile whenever it starts up or the logfile is rolled to a new file.

\textbf{2.24.2.2  Required Permission}

The executable for \texttt{pbs\_mom} is in \texttt{PBS\_EXEC/sbin}, and can be run only by root on Linux, and Admin on Windows.

\textbf{2.24.2.3  Cpusets}

A cpusetted machine can have a \texttt{boot cpuset} defined by the administrator. A boot cpuset contains one or more CPUs and memory boards and is used to restrict the default placement of system processes, including login. If defined, the boot cpuset contains CPU 0.

Run parallel jobs exclusively within a cpuset for repeatability of performance. HPE SGI states, “Using cpusets on an HPE SGI system improves cache locality and memory access times and can substantially improve an application’s performance and runtime repeatability.”

The \texttt{CPUSET\_CPU\_EXCLUSIVE} flag prevents CPU 0 from being used by the MoM in the creation of job cpusets. This flag is set by default, so this is the default behavior.

To find out which cpuset is assigned to a running job, use \texttt{qstat -f} to see the \texttt{cpuset} field in the job’s \textit{altid} attribute.

\textbf{2.24.2.3.i  HPE SGI Systems Running Supported Versions of HPE MPI}

The cpusets created for jobs are marked cpu-exclusive.

MoM does not use any CPU which was in use at startup.
A PBS job can run across multiple machines that run supported versions of HPE MPI.
PBS can run using HPE SGI’s MPI (MPT) over InfiniBand. See the PBS Professional Administrator’s Guide.

### 2.24.2.4 Effect on Jobs of Starting MoM

When MoM is started or restarted, her default behavior is to leave any running processes running, but to tell the PBS server to requeue the jobs she manages. MoM tracks the process ID of jobs across restarts.

In order to have all jobs killed and requeued, use the `-r` option when starting or restarting MoM.

In order to leave any running processes running, and not to requeue any jobs, use the `-p` option when starting or restarting MoM.

### 2.24.3 Options to `pbs_mom`

- `-a <alarm timeout>`
  Number of seconds before alarm timeout. Whenever a resource request is processed, an alarm is set for the given amount of time. If the request has not completed before `alarm timeout`, the OS generates an alarm signal and sends it to MoM.
  Format: `Integer`
  Default: 10 seconds

- `-C <checkpoint directory>`
  Specifies the path to the directory where MoM creates job-specific subdirectories used to hold each job’s restart files. MoM passes this path to checkpoint and restart scripts. Overrides other checkpoint path specification methods. Any directory specified with the `-C` option must be owned, readable, writable, and executable by root only (`rwx`, `---`, or `0700`), to protect the security of the restart files. See the `-d` option to `pbs_mom` and “Specifying Checkpoint Path” on page 421 in the PBS Professional Administrator’s Guide.
  Format: `String`
  Default: `PBS_HOME/checkpoint`

- `-c <config file>`
  MoM will read this alternate default configuration file upon starting. If this is a relative file name it is relative to `PBS_HOME/mom_priv`. If the specified file cannot be opened, `pbs_mom` will abort. See the `-d` option.
  MoM’s normal operation, when the `-c` option is not given, is to attempt to open the default configuration file `PBS_HOME/mom_priv/config`. If this file is not present, `pbs_mom` will log the fact and continue.

- `-d <MoM home directory>`
  Specifies the path of the directory to be used in place of `PBS_HOME` by `pbs_mom`. The default directory is given by `$PBS_HOME`.
  Format: `String`

- `-L <logfile>`
  Specifies an absolute path and filename for the log file. The default is a file named for the current date in `PBS_HOME/mom_logs/`. See the `-d` option.
  Format: `String`.

- `-M <MoM port>`
  Specifies the port number on which MoM will listen for server requests and instructions. Overrides `PBS_MOM_SERVICE_PORT` setting in `pbs.conf` and environment variable.
  Format: `Integer port number`.
  Default: 15002.
-n <nice value>
   Specifies the priority for the pbs_mom daemon.
   Format: Integer.

-N
   Specifies that when starting, MoM should not detach from the current session.

-p
   Specifies that when starting, MoM should allow any running jobs to continue running, and not have them
   requeued. This option can be used for single-host jobs only; multi-host jobs cannot be preserved. Cannot be
   used with the -r option. MoM is not the parent of these jobs.

   HPE SGI systems running HPE MPI:
   The cpuset-enabled pbs_mom will, if given the -p flag, use the existing CPU and memory allocations for
   the /PBSPro cpusets. The default behavior is to remove these cpusets. Should this fail, MoM will exit, asking
   to be restarted with the -p flag.

-r
   Specifies that when starting, MoM should requeue any rerunnable jobs and kill any non-rerunnable jobs that she
   was tracking, and mark the jobs as terminated. Cannot be used with the -p option. MoM is not the parent of
   these jobs.
   It is not recommended to use the -r option after a reboot, because process IDs of new, legitimate tasks may
   match those MoM was previously tracking. If they match and MoM is started with the -r option, MoM will kill
   the new tasks.

-R <inter-MoM communication port>
   Specifies the port number on which MoM will listen for pings, resource information requests, communication
   from other MoMs, etc. Overrides PBS_MANAGER_SERVICE_PORT setting in pbs.conf and environment
   variable.
   Format: Integer port number
   Default: 15003

-S <server port>
   Specifies the port number on which pbs_mom initially contacts the server.
   Format: Integer port number
   Default: 15001

-s <script options>
   This option provides an interface that allows the administrator to add, delete, and display MoM’s configuration
   files. The script options are used this way:
   -s insert <scriptname> <inputfile>
      Reads inputfile and inserts its contents in a new site-defined pbs_mom configuration file with the filename
      scriptname. Example:
      
      pbs_mom -s insert <scriptname> <inputfile>

      If a site-defined configuration file with the name scriptname already exists, the operation fails, a diagnostic
      is presented, and pbs_mom exits with a nonzero status. Scripts whose names begin with the prefix “PBS”
      are reserved. An attempt to add a script whose name begins with “PBS” will fail and pbs_mom will print
      a diagnostic message and exit with a nonzero status.
   -s remove <scriptname>
      The configuration file named scriptname is removed if it exists. Example:
      
      pbs_mom -s remove <scriptname>
If the given name does not exist or if an attempt is made to remove a script with the reserved “PBS” prefix, the operation fails, a diagnostic is presented, and `pbs_mom` exits with a nonzero status.

```bash
-s show <scriptname>
```
Causes the contents of the named script to be printed to standard output. Example:

```bash
pbs_mom -s show <scriptname>
```
If `scriptname` does not exist, the operation fails, a diagnostic is presented, and `pbs_mom` exits with a nonzero status.

```bash
-s list
```
Causes `pbs_mom` to list the set of PBS-prefixed and site-defined configuration files in the order in which they are executed. Example:

```bash
pbs_mom -s list
```

**WINDOWS:**

Under Windows, the `-N` option must be used, so that `pbs_mom` will start up as a standalone program. For example:

```bash
pbs_mom -N -s insert <scriptname> <inputfile>
```
or

```bash
pbs_mom -N -s list
```

```bash
--version
```
The `pbs_mom` command returns its PBS version information and exits. This option can only be used alone.

### 2.24.4 Files and Directories

- `$PBS_HOME/mom_priv`  
  Default directory for default configuration files.

- `$PBS_HOME/mom_priv/config`  
  MoM’s default configuration file.

- `$PBS_HOME/mom_logs`  
  Default directory for log files written by MoM.

- `$PBS_HOME/mom_priv/prologue`  
  File containing administrative script to be run before job execution.

- `$PBS_HOME/mom_priv/epilogue`  
  File containing administrative script to be run after job execution.

### 2.24.5 Signal Handling

`pbs_mom` handles the following signals:

- **SIGHUP**
  The `pbs_mom` daemon rereads its configuration files, close and reopen the log file, and reinitialize resource structures.

- **SIGNALRM**
  MoM writes a log file entry. See the `-a <alarm timeout>` option.
Chapter 2  

PBS Commands

2.24.3  SIGINT

The pbs_mom daemon exits, leaving all running jobs still running. See the -p option.

2.24.3  SIGKILL

This signal is not caught. The pbs_mom daemon exits immediately.

2.24.3  SIGTERM, SIGXCPU, SIGXFSZ, SIGCPULIM, SIGSHUTDN

The pbs_mom daemon terminates all running children and exits.

2.24.3  SIGPIPE, SIGUSR1, SIGUSR2, SIGINFO

These are ignored.

All other signals have their default behavior installed.

2.24.6  Exit Status

Zero

Upon success

Greater than zero

• If the pbs_mom daemon fails to start
• If the -s insert option is used with an existing scriptname
• If the administrator attempts to add a script whose name begins with “PBS”
• If the administrator attempts to use the -s remove option on a non-existent configuration file, or on a configuration file whose name begins with “PBS”
• If the administrator attempts to use the -s show option on a non-existent script

2.24.7  See Also

The PBS Professional Administrator’s Guide
2.25 **pbs_mpihp**

Runs an MPI application in a PBS job with HP MPI

### 2.25.1 Synopsis

```bash
pbs_mpihp [-h <hostname>] [np <number>] [<other HP mpirun options>] <program> [<args>]
pbs_mpihp [<HP mpirun options>] -f <appfile> [-- [<extra args>]]
pbs_mpihp --version
```

### 2.25.2 Description

The PBS command `pbs_mpihp` replaces the standard `mpirun` command in a PBS HP MPI job, for executing programs. `pbs_mpihp` is a front end to the HP MPI version of `mpirun`.

When `pbs_mpihp` is invoked from a PBS job, it processes the command line arguments, then calls standard HP `mpirun` to actually start the MPI ranks. The ranks created are mapped onto CPUs on the vnodes allocated to the PBS job. The environment variable `MPI_REMSH` is set to `$PBS_EXEC/bin/pbs_tmrsh`. This causes the processes that are created to become part of the PBS job.

The path to standard HP `mpirun` is found by checking to see if a link exists with the name `PBS_EXEC/etc/pbs_mpihp`. If this link exists, it points to standard HP `mpirun`. If it does not exist, a call to `mpirun --version` is made to determine whether it is HP `mpirun`. If so, the call is made to `mpirun` without an absolute path. If HP `mpirun` cannot be found, an error is output, all temp files are cleaned up and the script exits with value 127.

If `pbs_mpihp` is invoked from outside a PBS job, it passes all of its arguments directly to standard HP `mpirun` without further processing.

#### 2.25.2.1 Configuration

When HP MPI is wrapped with `pbs_mpihp`, “rsh” is the default used to start the mpids. If you wish to use “ssh” or something else, be sure to set the following in `$PBS_HOME/pbs_environment`:

```bash
PBS_RSHCOMMAND=ssh
```

or put the following in the job script:

```bash
export PBS_RSHCOMMAND=<rsh_cmd>
```

#### 2.25.2.2 Usage

Usage is the same as for HP `mpirun`.

`pbs_mpihp <program>` allows one executable to be specified.

`pbs_mpihp -f <appfile>` uses an `appfile` to list multiple executables. The format is described in the HP `mpirun` man page. If this form is used from inside a PBS job, the file is read to determine what executables are to be run and how many processes are started for each.

Executing `pbs_mpihp` with the `-client` option is not supported under PBS.

### 2.25.3 Options to `pbs_mpihp`

All options except the following are passed directly to HP `mpirun` with no modification.
PBS Commands

Chapter 2

- client
  Not supported.
- f <appfile>
  The specified appfile is read by pbs_mpihp.
- h <hostname>
  Ignored by pbs_mpihp.
- l <username>
  Ignored by pbs_mpihp.
- np <number>
  Specifies the number of processes to run on the PBS vnodes.
  --version
  The pbs_mpihp command returns its PBS version information and exits. This option can only be used alone.

2.25.4 Exit Values

127
  If HP mpirun cannot be found

2.25.5 See Also

The PBS Professional Administrator’s Guide
mpirun(1)
2.26  **pbs_mpilam**

Runs MPI programs under PBS with LAM MPI

### 2.26.1 Synopsis

```
pbs_mpilam [mpilam options]
pbs_mpilam --version
```

### 2.26.2 Description

The PBS command `pbs_mpilam` replaces the standard `mpirun` command in a PBS LAM MPI job.

If used to run a single program, PBS tracks resource usage and controls all user processes spawned by the program. If used to run multiple programs as specified in an application file (no `<where>` argument and no `-np/-c` option), PBS does not manage the spawned user processes of each program.

If the `where` argument is not specified, `pbs_mpilam` will try to run the user's program on all available CPUs using the `C` keyword.

#### 2.26.2.1 Prerequisites

The `PATH` on remote machines must contain `PBS_EXEC/bin`.

#### 2.26.2.2 Usage

Usage is the same as for LAM `mpirun`. All options are passed directly to `mpirun`.

#### 2.26.3 Options to `pbs_mpilam`

- `<mpilam options>`
  - The `pbs_mpilam` command uses the same options as `mpirun`.

- `--version`
  - The `pbs_mpilam` command returns its PBS version information and exits. This option can only be used alone.

### 2.26.4 See Also

The PBS Professional Administrator’s Guide

`mpirun(1)`
2.27 pbs_mpirun

Deprecation: Runs MPI programs under PBS with MPICH

2.27.1 Synopsis

`pbs_mpirun [<mpirun options>]`

`pbs_mpirun --version`

2.27.2 Description

The PBS command `pbs_mpirun` replaces the standard `mpirun` command in a PBS MPICH job using P4.

On Windows, this command cannot be used to start job processes or track a job’s resource usage.

2.27.2.1 Prerequisite

The PATH on remote machines must contain `PBS_EXEC/bin`.

2.27.2.2 Usage

Usage is the same as for `mpirun`, except for the `-machinefile` option. All other options are passed directly to `mpirun`.

2.27.3 Options to `pbs_mpirun`

`<mpirun options>`

The options to `pbs_mpirun` are the same as for `mpirun`, except for the `-machinefile` option. This is generated by `pbs_mpirun`. The user should not attempt to specify `-machinefile`.

The value for `-machinefile` is a temporary file created from `PBS_NODEFILE` in the format:

```
hostname-1[:number of processors]
hostname-2[:number of processors]
hostname-n[:number of processors]
```

where if the number of processors is not specified, it is 1. An attempt by the user to specify the `-machinefile` option will result in a warning saying “Warning, -machinefile value replaced by PBS”.

The default value for the `-np` option is the number of entries in `PBS_NODEFILE`.

`--version`

The `pbs_mpirun` command returns its PBS version information and exits. This option can only be used alone.

2.27.4 Environment Variables

`pbs_mpirun` modifies `P4_RSHCOMMAND` and `PBS_RSHCOMMAND`. Users should not edit these.

`pbs_mpirun` copies the value of `P4_RSHCOMMAND` into `PBS_RSHCOMMAND`. 
2.27.5  See Also

The PBS Professional Administrator’s Guide, mpirun(1)
2.28  **pbs_password**  

Sets or updates password of a PBS user

2.28.1  **Synopsis**

```
pbs_password [-d] [-r] [-s <server name>] [<username>]
```

```
pbs_password --version
```

2.28.2  **Description**

The `pbs_password` command is used to set or update the password of a PBS user. The user does not have to have any jobs on the system. Available on Windows and supported Linux x86 and x86_64 platforms only. 

The `pbs_password` command has no effect on running jobs. Queued jobs use the new password. 

The `pbs_password` command does not change the user's login password on the current host, only the password that is cached in PBS. 

The `pbs_password` command encrypts the password obtained from the user before sending it to the PBS server.

2.28.2.1  **Required Privilege**

An unprivileged user can use this command to set or update their own password. 

Root or Admin can use this command to set or update the password of another user.

2.28.3  **Options to pbs_password**

(No options)

The user executing this command is prompted for a new password. This user’s password credential on the default PBS server is updated to the prompted password. Any user jobs previously held due to an invalid password are not released.

- **-d**

Deletes password. If `username` is not specified, deletes the password of the current user. If `username` is specified and the current user is root or Admin, deletes the password of the specified user.

- **-r**

Any user jobs previously held due to an invalid password are released.

- **-s <server name>**

The `server name` where the password is to be changed.

- **<username>**

The password credential of user `username` is updated to the prompted password. If `username` is not the current user, this action is only allowed if one of the following is true:

- The current user is root or Admin
- User `username` has given the current user explicit access via the `ruserok()` mechanism, i.e. the hostname of the machine from which the current user is logged in appears in the server `hosts.equiv` file, or the current user has an entry in `username`’s `HOMEDIR\.rhosts` file
The `pbs_password` command returns its PBS version information and exits. This option can only be used alone.

### 2.28.4 Exit Status

#### Table 2-4: Exit Status

<table>
<thead>
<tr>
<th>Exit Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
</tr>
<tr>
<td>-1</td>
<td><code>single_signon_password_enable</code> not set</td>
</tr>
<tr>
<td>-2</td>
<td>Password of user on server failed to be created/updated</td>
</tr>
<tr>
<td>-3</td>
<td>Failed to release jobs held due to bad password owned by user on server</td>
</tr>
<tr>
<td>-4</td>
<td>Failed to delete password of user on server</td>
</tr>
<tr>
<td>-5</td>
<td>Current user not authorized to change password of <code>user</code></td>
</tr>
</tbody>
</table>

### 2.28.5 See Also

The PBS Professional Administrator’s Guide
Deprecation. Reports PBS diagnostic information and fixes permission errors

2.29.1 Synopsis

```
pbs_probe [-f] [-v ]
```

```
pbs Probe --version
```

2.29.2 Description

The `pbs_probe` command reports post-installation information useful for PBS diagnostics, and fixes permission errors.

2.29.2.1 Information Sources

- Information that is supplied on the command line
- The file `/etc/pbs.conf`
- The file `/etc/init.d/pbs`
- The values of any of the following environment variables; these may be set in the environment in which `pbs probe` is run: `PBS_CONF_FILE`, `PBS_HOME`, `PBS_EXEC`, `PBS_START_SERVER`, `PBS_START_MOM`, and `PBS_START_SCHED`

2.29.2.2 Required Privilege

In order to execute `pbs probe`, you must have PBS Operator or Manager privilege.

2.29.3 Options to `pbs probe`

(no options)

- Run in "report" mode. In this mode `pbs probe` reports any permission errors detected in PBS infrastructure files. The command categorizes the errors and writes a list of them by category. Empty categories are not written.

- `-f`

  Run in "fix" mode. In this mode `pbs probe` examines each of the relevant infrastructure files and, where possible, fixes any permission errors that it detects, and prints a message saying what got changed. If it is unable to fix a problem, it prints a message saying what was detected.

- `-v`

  Run in "verbose" mode. In this mode `pbs probe` writes a complete list of the infrastructure files that it checked.

- `--version`

  The `pbs probe` command returns its PBS version information and exits. This option can only be used alone.

2.29.4 Standard Error

The `pbs probe` command writes a diagnostic message to standard error for each error occurrence.
2.29.5 Exit Status

Exit code does not reflect results of probe; it reflects whether or not the program ran.

Zero
When run correctly, whether or not `pbs_probe` finds any problems or errors

Non-negative
When run incorrectly

2.29.6 See Also

The PBS Professional Administrator’s Guide
2.30  pbs_python

Python interpreter for debugging a hook script from the command line

2.30.1  Synopsis

```
```

```
pbs_python <standard Python options>
```

```
pbs_python --version
```

2.30.2  Description

The PBS Python interpreter, `pbs_python`, is a wrapper for Python. You can use the `pbs_python` wrapper that is shipped with PBS to debug hooks. Either:

- Use the `--hook` option to `pbs_python` to run `pbs_python` as a wrapper to Python, employing the `pbs_python` options. With the `--hook` option, you cannot use the standard Python options. The rest of this section covers how to use `pbs_python` with the `--hook` option.
- Do not use the `--hook` option, so `pbs_python` runs the Python interpreter, with the standard Python options, and without access to the `pbs_python` options.

2.30.2.1  Debugging Hooks

You can get each hook to write out debugging files, and then modify the files and use them as debugging input to `pbs_python`. Alternatively, you can write the files yourself.

Debugging files can contain information about the event, about the site, and about what the hook changed. You can use these as inputs to a hook when debugging.

For a complete description of using `pbs_python` with debugging files, see "Debugging Hooks" on page 149 in the PBS Professional Hooks Guide.

2.30.3  Options to `pbs_python`

- `--hook`
  
  This option is a switch. When you use this option, you can use the PBS Python module (via "import pbs"), and the other options described here are available. When you use this option, you cannot use the standard Python options. This option is useful for debugging.

  When you do not use this option, you cannot use the other options listed here, but you can use the standard Python options.

- `-e <log event mask>`
  
  Sets the mask that determines which event types are logged by `pbs_python`. To see only debug messages, set the value to `0xd80`. To see all messages, set the value to `0xffffffff`. The `pbs_python` interpreter uses the same set of mask values that are used for the `$logevent <mask>` entry in the `pbs_mom` configuration file. See section 2.24, "pbs_mom", on page 72. Available only when `--hook` option is used.

- `-i <event input file>`
  
  Text file containing data to populate `pbs.event()` objects. Each line specifies an attribute value or a resource value. Syntax of each input line is one of the following:
<object name>.<attribute name>=<attribute value>
<object name>.<resource list>[<resource name>]=<resource value>

Where

<object name> is a PBS object name which can refer to its sub-objects. Examples: "pbs.event()", "pbs.event().job", "pbs.event().vnode_list["<vnode name>"]".

Example input file:

```bash
pbs.event().hook_name=proto
pbs.event().hook_type=site
pbs.event().type=queuejob
pbs.event().requestor=user1
pbs.event().requestor_host=host1
pbs.event().alarm=40
pbs.event().job.id=72
pbs.event().job.Job_Name=job1
pbs.event().job.Resource_List[ncpus]=5
pbs.event().job.Resource_List[mem]=6mb
pbs.event().vnode_list["host1"].resources_available["ncpus"] = 5
pbs.event().vnode_list["host1"].resources_available["mem"] = 300gb
```

Available only when --hook option is used.

-L <log dir>

Directory holding the log file where pbs.logmsg() and pbs.logjobmsg() write their output. Default is current working directory where pbs_python is executed. Available only when --hook option is used.

-l <log file>

Log file where pbs.logmsg() and pbs.logjobmsg() write their output. Default file name is current date in yyyyymmdd format. Available only when --hook option is used.

-o <hook execution record>

The hook execution record contains the changes made after executing the hook script, such as the attributes and resources set in any pbs.event() jobs and reservations, whether an action was accepted or rejected, and any pbs.reject() messages.

Example hook execution record:

```bash
pbs.event().job.Job_Name=job2
pbs.event().job.Resource_List[file]=60gb
pbs.event().job.Resource_List[ncpus]=5
pbs.event().job.Resource_List[mem]=20gb
pbs.event().job.Account_Name=account2
pbs.event().reject=True
pbs.event().reject_msg=No way!
```

Without this option, output goes to stdout. Available only when --hook option is used.

-r <resourcedef file>

File/path name containing a resource definition specifying a custom resource whose Python type is pbs.resource. Format:

```
<resource name> type=<typename> [flag=<value>]
```

Available only when --hook option is used.
Chapter 2  

PBS Commands

-s <site data file>
The site data file can contain any relevant information about the server, queues, vnodes, and jobs at the server. This file can be written by a hook or by the administrator.

When the hook writes it, this file contains the values that populate the server, queues, vnodes, reservations, and jobs, with all attributes and resources for which there are values.

The site data file is named hook_<event type>_<hook name>_<random integer>.data. It can be passed to pbs_python using the -s <site data file> option.

Available only when --hook option is used.

--version
The pbs_python command prints its version information and exits. This option can only be used alone.

2.30.4 Arguments

<Python script>
The hook script to execute. We recommend importing the PBS Python module at the start of the script:

import pbs

If you do not specify <Python script>, you can perform interactive debugging. If you type the following:

% pbs_python --hook -i hook.input
The interpreter displays a prompt:

>>

You can type your Python lines at the prompt:

>>import pbs
>> e=pbs.event().job
>> print e.id
<job-id>
...

...
2.31 pbs_ralter

Modifies an existing advance or standing reservation

2.31.1 Summary

Alter an existing advance or standing reservation.

2.31.2 Synopsis

```bash
pbs_ralter [-E <end time>] [-I <block time>] [-m <mail points>] [-M <mail list>] [-N <reservation name>] [-R <start time>] <reservation ID>
```

2.31.3 Description

You can use the `pbs_ralter` command to alter an existing advance or standing reservation. You can change the start time, end time, events that generate mail, mail recipient list, and reservation name. You can change the end time for a running reservation with running jobs.

You can use this command to change an advance reservation or the next or current instance of a standing reservation.

After the change is requested, the change is either confirmed or denied. On denial of the change, the reservation is not deleted and is left as is, and the following message appears in the server’s log:

```
Unable to alter reservation <reservation ID>
```

When a reservation is confirmed, the following message appears in the server’s log:

```
Reservation alter successful for <reservation ID>
```

To find out whether or not the change was allowed:

- Use the `pbs_rstat` command: see whether you altered reservation attribute(s)
- Use the interactive option: check for confirmation after the blocking time has run out
- Check the server log for confirmation or denial messages

Before the change is confirmed or denied, the change is unconfirmed, and the reservation state is `AL`.

Once a reservation change is confirmed, the reservation state is `CO` or `RN`.

If the reservation has not started and it cannot be confirmed on the same vnodes, PBS searches for another set of vnodes.

2.31.3.1 Required Privilege

You must be the reservation owner or the PBS Administrator to run this command.
2.31.4 Options to pbs_ralter

-R <start time>
Specifies reservation's new start time. This option can be used either when the reservation is not running or there are no jobs are submitted to the reservation. You cannot use this option when a reservation is not empty and has started running.

The specifications for providing the time are the same as for pbs_rsub:
If the day, DD, is not specified, it defaults to today if the time hhmm is in the future. Otherwise, the day is set to tomorrow. For example, if you alter a reservation with the specification -R 1110 at 11:15 a.m., it is interpreted as being for 11:10 a.m. tomorrow. If the month portion, MM, is not specified, it defaults to the current month, provided that the specified day DD, is in the future. Otherwise, the month is set to next month. Similar rules apply to the two other optional, left-side components.
Format: Datetime

-E <end time>
Specifies reservation's new end time. This option can be used even when the reservation is running and has jobs that are submitted to and/or are running in the reservation.
Format: Datetime

-I <block time>
Specifies interactive mode. The pbs_ralter command will block, up to block time seconds, while waiting for the reservation's change request to be confirmed or denied.

The value for block time must be positive. The pbs_ralter command returns either the status “CONFIRMED” or the status “DENIED”.
Format: Integer
Default: Not interactive

-m <mail points>
Specifies the set of events that cause mail to be sent to the list of users specified in the -M <mail list> option.
Format: string consisting of 1) any combination of “a”, “b”, “c” or “e”, or 2) the single character “n”.

Table 2-5: Suboptions to -m Option

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Notify if reservation is terminated for any reason</td>
</tr>
<tr>
<td>b</td>
<td>Notify when the reservation period begins</td>
</tr>
<tr>
<td>c</td>
<td>Notify when the reservation is confirmed</td>
</tr>
<tr>
<td>e</td>
<td>Notify when the reservation period ends</td>
</tr>
<tr>
<td>n</td>
<td>Send no mail. Cannot be used with any of a, b, c or e.</td>
</tr>
</tbody>
</table>

Default: No default; if not specified, mail events are unchanged.

-M <mail list>
The list of users to whom mail is sent whenever the reservation transitions to one of the states specified in the -m <mail points> option.
Format: `<username>[@<hostname>],[<username>[@<hostname>]]...`
Default: No default; if not specified, user list is unchanged.

-N <reservation name>
  Specifies a name for the reservation.
  Format: String up to 15 characters in length. It must consist of printable, non-white space characters with the first character alphabetic.
  Default: No default; if not specified, reservation name is unchanged.

2.31.5 Operands

The `pbs_ralter` command takes a reservation ID.
For an advance reservation this has the form:
```
R<sequence number>.[<server name>][@<remote server>]
```
For a standing reservation this has the form:
```
S<sequence number>.[<server name>][@<remote server>]
```
`@<remote server>` specifies a reservation at a server other than the default server.
2.32 pbs_rdel

Deletes a PBS advance or standing reservation

2.32.1 Synopsis

`pbs_rdel <reservation ID>[.<reservation ID>...]`
`pbs_rdel --version`

2.32.2 Description

The `pbs_rdel` command deletes reservations in the order specified.
This command deletes the specified reservations, whether or not they are running, all jobs in the reservations, and the reservation queues.

2.32.2.1 Required Privilege

A reservation may be deleted by its owner, a PBS Operator, or a PBS Manager.

2.32.3 Options

   --version

   The `pbs_rdel` command returns its PBS version information and exits. This option can only be used alone.

2.32.4 Operands

The `pbs_rdel` command accepts one or more `reservation ID` operands.
For an advance reservation this has the form:
`R<sequence number>[.<server name>][@<remote server>]`
For a standing reservation this has the form:
`S<sequence number>[.<server name>][@<remote server>]`
`@<remote server>` specifies a reservation at a server other than the default server.

2.32.5 Exit Status

   Zero
     Upon success
   Greater than zero
     Upon failure to process any operand

2.32.6 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide, "pbs_rsub" on page 97, "pbs_rstat" on page 95, "Reservation Attributes" on page 317
2.33  pbs_release_nodes

Releases vnodes assigned to a PBS job

2.33.1  Synopsis

pbs_release_nodes [-j <job ID>] <vnode> [<vnode> [<vnode>] ...]
pbs_release_nodes [-j <job ID>] -a
pbs_release_nodes --version

2.33.2  Description

You can use the pbs_release_nodes command to release no-longer-needed vnodes assigned to a running job, before the job would normally release them. These vnodes are then available for use by other jobs.

You can specify the names of vnodes to be released, or you can release all vnodes not on the primary execution host that are assigned to a running job via the -a option.

2.33.2.1  Caveats and Restrictions

• You can release only vnodes that are not on the primary execution host. You cannot release vnodes on the primary execution host.
• The job must be running (in the R state).
• The pbs_release_nodes command is not supported on vnodes tied to Cray X* series systems (vnodes whose vntype has the "cray_" prefix).
• The pbs_release_nodes command is not supported for vnodes managed by cpuset MoMs; partial release of vnodes may result in leftover cpusets.
• If cgroups support is enabled, and pbs_release_nodes is called to release some but not all the vnodes managed by a MoM, resources on those vnodes that are part of a cgroup are not released until the entire cgroup is released.

2.33.2.2  Required Privilege

This command can be run by the job owner, the PBS Manager, Operator, and Administrator, as well as root on Linux and Admin on Windows.

2.33.3  Options to pbs_release_nodes

-a
    Releases all job vnodes not on the primary execution host. Cannot be used with list of vnode names.
-j <job ID>
    Specifies the job ID for the job whose vnode(s) are to be released.
(no options)
    Without the -j option, pbs_release_nodes uses the value of the PBS_JOBID environment variable as the job ID of the job whose vnodes are to be released.
--version
    The pbs_release_nodes command returns its PBS version information and exits. This option can only be used alone.
2.33.4 Operands for pbs_release_nodes

The pbs_release_nodes command can take as an operand a list of vnodes. Format:

\[<vnode\ name>\ [<vnode\ name>\ [<vnode\ name>\ ]\ ...]\]

Cannot be used with the -a option.

2.33.5 Usage

This command can be run at the command line, or called inside a job script, where it can use the value of the PBS_JOBID environment variable.

You can release any vnode that appears in the job's exec_vnode attribute that is not on the primary execution host. You can release a particular set of a job’s vnodes, or you can release all of a job’s non-primary-execution-host vnodes.

To release specific vnodes:

\[pbs\_release\_nodes\ [-j\ <job\ ID>\ ]\ <vnode\ name>\ [<vnode\ name>\ ]\ ...]\]

To release all of a job’s vnodes that are not on the primary execution host:

\[pbs\_release\_nodes\ [-j\ <job\ ID>\ ]\ -a\]
2.34  pbs_rstat

Shows status of PBS advance or standing reservations

2.34.1  Synopsis

```
pbs_rstat [-B] [-F] [-S] [<reservation ID>...]
pbs_rstat --version
```

2.34.2  Description

The `pbs_rstat` command shows the status of all reservations at the PBS server. Denied reservations are not displayed.

2.34.2.1  Required Privilege

This command can be run by a user with any level of PBS privilege. For full output, users without manager or operator privilege cannot print custom resources which were created to be invisible to users.

2.34.3  Output

The `pbs_rstat` command displays output in any of brief, short, or full formats.

See section 6.8, “Reservation Attributes”, on page 317 and section 8.6, “Reservation States”, on page 377.

2.34.4  Options to `pbs_rstat`

- `-B`
  Brief output. Displays each reservation identifier only.

- `-F`
  Full output. Displays all reservation attributes that are not set to the default value. Users without manager or operator privilege cannot print custom resources which were created to be invisible to users.

- `-S`
  Short output. Displays a table showing the name, queue, owner, state, start time, duration, and end time of each reservation.

- `--version`
  The `pbs_rstat` command returns its PBS version information and exits. This option can only be used alone.

(no options)
  Short output. Same behavior as `-S` option.

2.34.5  Operands

The `pbs_rstat` command accepts one or more `reservation ID` operands.

For an advance reservation this has the form:

```
R<sequence number>[.<server name>][@<remote server>]
```
For a standing reservation this has the form:

S<sequence number>[.<server name>][[@<remote server>]]

@<remote server> specifies a reservation at a server other than the default server.

2.34.6 See Also

2.35 pbs_rsub

Creates a PBS advance or standing reservation

2.35.1 Synopsis

```
pbs_rsub [-D <duration>] [-E <end time>] [-g <group list>] [-G <auth group list>] [-H <auth host list>] [-I <block
  time>] [-m <mail events>] [-M <mail list>] [-N <reservation name>] [-q <destination>] [-r <recurrence rule>]
  [-R <start time>] [-u <user list>] [-U <auth user list>] [-W <attribute value list>] -l <resource request> [-l
  <placement>]
```

```
pbs_rsub --version
```

2.35.2 Description

The `pbs_rsub` command is used to create an advance or standing reservation. An advance reservation reserves specific
resources for the requested time period, and a standing reservation reserves specific resources for recurring time periods.
When a reservation is created, it has an associated queue.

After the reservation is requested, it is either confirmed or denied. Once the reservation has been confirmed, authorized
users submit jobs to the reservation’s queue via `qsub` and `qmove`.

A confirmed reservation will accept jobs at any time. The jobs in its queue can run only during the reservation period,
whether during a single advance reservation or during the occurrences of a standing reservation.

When an advance reservation ends, all of its jobs are deleted, whether running or queued. When an occurrence of a
standing reservation ends, only its running jobs are deleted; those jobs still in the queue are not deleted.

To get information about a reservation, use the `pbs_rstat` command.

To delete a reservation, use the `pbs_rdel` command. Do not use the `qdel` command.

The behavior of the `pbs_rsub` command may be affected by any site hooks. Site hooks can modify the reservation’s
attributes.

2.35.2.1 Requirements

When using `pbs_rsub` to request a reservation, you must specify two of the following options: -R, -E, and -D. The
resource request `-l walltime` can be used instead of the `-D` option.

If you want to run jobs in the reservation that will request exclusive placement, you must create the reservation with
exclusive placement via `-l place=excl`.

2.35.3 Options to `pbs_rsub`

- `-D <duration>`
  Specifies reservation duration. If the start time and end time are the only times specified, this duration time is
calculated.
  Format: `Duration`
  Default: none

- `-E <end time>`
  Specifies the reservation end time. If start time and duration are the only times specified, the end time value is
calculated.
Format: Datetime.
Default: none
-g <group_list>
The group list is a comma-separated list of group names. The server uses entries in this list, along with an ordered set of rules, to associate a group name with the reservation. The reservation creator’s primary group is automatically added to this list.
Format: <group>@<hostname>[,<group>@<hostname> ...]
-G <auth group list>
Comma-separated list of names of groups who can or cannot submit jobs to this reservation. Group names are interpreted in the context of the server host, not the context of the host from which the job is submitted.
This list becomes the acl_groups list for the reservation’s queue. More specific entries should be listed before more general, because the list is read left-to-right, and the first match determines access.
If both the Authorized_Users and Authorized_Groups reservation attributes are set, a user must belong to both in order to be able to submit jobs to this reservation.
Refer to the Authorized_Groups reservation attribute in section 6.8, “Reservation Attributes”, on page 317.
Format: [+|-]<group name>[,+|-]<group name> ...
Default: All groups are authorized to submit jobs.
-H <auth host list>
Comma-separated list of hosts from which jobs can and cannot be submitted to this reservation. This list becomes the acl_hosts list for the reservation’s queue. More specific entries should be listed before more general, because the list is read left-to-right, and the first match determines access. If the reservation creator specifies this list, the creator’s host is not automatically added to the list.
See the Authorized_Hosts reservation attribute in section 6.8, “Reservation Attributes”, on page 317.
Format: [+|-]<hostname>[,+|-]<hostname> ...
Default: All hosts are authorized to submit jobs
-I <block time>
Specifies interactive mode. The pbs_rsub command will block, up to block time seconds, while waiting for the reservation request to be confirmed or denied.
If block time is positive, and the reservation isn’t confirmed or denied in the specified time, the ID string for the reservation is returned with the status “UNCONFIRMED”.
If block time is negative, and a scheduler doesn’t confirm or deny the reservation in the specified time, the reservation is deleted.
Format: Integer
Default: Not interactive.
-I <placement>
The placement specifies how vnodes are reserved. The place statement can contain the following elements, in any order:
-I place=[<arrangement>]:[<sharing>]:[<grouping>]]
where
arrangement
Whether this reservation chunk is willing to share this vnode or host with other chunks from this reservation. One of free | pack | scatter | vscatter
sharing
Whether this reservation chunk is willing to share this vnode or host with other reservations or jobs. One of excl | shared | exclhost

**grouping**

Whether the chunks from this reservation should be placed on vnodes that all have the same value for a resource. Can have only one instance of group=<resource name>.

**free**

Place reservation on any vnode(s).

**pack**

All chunks are taken from one host.

**scatter**

Only one chunk with any MPI processes is taken from a host. A chunk with no MPI processes may be taken from the same vnode as another chunk.

**vscatter**

Only one chunk is taken from any vnode. Each chunk must fit on a vnode.

**excl**

Only this reservation uses the vnodes chosen.

**shared**

This reservation can share the vnodes chosen.

**exclhost**

The entire host is allocated to the reservation.

**group=<resource name>**

Chunks are grouped according to the specified resource. All vnodes in the group must have a common value for resource, which can be either the built-in resource host or a custom vnode-level resource.

Resource name must be a string or a string array.

If you want to run jobs in the reservation that will request exclusive placement, you must create the reservation with exclusive placement via -l place=excl.

The place statement cannot start with a colon. Colons are delimiters; use them only to separate parts of a place statement, unless they are quoted inside resource values.

Note that vnodes can have sharing attributes that override reservation placement requests.

See section 6.10, “Vnode Attributes”, on page 333.
-l <resource request>

The resource request specifies the resources required for the reservation. These resources are used for the limits on the queue that is dynamically created for the reservation. The aggregate amount of resources for currently running jobs from this queue will not exceed these resource limits. Jobs in the queue that request more of a resource than the queue limit for that resource are not allowed to run. Also, the queue inherits the value of any resource limit set on the server, and these are used for the job if the reservation request itself is silent about that resource. A non-privileged user cannot submit a reservation requesting a custom resource which has been created to be invisible or read-only for users.

Resources are requested by using the -l option, either in chunks inside of selection statements, or in job-wide requests using <resource name>=<value> pairs.

Requesting resources in chunks:
-l select=[N:]<chunk>[/+[N:]<chunk> ...]

where N specifies how many of that chunk, and a chunk is of the form:
<resource name>=<value>[:<resource name>=<value> ...]

Requesting job-wide resources:
-l <resource name>=<value>[:<resource name>=<value> ...]

-m <mail events>

Specifies the set of events that cause mail to be sent to the list of users specified in the -M <mail list> option.

Format: string consisting of one of the following:
• Any combination of “a”, “b”, “c” or “e”
• The single character “n”

The following table lists the sub-options to the -m option:

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Notify if the reservation is terminated for whatever reason</td>
</tr>
<tr>
<td>b</td>
<td>Notify when the reservation period begins</td>
</tr>
<tr>
<td>c</td>
<td>Notify when the reservation is confirmed</td>
</tr>
<tr>
<td>e</td>
<td>Notify when the reservation period ends</td>
</tr>
<tr>
<td>n</td>
<td>Send no mail. Cannot be used with any of a, b, c, or e.</td>
</tr>
</tbody>
</table>

Default: “ac”.

-M <mail list>

The list of users to whom mail is sent whenever the reservation transitions to one of the states specified in the -m <mail events> option.

Format: <username>[@<hostname>],<username>[@<hostname>] ...

Default: Reservation owner.

-N <reservation name>

Specifies a name for the reservation.

Format: Reservation Name. See "Reservation Name" on page 367.

Default: None.
-q <destination>
  Specifies the destination server at which to create the reservation.
  Default: The default server is used if this option is not selected.

-r <recurrence rule>
  Specifies rule for recurrence of standing reservations.  Rule must conform to iCalendar syntax, and is specified
  using a subset of parameters from RFC 2445.
  Valid syntax for recurrence rule takes one of two forms:
  
  \[ FREQ=<freq\ spec>;<COUNT=<count\ spec>;<interval\ spec> \]
  or
  
  \[ FREQ=<freq\ spec>;<UNTIL=<until\ spec>;<interval\ spec> \]
  
  where

  \[ freq\ spec \]
  Frequency with which the standing reservation repeats.  Valid values are:
  
  \[ WEEKLY|DAILY|HOURLY \]

  \[ count\ spec \]
  The exact number of occurrences.  Number up to 4 digits in length.
  
  Format: Integer.

  \[ interval\ spec \]
  Specifies interval.  Format is one or both of:
  
  \[ BYDAY=MO|TU|WE|TH|FR|SA|SU \]
  or
  
  \[ BYHOUR=0|1|2|...|23 \]
  
  When using both, separate them with a semicolon.
  Elements specified in the recurrence rule override those specified in the arguments to the -R and -E options.
  For example, the BYHOUR specification overrides the hourly part of the -R option.  For example, -R 0730 -E 0830 ...
  BYHOUR=9 results in a reservation that starts at 9:30 and runs for 1 hour.

  \[ until\ spec \]
  Occurrences will start up to but not after date and time specified.  Format:
  
  \[ <YYYYMMDD>T<HHMMSS> \]
Note that the year-month-day section is separated from the hour-minute-second section by a capital T.

Requirements:

- The recurrence rule must be on one unbroken line and must be enclosed in double quotes.
- A start and end date must be used when specifying a recurrence rule. See the R and E options.
- The PBS_TZID environment variable must be set at the submission host. The format for PBS_TZID is a timezone location. Examples: America/Los_Angeles, America/Detroit, Europe/Berlin, Asia/Calcutta. See the PBS Professional User’s Guide.
- Spaces are not allowed.

Examples of Standing Reservations

For a reservation that runs every day from 8am to 10am, for a total of 10 occurrences:

```
pbs_rsub -R 0800 -E 1000 -r "FREQ=DAILY;COUNT=10"
```

Every weekday from 6am to 6pm until December 10 2008

```
pbs_rsub -R 0600 -E 1800 -r "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR;UNTIL=20081210"
```

Every week from 3pm to 5pm on Monday, Wednesday, and Friday, for 9 occurrences, i.e., for three weeks:

```
pbs_rsub -R 1500 -E 1700 -r "FREQ=WEEKLY;BYDAY=MO,WE,FR;COUNT=3"
```

-R <start time>

Specifies reservation starting time. If the reservation’s end time and duration are the only times specified, this start time is calculated.

If the day, DD, is not specified, it defaults to today if the time hhmm is in the future. Otherwise, the day is set to tomorrow. For example, if you submit a reservation with the specification -R 1110 at 11:15 a.m., it is interpreted as being for 11:10am tomorrow. If the month portion, MM, is not specified, it defaults to the current month, provided that the specified day DD, is in the future. Otherwise, the month is set to next month. Similar rules apply to the two other optional, left-side components.

Format: Datetime

-u <user list>

Not used. Comma-separated list of user names.

Format: <username>[@<hostname>],[<username>[@<hostname>],...]

Default: None.

-U <auth user list>

Comma-separated list of users who are and are not allowed to submit jobs to this reservation. This list becomes the acl_users attribute for the reservation’s queue. More specific entries should be listed before more general, because the list is read left-to-right, and the first match determines access.

If both the Authorized_Users and Authorized_Groups reservation attributes are set, a user must belong to both in order to be able to submit jobs to this reservation. The reservation creator’s username is automatically added to this list, whether or not the reservation creator specifies this list.

Refer to the Authorized_Users reservation attribute in section 6.8, “Reservation Attributes”, on page 317.

Format: [+]<username>[@<hostname>],[[+]<username>[@<hostname>]...]

Default: Job owner only.

-W <attribute value list>

This allows you to define other attributes for the reservation.

Supported attributes:

```
qmove=<job ID> [-I -<timeout>]
```
PBS Commands

Converts a normal job designated by job ID into a reservation job that will run as soon as possible. Creates the reservation with its queue and moves the job into the reservation’s queue. Uses the resources requested by the job to create the reservation.

When the reservation is created, it inherits its resources from the job, not from the resources requested through the pbs_rsub command.

You can use the -I option to specify a timeout for the conversion. If you use the qmove option to convert a job to a reservation, and the reservation is not confirmed within the timeout period, the reservation is deleted. The default timeout period is 10 seconds. There is no option for this kind of reservation to be unconfirmed.

To specify the timeout, you must give a negative value for the -I option. For example, to specify a timeout of 300 seconds:

```
pbs_rsub -Wqmove=<job ID> -I -300
```

The -R and -E options to pbs_rsub are disabled when using the qmove=<job ID> option.

Some shells require that you enclose a job array ID in double quotes.

```bash
--version
```

The pbs_rsub command returns its PBS version information and exits. This option can only be used alone.

### 2.35.4 Output

The pbs_rsub command returns the reservation identifier.

For an advance reservation, this has the form

```
R<NNNN>.<server name>
```

where NNNN is a unique integer. The associated queue’s name is the prefix, R<NNNN>.

For a standing reservation, this has the form

```
S<NNNN>.<server name>
```

where <NNNN> is a unique integer. The associated queue’s name is the prefix, S<NNNN>.

### 2.35.5 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide, "pbs_rstat" on page 95, "pbs_rdel" on page 92, "Reservation Attributes" on page 317
2.36 pbs_sched

Runs a PBS scheduler

2.36.1 Synopsis


`pbs_sched --version`

2.36.2 Description

`pbs_sched` is a PBS scheduling daemon. It schedules PBS jobs.

2.36.2.1 Required Permission

`pbs_sched` must be executed with root permission on Linux and Admin privilege on Windows.

2.36.3 Options to `pbs_sched`

- `-a <alarm>`
  Deprecated. Overwrites value of `sched_cycle_length` scheduler attribute. Time in seconds to wait for a scheduling cycle to finish.
  Format: *Time, in seconds*.

- `-c <clientsfile>`
  Add clients to this scheduler’s list of known clients. The `clientsfile` contains single-line entries of the form
  `$clienthost <hostname>`
  Each `hostname` is added to the list of hosts allowed to connect to this scheduler. If `clientsfile` cannot be opened, this scheduler aborts. Path can be absolute or relative. If relative, it is relative to `PBS_HOME/sched_priv`.

- `-d <home dir>`
  The directory in which this scheduler will run.
  Default: `PBS_HOME/sched_priv`.

- `-I <scheduler name>`
  Name of scheduler to start. Required when starting a multisched.

- `-L <logfile>`
  The absolute path and filename of the log file. This scheduler writes its PBS version and build information to `logfile` whenever it starts up or `logfile` is rolled to a new file.
  See the `-d` option.
  Default: This scheduler opens a file named for the current date in the `PBS_HOME/sched_log` directory.

- `-n`
  Tells this scheduler to not restart itself if it receives a `sigsegv` or a `sigbus`. A scheduler by default restarts itself if it receives either of these two signals more than five minutes after starting. A scheduler does not restart itself if it receives either one within five minutes of starting.
-N
Instructs this scheduler not to detach itself from the current session.

-p <output file>
Any output which is written to standard out or standard error is written to output file. The pathname can be absolute or relative, in which case it is relative to PBS_HOME/sched_priv.

See the -d option.
Default: PBS_HOME/sched_priv/sched_out

-R <port number>
The port for MoM to use. If this option is not given, the port number is taken from PBS_MANAGER_SERVICE_PORT, in pbs.conf.
Default: 15003

-S <port number>
The port for this scheduler to use.
Required when starting a multisched.
For the default scheduler, if this option is not specified, the default port is taken from PBS_SCHEDULER_SERVICE_PORT, in pbs.conf.
Default value for default scheduler: 15004
Default value for multisched: none

--version
The pbs_sched command returns its PBS version information and exits. This option can only be used alone.

### 2.36.4 Signal Handling

All signals are ignored until the end of the cycle. Most signals are handled in the standard UNIX fashion.

SIGHUP
This scheduler closes and reopens its log file and rereads its configuration file if one exists.

SIGALRM, SIGBUS, etc.
Ignored until end of scheduling cycle. This scheduler quits.

SIGINT and SIGTERM
This scheduler closes its log file and shuts down.

All other signals have the default action installed.

### 2.36.5 Exit Status

Zero
Upon normal termination

### 2.36.6 See Also

The PBS Professional Administrator’s Guide
2.37  pbs_server

Starts a PBS batch server

2.37.1  Synopsis

```
```

2.37.2  Description

The `pbs_server` command starts a batch server on the local host. Typically, this command is in a local boot file such as `/etc/rc.local`. If the batch server is already running, `pbs_server` exits with an error.

2.37.2.1  Required Permission

To ensure that the `pbs_server` command is not runnable by the general user community, the server runs only if its real and effective UID is zero. You must be root on Linux or Admin on Windows.

2.37.3  Options to pbs_server

- `-A <acctfile>`
  Specifies an absolute path name for the file to use as the accounting file. If not specified, the file is named for the current date in the `PBS_HOME/server_priv/accounting` directory.

- `-a <value>`
  When `True`, the server is in state “active” and the default scheduler is called to schedule jobs. When `False`, the server is in state “idle” and the default scheduler is not called to schedule jobs. Sets the server’s scheduling attribute. If this option is not specified, the server uses the previously specified value for the scheduling attribute.
  Format: Boolean

- `-C`
  The server starts up, creates the database, and exits. Windows only.

- `-d <config path>`
  Specifies the absolute path to the directory containing the server configuration files, `PBS_HOME`. A host may have multiple servers. Each server must have a different configuration directory. The default configuration directory is specified in `$PBS_HOME`, and is typically `/var/spool/pbs`.

- `-e <mask>`
  Specifies a log event mask to be used when logging. See “log_events” in section 6.6, “Server Attributes”, on page 295.

- `-F <delay>`
  Specifies the number of seconds that the secondary server should wait before taking over when it believes the primary server is down. If the number of seconds is specified as `-1`, the secondary will make one attempt to contact the primary and then become active.
  Default: 30 seconds
PBS Commands

Chapter 2

-L <logfile>
Specifies the absolute path name for the log file. If not specified, the file is named for the current date in the
PBS_HOME/server_logs directory. PBS_HOME is specified in the $PBS_HOME environment variable or in
/etc/pbs.conf; see the -d option.

-M <MoM port>
Specifies the hostname and/or port number on which the server should connect to MoM. The option argument,
MoM port, uses the syntax:

[<hostname>][:<port number>]
If hostname not specified, the local host is assumed.
If port number is not specified, the default port is assumed.
See the -M option in section 2.24, “pbs_mom”, on page 72.
Default: 15002

-N
The server runs in standalone mode. On Windows, it does not register as a Windows service. On other plat-
forms, MoM does not detach from the current session.

-p <port number>
Specifies the port number on which the server is to listen for batch requests. If multiple servers are running on
a single host, each must have its own unique port number. This option is for testing with multiple batch systems
on a single host.
Format: Integer port number
Default: 15001

-R <MoM RPP port>
Specifies the port number on which the server should query the up/down status of MoM. See the -R option in
section 2.24, “pbs_mom”, on page 72.
Default: 15003

-S <default scheduler port>
Specifies the port number to which the server should connect when contacting the default scheduler. The option
argument, default scheduler port, uses the syntax:

[<hostname>][:<port number>]
If hostname not specified, the local host is assumed. If port number is not specified, the default port is assumed.
Default: 15004

-s <replacement string>
Specifies the string to use when replacing spaces in accounting entity names. Only available under Windows.

-t <restart type>
Specifies behavior when the server restarts. The restart type argument is one of the following:
cold
All jobs are purged. Positive confirmation is required before this direction is accepted.
create
The server discards any existing configuration files: server, nodes, queues, and jobs, and initializes config-
uration files to the default values. The default scheduler is idled (scheduling is set to False). Any multi-
scheds are deleted.
All jobs in the Running state are retained in that state. Any job that was requeued into the Queued state from the Running state when the server last shut down is run immediately, assuming the required resources are available. This returns the server to the same state as when it went down. After those jobs are restarted, normal scheduling takes place for all remaining queued jobs. All other jobs are retained in their current state.

If a job cannot be restarted immediately because of a missing resource, such as a vnode being down, the server attempts to restart it periodically for up to 5 minutes. After that period, the server will revert to a normal state, as if warm started, and will no longer attempt to restart any remaining jobs which were running prior to the shutdown.

**updatedb**
Updates format of PBS data from the previous format to the data service format.

**warm**
All jobs in the Running state are retained in that state. All other jobs are maintained in their current state. The default scheduler typically chooses new jobs for execution. **warm** is the default if -t is not specified.

--version
The pbs_server command returns its PBS version information and exits. This option can only be used alone.

## 2.37.4 Files

$PBS_HOME/server_priv
Default directory for configuration files.

$PBS_HOME/server_logs
Directory for log files recorded by the server.

## 2.37.5 Signal Handling for pbs_server

When it receives the following signals, the server performs the following actions:

- **SIGHUP**
  The current server log and accounting log are closed and reopened. This allows for the prior log to be renamed and a new log started from the time of the signal.

- **SIGTERM**
  Causes a rapid orderly shutdown of pbs_server, identical to “qterm -t quick”.

- **SIGSHUTDN**
  On systems where SIGSHUTDN is defined, causes an orderly “quick” shutdown of the server.

- **SIGPIPE, SIGUSR1, SIGUSR2**
  These signals are ignored.

All other signals have their default behavior installed.

## 2.37.6 Diagnostic Messages

The server records a diagnostic message in a log file for any error occurrence. The log files are maintained in the server_logs directory below the home directory of the server. If the log file cannot be opened, the diagnostic message is written to the system console. The server writes its PBS version and build information to the logfile whenever it starts up or the logfile is rolled to a new file.
2.37.7  Stopping the PBS Server

2.37.7.1  Stopping the Server on Linux

Use the `qterm` command (see section 2.62, “qterm”, on page 226):

```
qterm
```
or send a SIGTERM:

```
kil <server PID>
```

2.37.7.2  Stopping the Server on Windows

If you’re running “pbs_server -N” for a standalone mode server, use

```
<cntrl>-<break>
```

2.37.8  Exit Status

Zero

When the server has run in the background and then exits

Greater than zero

If the server daemon fails to begin batch operation

2.37.9  See Also

The PBS Professional Administrator’s Guide
2.38 **pbs_snapshot**

Linux only. Captures PBS data to be used for diagnostics

### 2.38.1 Synopsis

```
pbs_snapshot -o <output directory path> [-H <server host>] [-l <log level>] [--accounting-logs=<number of days>] [--additional-hosts=<hostname list>] [--daemon-logs=<number of days>] [--map=<file path>] [--obfuscate]
```

```
pbs_snapshot --version
```

### 2.38.2 Description

You use `pbs_snapshot` to capture PBS data for diagnostics. This tool is written in Python and uses PTL libraries, including PBSSnapUtils, to extract the data. You can optionally anonymize the PBS data.

#### 2.38.2.1 Required Privilege

To run `pbs_snapshot`, you must be root on Linux or Admin on Windows.

### 2.38.3 Options to pbs_snapshot

- **-H <server host>**
  
  Specifies server hostname. By default, `pbs_snapshot` uses the value of the PBS_SERVER parameter in `pbs.conf`. When you use this option, `pbs_snapshot` uses `server host` instead.

- **-l <log level>**
  
  Specifies level at which `pbs_snapshot` writes its log. The log file is `pbs_snapshot.log`, in the output directory path specified using the `-o <output directory path>` option.

  Valid values, from most comprehensive to least: `DEBUG2, DEBUG, INFOCLI2, INFOCLI, INFO, WARNING, ERROR, FATAL`

  Default: `INFOCLI2`

- **--accounting-logs=<number of days>**
  
  Specifies number of days of accounting logs to be collected; this count includes the current day.

  Value of `number of days` must be >=0:

  - If number of days is 0, no logs are captured.
  - If number of days is 1, only the logs for the current day are captured.

  Default: `pbs_snapshot` collects 30 days of accounting logs
--additional-hosts=<hostname list>

Specifies that pbs_snapshot should gather data from the specified list of non-server hosts.
pbs_snapshot always gathers data from the server host.

The command collects the following information from the specified hosts:

- MoM and comm logs, for the number of days of logs being captured, specified via the --daemon-logs=<number of days> option
- The PBS_HOME/mom_priv directory
- System information

Format for hostname list is a comma-separated list of one or more hostnames:
<hostname>[, <hostname> ...]

This option can greatly bloat the size of the snapshot, and cause pbs_snapshot to take a long time copying what may be large amounts of data over the network.

--daemon-logs=<number of days>

Specifies number of days of daemon logs to be collected; this count includes the current day. All daemon logs are captured on the server host, and if you specify --additional-hosts=<hostname list>, MoM logs are captured on those hosts as well.

Value of number of days must be >=0:
- If number of days is 0, no logs are captured.
- If number of days is 1, only the logs for the current day are captured.

Default: pbs_snapshot collects 5 days of daemon logs

--map=<file path>

Specifies path for file containing obfuscation map, which is a <key>:<value> pair-mapping of obfuscated data. Path can be absolute or relative to current working directory.

Default: pbs_snapshot writes its obfuscation map in a file called "obfuscate.map" in the location specified via the -o <output directory path> option.

Can only be used with the --obfuscate option.

--obfuscate

Obfuscates (anonymizes) or deletes sensitive PBS data captured by pbs_snapshot.

- Obfuscates the following data: euser, egroup, project, Account_Name, operators, managers, group_list, Mail_Users, User_List, server_host, acl_groups, acl_users, acl_resv_groups, acl_resv_users, sched_host, acl_resv_hosts, acl_hosts, Job_Owner, exec_host, Host, Mom, resources_available.host, resources_available.vnode
- Deletes the following data: Variable_List, Error_Path, Output_Path, mail_from, Mail_Points, Job_Name, jobdir, Submit_arguments, Shell_Path_List

--version

Prints its PBS version number and exits. Can only be used alone.

2.38.4 Arguments to pbs_snapshot

-o <output directory path>

Path to directory where pbs_snapshot writes its output tarball. Required. Path can be absolute or relative to current working directory.

For example, if you specify "-o /temp", pbs_snapshot writes "/temp/snapshot_<timestamp>.tgz".
The output directory path must already exist.
2.38.5 Output

2.38.5.1 Output Location

You must use the \(-o <output\ directory\ path>\) option to specify the directory where \texttt{pbs\_snapshot} writes its output. The path can be absolute or relative to the current working directory. The output directory must already exist. As an example, if you specify \("-o /temp\)\), \texttt{pbs\_snapshot} writes \("/temp/snapshot_<timestamp>.tgz\)\.

2.38.5.2 Output Contents

The \texttt{pbs\_snapshot} command writes its output as a tarball. The tarball contains the following directory structure and files:

\begin{table}[h]
\centering
\caption{Contents of Snapshot}
\begin{tabular}{|l|l|l|}
\hline
\textbf{Directory or File} & \textbf{Directory Contents} & \textbf{Description} \\
\hline
server/ & qstat\_B.out & Output of \texttt{qstat -B} \\
 & qstat\_Bf.out & Output of \texttt{qstat -Bf} \\
 & qmgr\_ps.out & Output of \texttt{qmgr print server} \\
 & qstat\_Q.out & Output of \texttt{qstat -Q} \\
 & qstat\_Qf.out & Output of \texttt{qstat -Qf} \\
 & qmgr\_pr.out & Output of \texttt{qmgr print resource} \\
\hline
server\_priv/ & Copy of the PBS\_HOME/server\_priv directory. Core files are captured separately; see \texttt{core\_file\_bt/}. & \\
accounting/ & Accounting logs from PBS\_HOME/server\_priv/accounting/ directory for the number of days specified via --accounting-logs option & \\
server\_logs/ & Server logs from the PBS\_HOME/server\_logs directory for the number of days specified via --daemon-logs option & \\
\hline
job/ & qstat.out & Output of \texttt{qstat} \\
 & qstat\_f.out & Output of \texttt{qstat -f} \\
 & qstat\_t.out & Output of \texttt{qstat -t} \\
 & qstat\_tf.out & Output of \texttt{qstat -tf} \\
 & qstat\_x.out & Output of \texttt{qstat -x} \\
 & qstat\_xf.out & Output of \texttt{qstat -xf} \\
 & qstat\_ns.out & Output of \texttt{qstat -ns} \\
 & qstat\_fx\_F\_dsv.out & Output of \texttt{qstat -fx -F dsv} \\
 & qstat\_f\_F\_dsv.out & Output of \texttt{qstat -f -F dsv} \\
\hline
\end{tabular}
\end{table}
### Table 2-7: Contents of Snapshot

<table>
<thead>
<tr>
<th>Directory or File</th>
<th>Directory Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pbsnodes_va.out</td>
<td>Output of pbsnodes -va</td>
</tr>
<tr>
<td></td>
<td>pbsnodes_a.out</td>
<td>Output of pbsnodes -a</td>
</tr>
<tr>
<td></td>
<td>pbsnodes_avSj.out</td>
<td>Output of pbsnodes -avSj</td>
</tr>
<tr>
<td></td>
<td>pbsnodes_aSj.out</td>
<td>Output of pbsnodes -aSj</td>
</tr>
<tr>
<td></td>
<td>pbsnodes_avS.out</td>
<td>Output of pbsnodes -avS</td>
</tr>
<tr>
<td></td>
<td>pbsnodes_aS.out</td>
<td>Output of pbsnodes -aS</td>
</tr>
<tr>
<td></td>
<td>pbsnodes_aFdsV.out</td>
<td>Output of pbsnodes -aFdsV</td>
</tr>
<tr>
<td></td>
<td>pbsnodes_avFdsV.out</td>
<td>Output of pbsnodes -avFdsV</td>
</tr>
<tr>
<td></td>
<td>qmgr_pn_default.out</td>
<td>Output of qmgr print node @default</td>
</tr>
<tr>
<td>mom_priv/</td>
<td>Copy of the PBS_HOME/mom_priv directory. Core files are captured separately; see core_file_bt/</td>
<td></td>
</tr>
<tr>
<td>mom_logs/</td>
<td>MoM logs from the PBS_HOME/mom_logs directory for the number of days specified via --daemon-logs option</td>
<td></td>
</tr>
<tr>
<td>comm_logs/</td>
<td>Comm logs from the PBS_HOME/comm_logs directory for the number of days specified via --daemon-logs option</td>
<td></td>
</tr>
<tr>
<td>sched_priv/</td>
<td>Copy of the PBS_HOME/sched_priv directory, with all files. Core files are not captured; see core_file_bt/</td>
<td></td>
</tr>
<tr>
<td>sched_log/</td>
<td>Scheduler logs from the PBS_HOME/sched_log directory for the number of days specified via --daemon-logs option</td>
<td></td>
</tr>
<tr>
<td>sched_priv_&lt;multi-sched name&gt;/</td>
<td>Copy of the PBS_HOME/sched_priv_&lt;multisched name&gt; directory, with all files. Core files are not captured; see core_file_bt/</td>
<td></td>
</tr>
<tr>
<td>sched_log_&lt;multi-sched name&gt;/</td>
<td>Multisched logs from the PBS_HOME/sched_log_&lt;multisched name&gt; directory for the number of days specified via --daemon-logs option</td>
<td></td>
</tr>
<tr>
<td>reservation/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pbs_rstat_f.out</td>
<td>Output of pbs_rstat -f</td>
</tr>
<tr>
<td></td>
<td>pbs_rstat.out</td>
<td>Output of pbs_rstat</td>
</tr>
<tr>
<td>scheduler/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>qmgr_lsched.out</td>
<td>Output of qmgr list sched</td>
</tr>
<tr>
<td>hook/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>qmgr_ph_default.out</td>
<td>Output of qmgr print hook @default</td>
</tr>
<tr>
<td></td>
<td>qmgr_pbshook.out</td>
<td>Output of qmgr list pbshook</td>
</tr>
<tr>
<td>datastore/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pg_log/</td>
<td>Copy of the PBS_HOME/datastore/pg_log directory for the number of days specified via --daemon-logs option</td>
</tr>
</tbody>
</table>
### Table 2-7: Contents of Snapshot

<table>
<thead>
<tr>
<th>Directory or File</th>
<th>Directory Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>core_file_bt/</td>
<td>Stack backtrace from core files</td>
<td></td>
</tr>
<tr>
<td>sched_priv/</td>
<td>Files containing the output of <code>thread apply all backtrace full</code> on all core files captured from PBS_HOME/sched_priv</td>
<td></td>
</tr>
<tr>
<td>sched_priv_&lt;multisched name&gt;/</td>
<td>Files containing the output of <code>thread apply all backtrace full</code> on all core files captured from PBS_HOME/sched_priv_&lt;multisched name&gt;</td>
<td></td>
</tr>
<tr>
<td>server_priv/</td>
<td>Files containing the output of <code>thread apply all backtrace full</code> on all core files captured from PBS_HOME/server_priv</td>
<td></td>
</tr>
<tr>
<td>mom_priv/</td>
<td>Files containing the output of <code>thread apply all backtrace full</code> on all core files captured from PBS_HOME/mom_priv</td>
<td></td>
</tr>
<tr>
<td>misc/</td>
<td>Files containing the output of <code>thread apply all backtrace full</code> on any other core files found inside PBS_HOME</td>
<td></td>
</tr>
<tr>
<td>system/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pbs_probe_v.out</td>
<td>Output of <code>pbs probe -v</code></td>
<td></td>
</tr>
<tr>
<td>pbs_hostn_v.out</td>
<td>Output of <code>pbs_hostn -v $(hostname)</code></td>
<td></td>
</tr>
<tr>
<td>pbs_environment</td>
<td>Copy of PBS_HOME/pbs_environment file</td>
<td></td>
</tr>
<tr>
<td>os_info</td>
<td>Information about the OS</td>
<td></td>
</tr>
<tr>
<td>process_info</td>
<td>List of processes running on the system when the snapshot was taken. Output of `ps -aux</td>
<td>grep [p]bs<code>on Linux systems, or</code>tasklist /v` on Windows systems</td>
</tr>
<tr>
<td>ps_leaf.out</td>
<td>Output of <code>ps -lleaf</code>. Linux only.</td>
<td></td>
</tr>
<tr>
<td>lsof_pbs.out</td>
<td>Output of `lsof</td>
<td>grep [p]bs`. Linux only.</td>
</tr>
<tr>
<td>etc_hosts</td>
<td>Copy of <code>/etc/hosts</code> file. Linux only.</td>
<td></td>
</tr>
<tr>
<td>etc_nsswitch_conf</td>
<td>Copy of <code>/etc/nsswitch.conf</code> file. Linux only.</td>
<td></td>
</tr>
<tr>
<td>vmstat.out</td>
<td>Output of the command <code>vmstat</code>. Linux only.</td>
<td></td>
</tr>
<tr>
<td>df_h.out</td>
<td>Output of the command <code>df -h</code>. Linux only.</td>
<td></td>
</tr>
<tr>
<td>dmesg.out</td>
<td>Output of the <code>dmesg</code> command. Linux only.</td>
<td></td>
</tr>
<tr>
<td>pbs.conf</td>
<td>Copy of the pbs.conf file on the server host</td>
<td></td>
</tr>
<tr>
<td>ctime</td>
<td>Contains the time in seconds since epoch when the snapshot was taken</td>
<td></td>
</tr>
<tr>
<td>pbs_snapshot.log</td>
<td>Log messages written by <code>pbs_snapshot</code></td>
<td></td>
</tr>
</tbody>
</table>
2.38.6 Examples

```bash
pbs_snapshot -o /tmp
```
Writes a snapshot to /temp/snapshot_<timestamp>.tgz that includes 30 days of accounting logs and 5 days of daemon logs from the server host.

```bash
pbs_snapshot --daemon-logs=1 --accounting-logs=1 -o /tmp --obfuscate --map=mapfile.txt
```
Writes a snapshot to /temp/snapshot_<timestamp>.tgz that includes 1 day of accounting and daemon logs. Obfuscates the data and stores the data mapping in the map file named “mapfile.txt”.
Chapter 2  

PBS Commands

2.39  **pbs_tclsh**

TCL shell with TCL-wrapped PBS API

### 2.39.1 Synopsis

`pbs_tclsh`  
`pbs_tclsh --version`

### 2.39.2 Description

The `pbs_tclsh` command starts a version of the TCL shell which includes wrapped versions of the PBS external API. The PBS TCL API is documented in "TCL/tk Interface" on page 91 in the PBS Professional Programmer’s Guide.

The `pbs_tclsh` command is used to query MoM. For example:

```
> pbs_tclsh
  tclsh> openrm <hostname>
  <file descriptor>
  tclsh> addreq <file descriptor> "loadave"
  tclsh> getreq <file descriptor>
  <load average>
  tclsh> closereq <file descriptor>
```

#### 2.39.2.1 Required Permission

Root privilege is required in order to query MoM for dynamic resources. Root privilege is not required in order to query MoM for built-in resources and site-defined static resources.

### 2.39.3 Options

```
--version
```

The `pbs_tclsh` command returns its PBS version information and exits. This option can only be used alone.

### 2.39.4 Standard Error

The `pbs_tclsh` command writes a diagnostic message to standard error for each error occurrence.

### 2.39.5 See Also

The PBS Professional Administrator’s Guide, the PBS Programmer’s Guide, "pbs_wish" on page 121
2.40  PBS Commands

2.40.1 Synopsis

```
pbs_tmrsh <hostname> [-l <username>] [-n] <command> [<args> ...]
pbs_tmrsh --version
```

2.40.2 Description

The `pbs_tmrsh` command attempts to emulate an “rsh” connection to the specified host, via underlying calls to the Task Management (TM) API. The program is intended to be used during MPI integration activities, and not by end-users.

Running “`pbs_tmrsh <hostname> <command>`” causes a PBS task to be started on `hostname` running `command`.

2.40.2.1 Requirements for Environment Variables

The environment variables used by the two MPI implementations to point to the `rsh` work-alike (`MPI_REMSH` in the case of HP and `P4_RSHCOMMAND` for MPICH) must be set in the job environment and point to the full path for `pbs_tmrsh`.

The file `$PBS_HOME/pbs_environment` should contain the environment variable `PATH` in which to search for the program executable. This applies to both Windows and Linux. It is expected that a full path will be specified for the `command` and the `PATH` variable will not be needed.

2.40.3 Options

- `-l <username>`
  Specifies the username under which to execute the task. If used, `username` must match the username running the `pbs_tmrsh` command.

- `-n`
  A no-op; provided for MPI implementations that expect to call `rsh` with the “-n” option.

- `--version`
  The `pbs_tmrsh` command returns its PBS version information and exits. This option can only be used alone.

2.40.4 Operands

- `command`
  Specifies command to be run as a PBS task.

- `hostname`
  Specifies host on which to run PBS task. The `hostname` may be specified in IP-dot-address form.

2.40.5 Output and Error

Output and errors are written to the PBS job’s output and error files, not to standard output/error.
The `pbs_tmrsh` command writes a diagnostic message to the PBS job’s error file for each error occurrence.

### 2.40.6 Exit Status

The `pbs_tmrsh` program exits with the exit status of the remote command or with 255 if an error occurred. This is because `ssh` works this way.

### 2.40.7 See Also

2.41  **pbs_topologyinfo**

Reports topological information used for licensing purposes

### 2.41.1  Synopsis

```plaintext
pbs_topologyinfo  (-a | --all)  [(-l | --license) | (-s | --sockets)]
pbs_topologyinfo  (-l | --license)  <vnode name>  [<vnode name> ...]
pbs_topologyinfo  (-s | --sockets)  <vnode name>  [<vnode name> ...]
pbs_topologyinfo  -h | --help
```

### 2.41.2  Description

The `pbs_topologyinfo` command reports topological information for one or more vnodes. This information is used for licensing purposes. To use the command, you must specify what kind of topological information you want. The command reports only the requested information.

This command must be run on the server host.

#### 2.41.2.1  Usage

- `pbs_topologyinfo -a` reports number of node licenses needed for all vnodes.
- `pbs_topologyinfo -l <vnode name>` reports number of node licenses needed for vnode `vnode name`.
- `pbs_topologyinfo -s` reports socket counts for all vnodes that have reported sockets.
- `pbs_topologyinfo -s <vnode name>` reports socket count for vnode `vnode name`.

### 2.41.2.2  Prerequisites

Before you use this command, the server and MoMs must be configured so that they can contact each other, and must have been run.

### 2.41.2.3  Required Privilege for `pbs_topologyinfo`

This command can be run only by root or Admin on Windows.

### 2.41.3  Options for `pbs_topologyinfo`

- `-a, --all`
  Reports requested topological information for all vnodes. When this option is used alone, the command does not report any information.
- `-h, --help`
  Prints usage and exits.
- `-l, --license [vnode name(s)]`
  Reports number of node licenses required. If you specify vnode name(s), the command reports node licenses needed for the specified vnode(s) only.
-s, --sockets [<vnode name(s)>]
   Reports derived socket counts. If you specify vnode name(s), the command reports socket count information for the specified vnode(s) only.
   (no options)
   Does not report any information.

2.41.4 Errors
If you specify an invalid vnode name, the command prints a message to standard error.

2.41.5 Operands
vnode name [<vnode name> ...]
   Name(s) of vnode(s) about which to report.

2.41.6 Exit Status
0
   Success
1
   Any error following successful command line processing

2.41.7 Standard Error
If an invalid vnode name is specified, a message is printed to standard error.

2.41.8 See Also
The PBS Professional Administrator’s Guide
2.42  pbs_wish

TK window shell with TCL-wrapped PBS API

2.42.1  Synopsis

    pbs_wish
    pbs_wish --version

2.42.2  Description

The pbs_wish command is a version of the TK window shell which includes wrapped versions of the PBS external API. The PBS TCL API is documented in "TCL/tk Interface" on page 91 in the PBS Professional Programmer’s Guide.

2.42.3  Options

    --version

    The pbs_wish command returns its PBS version information and exits. This option can only be used alone.

2.42.4  Standard Error

The pbs_wish command writes a diagnostic message to standard error for each error occurrence.

2.42.5  See Also

The PBS Professional Administrator’s Guide, "pbs_tclsh” on page 116
# 2.43 printjob

Prints job information

## 2.43.1 Synopsis

```bash
printjob [ -a | -s ] <job ID>
printjob [ -a ] <file path> [ <file path> ...]
printjob --version
```

## 2.43.2 Description

Prints job information. This command is mainly useful for troubleshooting, as during normal operation, the "qstat" command is the preferred method for displaying job-specific data and attributes. The server and MoM do not have to be running to execute this command.

### 2.43.2.1 Usage

For a running job, you can run this command at any host using a job ID, and you can run this command at any execution host where the job is running using a .JB file path.

For a finished job, if job history is enabled, you can run this command at the server using the job ID.

When querying the server, you must use the job ID, and the data service must be running.

Results will vary depending on whether you use the job ID or a .JB file, and on which execution host you query with a .JB file.

## 2.43.2.2 Permissions

In order to execute `printjob`, you must have root or Windows Administrator privilege.

## 2.43.3 Options to printjob

(no options>

- `a`
  
  Suppresses the printing of job attributes. Cannot be used with `-s` option.

- `s`
  
  Prints out the job script only. Can be used at server or primary execution host. Cannot be used with `-a` option. Must be used with a job ID.

- `--version`
  
  The `printjob` command returns its PBS version information and exits. This option can only be used alone.
2.43.4 Operands for printjob

file path

The printjob command accepts one or more file path operands at the execution host. Files are found in PBS_HOME/mom_priv/jobs/ on the primary execution host. File path must include full path to file. Cannot be used with -s option.

job ID

The printjob command accepts a job ID at the server host. The format is described in "Job Identifier" on page 365. Data service must be running.

2.43.5 Standard Error

The printjob command writes a diagnostic message to standard error for each error occurrence.

2.43.6 Exit Status

Zero

Upon successful processing of all operands presented

Greater than zero

If the printjob command fails to process any operand

2.43.7 See Also

The PBS Professional Administrator's Guide, "qstat" on page 192
2.44  qalter

Alters a PBS job

2.44.1  Synopsis


qalter --version

2.44.2  Description

The qalter command is used to alter one or more PBS batch jobs. Each of certain job attributes can be modified using the qalter option for that attribute. You can alter a job or a job array, but not a subjob or range of subjobs.

2.44.2.1  Required Privilege

A non-privileged user can alter their own jobs, whether they are queued or running. An Operator or Manager can alter any job, whether it is queued or running.

A non-privileged user can only lower resource requests. An Operator or Manager can raise or lower resource requests.

2.44.2.2  Modifying Resources and Job Placement

A Manager or Operator may lower or raise requested resource limits, except for per-process limits such as pcpus and pmem, because these are set when the process starts, and enforced by the kernel. A non-privileged user can only lower resource requests.

The qalter command cannot be used by a non-privileged user to alter a custom resource which has been created to be invisible or read-only for users.

If a job is running, the only resources that can be modified are cput, walltime, min_walltime, and max_walltime.

If a job is queued, any resource mentioned in the options to the qalter command can be modified, but requested modifications must fit within the limits set at the server and queue for the amount of each resource allocated for queued jobs. If a requested modification does not fit within these limits, the modification is rejected.

A job’s resource request must fit within the queue’s and server’s resource run limits. If a modification to a resource exceeds the amount of the resource allowed by the queue or server to be used by running jobs, the job is never run. Requesting resources includes setting limits on resource usage and controlling how the job is placed on vnodes.

See Chapter 5, "List of Built-in Resources", on page 277.

2.44.2.2.i  Syntax for Modifying Resources and Job Placement

Resources are modified by using the -l option, either in chunks inside of selection statements, or in job-wide requests using <resource name>=<value> pairs. The selection statement is of the form:

-l select=\[<N>:\]<chunk>[+\[<N>:\]<chunk> ...]

where N specifies how many of that chunk, and a chunk is of the form:

<resource name>=<value>[<resource name>=<value> ...]
Job-wide \(< resource name >= < value >\) requests are of the form:
-\(l\) \(< resource name >= < value >\)[,\(< resource name >= < value >\) ...]

2.44.2.2.i  The Place Statement

You choose how your chunks are placed using the place statement. The place statement can contain the following elements, in any order:
-\(l\) place=\[< arrangement >]\[: < sharing > ][: < grouping >]

where

\arrangement

Whether this chunk is willing to share this vnode or host with other chunks from the same job. One of free | pack | scatter | vscatter

\sharing

Whether this this chunk is willing to share this vnode or host with other jobs. One of excl | shared | exclhost

\grouping

Whether the chunks from this job should be placed on vnodes that all have the same value for a resource. Can have only one instance of group=\(< resource name >\)

\free

Place job on any vnode(s).

\pack

All chunks are taken from one host.

\scatter

Only one chunk with any MPI processes is taken from a host. A chunk with no MPI processes may be taken from the same vnode as another chunk.

\vscatter

Only one chunk is taken from any vnode. Each chunk must fit on a vnode.

\excl

Only this job uses the vnodes chosen.

\shared

This job can share the vnodes chosen.

\exclhost

The entire host is allocated to the job.

\group=\(< resource name >\)

Chunks are grouped according to a resource. All vnodes in the group must have a common value for resource, which can be either the built-in resource host or a custom vnode-level resource. The resource name must be a string or a string array.

The place statement cannot begin with a colon. Colons are delimiters; use them only to separate parts of a place statement, unless they are quoted inside resource values.

Note that vnodes can have sharing attributes that override job placement requests. See section 6.10, “Vnode Attributes”, on page 333.

For more on resources, resource requests, usage limits, and job placement, see "Using PBS Resources" on page 225 in the PBS Professional Administrator’s Guide and "Allocating Resources & Placing Jobs", on page 49 of the PBS Professional User’s Guide.
2.44.2.3 Modifying Attributes

The user alters job attributes via options to the `qalter` command. Each `qalter` option changes a job attribute. The behavior of the `qalter` command may be affected by any site hooks. Site hooks can modify the job’s attributes, change its routing, etc.

2.44.2.4 Caveats and Restrictions for Altering Jobs

- When you lengthen the `walltime` of a running job, make sure that the new `walltime` will not interfere with any existing reservations etc.
- If any of the modifications to a job fails, none of the job’s attributes is modified.
- A job that is in the process of provisioning cannot be altered.

2.44.3 Options to `qalter`

-a `<date and time>`
Changes the point in time after which the job is eligible for execution. Given in pairs of digits. Sets job’s `Execution_Time` attribute to `date and time`.

Format: `Datetime`

Each portion of the date defaults to the current date, as long as the next-smaller portion is in the future. For example, if today is the 3rd of the month and the specified day `DD` is the 5th, the month `MM` is set to the current month.

If a specified portion has already passed, the next-larger portion is set to one after the current date. For example, if the day `DD` is not specified, but the hour `hh` is specified to be 10:00 a.m. and the current time is 11:00 a.m., the day `DD` is set to tomorrow.

The job’s `Execution_Time` attribute can be altered after the job has begun execution, in which case it will not take effect until the job is rerun.

-A `<account string>`
Replaces the accounting string associated with the job. Used for labeling accounting data. Sets job’s `Account_Name` attribute to `account string`. This attribute cannot be altered once the job has begun execution.

Format: `String`

-c `<checkpoint spec>`
Changes when the job will be checkpointed. Sets job’s `Checkpoint` attribute. An action script is required to checkpoint the job. This attribute can be altered after the job has begun execution, in which case the new value will not take effect until the job is rerun.

The argument `checkpoint spec` can take one of the following values:

- `c`
  Checkpoint at intervals, measured in CPU time, set on job’s execution queue. If no interval set at queue, job is not checkpointed.

- `c=<minutes of CPU time>`
  Checkpoint at intervals of specified number of minutes of job CPU time. This value must be greater than zero. If interval specified is less than that set on job’s execution queue, queue’s interval is used.

  Format: `Integer`

- `w`
  Checkpoint at intervals, measured in walltime, set on job’s execution queue. If no interval set at queue, job is not checkpointed.
 PBS Commands

Chapter 2

w=<minutes of walltime>
  Checkpoint at intervals of the specified number of minutes of job walltime. This value must be greater than zero. If the interval specified is less that that set on the job’s execution queue, the queue's interval is used.
  Format: Integer

n
  No checkpointing.

s
  Checkpoint only when the server is shut down.

u
  Unset. Defaults to behavior when interval argument is set to $s$.
  Default: $u$
  Format: String

-e <error path>
  Replaces the path to be used for the job’s standard error stream. Sets job’s Error_Path attribute to error path. Overridden by -k option.
  Format: [<hostname>]:<path>
  The error path is interpreted as follows:

  path
    If path is relative, it is taken to be relative to the current working directory of the qalter command, where it is executing on the current host.
    If path is absolute, it is taken to be an absolute path on the current host where the qalter command is executing.

  hostname:path
    If path is relative, it is taken to be relative to the user’s home directory on the host named hostname.
    If path is absolute, it is the absolute path on the host named hostname.
    If path does not include a filename, the default filename is <job ID>.ER
  If the -e option is not specified, PBS writes standard error to the default filename, which has this form:
  <job name>.e<sequence number>
  This attribute can be altered after the job has begun execution, in which case the new value will not take effect until the job is rerun.
  If you use a UNC path, the hostname is optional. If you use a non-UNC path, the hostname is required.
-h <hold list>
Updates the job's hold list. Adds *hold list* to the job's *Hold_Types* attribute. The *hold list* is a string of one or more characters. The following table shows the holds and the privilege required to set each:

### Table 2-8: Hold Types

<table>
<thead>
<tr>
<th>Hold Type</th>
<th>Meaning</th>
<th>Who Can Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>User</td>
<td>Job owner, Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>o</td>
<td>Other</td>
<td>Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>s</td>
<td>System</td>
<td>Manager, administrator, root, PBS (dependency)</td>
</tr>
<tr>
<td>n</td>
<td>None</td>
<td>Job owner, Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>p</td>
<td>Bad password</td>
<td>Administrator, root</td>
</tr>
</tbody>
</table>

This attribute can be altered after the job has begun execution, in which case the new value will not take effect until the job is rerun.

-j <join>
Changes whether and how to join the job’s standard error and standard output streams. Sets job’s *Join_Path* attribute to *join*.

This attribute can be altered after the job has begun execution, in which case the new value will not take effect until the job is rerun.

Default: *n*, not merged

The *join* argument can take the following values:

### Table 2-9: Join Path Options

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>oe</td>
<td>Standard error and standard output are merged into standard output.</td>
</tr>
<tr>
<td>eo</td>
<td>Standard error and standard output are merged into standard error.</td>
</tr>
<tr>
<td>n</td>
<td>Standard error and standard output are not merged.</td>
</tr>
</tbody>
</table>
-k <discard>

Changes whether and which of the standard output and standard error streams is left behind on the execution host, and whether they are written to their final destinations. Sets the job’s `Keep_Files` attribute to `discard`. Overrides default path names for these streams. Overrides `-o` and `-e` options.

This attribute cannot be altered once the job has begun execution.

In the case where output and/or error is retained on the execution host in a job-specific staging and execution directory created by PBS, these files are deleted when PBS deletes the directory.

Default: `n`; neither is retained, and files are not written to final destinations

The `discard` argument can take the following values:

**Table 2-10: discard Argument Values**

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>The standard error stream is retained on the execution host, in the job’s staging and execution directory. The filename is <code>&lt;job name&gt;.e&lt;sequence number&gt;</code></td>
</tr>
<tr>
<td>o</td>
<td>The standard output stream is retained on the execution host, in the job’s staging and execution directory. The filename is <code>&lt;job name&gt;.o&lt;sequence number&gt;</code></td>
</tr>
<tr>
<td>eo, oe</td>
<td>Both standard output and standard error streams are retained on the execution host, in the job’s staging and execution directory.</td>
</tr>
<tr>
<td>d</td>
<td>Output and/or error are written directly to their final destination. Overrides the action of leaving files behind on execution host.</td>
</tr>
<tr>
<td>n</td>
<td>Neither stream is retained.</td>
</tr>
</tbody>
</table>

-1 <resource list>

Allows the user to change requested resources and job placement. Sets job’s `Resource_list` attribute to `resource list`. Uses resource request syntax. Requesting a resource places a limit on its usage. Users without manager or operator privilege cannot alter a custom resource which was created to be invisible or read-only for users. For syntax, see section 2.44.2.2.i, “Syntax for Modifying Resources and Job Placement”, on page 124.

If a requested modification to a resource would exceed the server’s or the job queue’s limits, the resource request is rejected. Which resources can be altered is system-dependent.

If the job was submitted with an explicit “-1 select=”, vnode-level resources must be altered using the “-1 select=” form. In this case a vnode-level resource `resource` cannot be altered with the “-1 <resource name>” form.

The place statement cannot begin with a colon.

Examples:

1. Submit the job:
   ```
   % qsub -1 select=1:ncpus=2:mem=512mb jobscript
   Job’s ID is 230
   ```

2. qalter the job using “-1 <resource name>” form:
   ```
   % qalter -1 ncpus=4 230
   Error reported by qalter:
   qalter: Resource must only appear in “select” specification when select is used: ncpus 230
   ```

3. qalter the job using the “-1 select=” form:
   ```
   % qalter -1 select=1:ncpus=4:mem=512mb 230
   ```
No error reported by `qalter`:

```
%
```

For more on resource requests, usage limits and job placement, see "Allocating Resources & Placing Jobs", on page 49 of the PBS Professional User’s Guide.

- `m <mail events>`

Changes the set of conditions under which mail about the job is sent. Sets job’s `Mail_Points` attribute to `mail events`. The `mail events` argument can be one of the following:

- The single character “n”
- Any combination of “a”, “b”, and “e”, with optional “j”

The following table lists the sub-options to the `-m` option:

<table>
<thead>
<tr>
<th>Suboption</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>No mail is sent.</td>
</tr>
<tr>
<td>a</td>
<td>Mail is sent when the job is aborted by PBS.</td>
</tr>
<tr>
<td>b</td>
<td>Mail is sent when the job begins execution.</td>
</tr>
<tr>
<td>e</td>
<td>Mail is sent when the job terminates.</td>
</tr>
<tr>
<td>j</td>
<td>Mail is sent for subjobs. Must be combined with one or more of a, b, or e options</td>
</tr>
</tbody>
</table>

Can be used with job arrays but not subjobs.

Format: `String`

Syntax: `n | [j](one or more of a, b, e)`

Example: `-m ja`

Default value: `a`

- `M <user list>`

Alters list of users to whom mail about the job is sent. Sets job’s `Mail_Users` attribute to `user list`.

Format: `<username>[@<hostname>],<username>[@<hostname>],...`

Default: Job owner.

- `N <name>`

Renames the job. Sets job’s `Job_Name` attribute to `name`.

Format: `Job Name`. See "Job Name, Job Array Name" on page 365.

Default: if a script is used to submit the job, the job’s name is the name of the script. If no script is used, the job’s name is “STDIN”.

- `o <output path>`

Alters path to be used for the job’s standard output stream. Sets job’s `Output_Path` attribute to `output path`. Overridden by `-k` option.

Format: `[@<hostname>]:<path>`

The `output path` is interpreted as follows:

- `path` is relative, it is taken to be relative to the current working directory of the command, where it is executing on the current host.
If path is absolute, it is taken to be an absolute path on the current host where the command is executing.

<hostname>::<path>
If path is relative, it is taken to be relative to the user’s home directory on the host named hostname.
If path is absolute, it is the absolute path on the host named hostname.
If path does not include a filename, the default filename is:

<job ID>.OU

If the -o option is not specified, PBS writes standard output to the default filename, which has this form:

<job name>.o<sequence number>

This attribute can be altered after the job has begun execution, in which case the new value will not take effect until the job is rerun.

If you use a UNC path, the hostname is optional. If you use a non-UNC path, the hostname is required.

-p <priority>
Alters priority of the job. Sets job’s Priority attribute to priority.
This attribute can be altered after the job has begun execution, in which case the new value will not take effect until the job is rerun.
Format: Host-dependent integer
Range: [-1024, +1023] inclusive
Default: zero

-P <project>
Specifies a project for the job. Sets job’s project attribute to specified value.
Format: Project Name; see "Project Name" on page 367
Default: "_pbs_project_default"

-r <y|n>
Changes whether the job is rerunnable. Sets job’s Rerunable attribute to the argument. Does not affect how job is handled when the job is unable to begin execution.
See "qrerun" on page 173.
Format: Single character, “y” or “n”.
y
Job is rerunnable.
n
Job is not rerunnable.
Default: “y”.
Interactive jobs are not rerunnable. Job arrays are always rerunnable.
-R <remove options>
Changes whether standard output and/or standard error files are automatically removed upon job completion.
Sets the job’s Remove_Files attribute to remove options. Overrides default path names for these streams.
Overrides -o and -e options.
This attribute cannot be altered once the job has begun execution.
Default: unset; neither is removed
The remove options argument can take the following values:

Table 2-12: discard Argument Values

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>The standard error stream is removed (deleted) upon job completion</td>
</tr>
<tr>
<td>o</td>
<td>The standard output stream is removed (deleted) upon job completion</td>
</tr>
<tr>
<td>eo, oe</td>
<td>Both standard output and standard error streams are removed (deleted) upon job completion</td>
</tr>
<tr>
<td>unset</td>
<td>Neither stream is removed</td>
</tr>
</tbody>
</table>

-S <path list>
Specifies the interpreter or shell path for the job script. Sets job’s Shell_Path_List attribute to path list.
The path list argument is the full path to the interpreter or shell including the executable name.
Only one path may be specified without a hostname. Only one path may be specified per named host. The path
selected is the one whose hostname is that of the server on which the job resides.
This attribute can be altered after the job has begun execution, but in this case the new value will not take effect
until the job is rerun.
Format:
<path>[@<hostname>][,<path>@<hostname> ...]
If the path contains spaces, it must be quoted. For example:
gsub -S "C:Program Files\PBS Pro\bin\pbs_python.exe" <script name>
Default: user’s login shell on execution node
Example of using bash via a directive:
PBS -S /bin/bash@mars,/usr/bin/bash@jupiter
Example of running a Python script from the command line on Linux:
gsub -S $PBS_EXEC/bin/pbs_python <script name>
Example of running a Python script from the command line on Windows:
gsub -S %PBS_EXEC%\bin\pbs_python.exe <script name>
-u <user list>
Alters list of usernames. Job will run under a username from this list. Sets job’s User_List attribute to user list.
Only one username may be specified without a hostname. Only one username may be specified per named host.
The server on which the job resides will select first the username whose hostname is the same as the server
name. Failing that, the next selection will be the username with no specified hostname. The usernames on the
server and execution hosts must be the same. The job owner must have authorization to run as the specified
user.
This attribute cannot be altered once the job has begun execution.
Format: `<username>@<hostname>[,]<username>@<hostname> ...`

Default: Job owner (username on submit host)

-W `<additional attributes>`

Each sub-option to the `-W` option allows you to change a specific job attribute.

Format: `-W <attribute name> = <attribute value> [, <attribute name>= <attribute value> ...]`

If white space occurs within the `additional attributes` argument, or the equal sign ("=") occurs within an `attribute value` string, that argument or string must be enclosed in single or double quotes. PBS supports setting the following attributes via the `-W` option:

`depend=<dependency list>`

Defines dependencies between this and other jobs. Sets the job’s `depend` attribute to `dependency list`. The `dependency list` has the form:

```plaintext
<type>:<arg list> [, <type>:<arg list> ...]
```

where except for the `on` type, the `<arg list>` is one or more PBS job IDs in the form:

```plaintext
<job ID> [: <job ID> ...]
```

The types and their argument lists can be:

`after: <arg list>`

This job may be scheduled for execution at any point after all jobs in `arg list` have started execution.

`afterok: <arg list>`

This job may be scheduled for execution only after all jobs in `arg list` have terminated with no errors. See section 2.44.6.1, “Warning About Exit Status with csh”, on page 136.

`afternotok: <arg list>`

This job may be scheduled for execution only after all jobs in `arg list` have terminated with errors. See section 2.44.6.1, “Warning About Exit Status with csh”, on page 136.

`afterany: <arg list>`

This job may be scheduled for execution after all jobs in `arg list` have terminated, with or without errors. This job will not run if a job in the `arg list` was deleted without ever having been run.

`before: <arg list>`

Jobs in `arg list` may begin execution once this job has begun execution.

`beforeok: <arg list>`

Jobs in `arg list` may begin execution once this job terminates without errors. See section 2.44.6.1, “Warning About Exit Status with csh”, on page 136.

`beforenotok: <arg list>`

If this job terminates execution with errors, jobs in `arg list` may begin. See section 2.44.6.1, “Warning About Exit Status with csh”, on page 136.

`beforeany: <arg list>`

Jobs in `arg list` may begin execution once this job terminates execution, with or without errors.

`on: <count>`

This job may be scheduled for execution after `count` dependencies on other jobs have been satisfied.

This type is used in conjunction with one of the `before` types listed. `count` is an integer greater than 0.

Restrictions:

Job IDs in the `arg list` of `before` types must have been submitted with a type of `on`.

To use the `before` types, the user must have the authority to alter the jobs in `arg list`. Otherwise, the dependency is rejected and the new job aborted.
Error processing of the existence, state, or condition of the job on which the newly-submitted job depends is performed after the job is queued. If an error is detected, the new job is deleted by the server. Mail is sent to the job submitter stating the error.

Dependency examples:

```
qalter -W depend = afterok:123.host1.domain.com /tmp/script
qalter -W depend= before:234.host1.com:235.host1.com /tmp/script
```

**group_list=<group list>**

Alters list of group names. Job will run under a group name from this list. Sets job’s `group_list` attribute to `group list`.

Only one group name may be specified without a hostname. Only one group name may be specified per named host. The server on which the job resides will select first the group name whose hostname is the same as the server name. Failing that, the next selection is the group name with no specified hostname. The group names on the server and execution hosts must be the same.

Format: `<group>[[@<hostname>]][,<group>@<hostname> ...]`

Default: Login group name of job owner.

**release_nodes_on_stageout=<value>**

When set to `True`, all of the job’s vnodes not on the primary execution host are released when stageout begins.

Cannot be used with vnodes managed by cpuset MoMs, (whose arch is `linux_cpuset`), or with vnodes tied to Cray X® series systems.

When cgroups is enabled and this is used with some but not all vnodes from one MoM, resources on those vnodes that are part of a cgroup are not released until the entire cgroup is released.

The job’s `stageout` attribute must be set for the `release_nodes_on_stageout` attribute to take effect.

Format: `Boolean`

Default: `False`

**run_count=<count>**

Sets the number of times the server thinks it has run the job. Sets the job’s `run_count` attribute to `count`.

Can be altered while job is running. Job is held when the value of this attribute goes over 20.

Format: Integer greater than or equal to zero

**sandbox=<<sandbox spec>**

Changes which directory PBS uses for the job’s staging and execution. Sets job’s `sandbox` attribute to the value of `sandbox spec`.

Format: `String`

Allowed values for `sandbox spec`:

`PRIVATE`

PBS creates a job-specific directory for staging and execution.

`HOME` or unset

PBS uses the user’s home directory for staging and execution.

**stagein=<path list>**

**stageout=<path list>**

Changes files or directories to be staged in before execution or staged out after execution is complete. Sets the job’s `stagein` and `stageout` attributes to the specified `path lists`. On completion of the job, all staged-in and staged-out files and directories are removed from the execution host(s). A `path list` has the form:

`<filespec>[,<filespec>]`
where `filespec` is

\[ <\text{execution path}>@<\text{hostname}>:<\text{storage path}> \]

regardless of the direction of the copy. The `execution path` is the name of the file or directory on the primary execution host. It can be relative to the staging and execution directory on the execution host, or it can be an absolute path.

The “@” character separates `execution path` from `storage path`.

The `storage path` is the path on `hostname`. The name can be relative to the staging and execution directory on the primary execution host, or it can be an absolute path.

If `path list` has more than one `filespec`, i.e. it contains commas, it must be enclosed in double quotes.

If you use a UNC path, the hostname is optional. If you use a non-UNC path, the hostname is required.

`umask=<mask value>`

Alters the umask with which the job is started. Controls umask of job’s standard output and standard error.

Sets job’s `umask` attribute to `mask value`.

Format: one to four digits; typically two

The following example allows group and world read of the job’s output and error:

```
-W umask=33
```

Default: 077

`--version`

The `qalter` command returns its PBS version information and exits. This option can only be used alone.

### 2.44.4 Operands

The `qalter` command accepts a `job ID` list as its operand. The `job ID` list is a space-separated list of one or more job IDs for normal jobs or array jobs.

Subjobs and ranges of subjobs are not alterable.

Job IDs have the form:

\[ <\text{sequence number}>[.<\text{server name}>][@<\text{server name}>] \]

\[ <\text{sequence number}>][[.<\text{server name}>][@<\text{server name}>] \]

Note that some shells require that you enclose a job array ID in double quotes.

### 2.44.5 Standard Error

The `qalter` command writes a diagnostic message to standard error for each error occurrence.

### 2.44.6 Exit Status

- **Zero**
  - Upon successful processing of input

- **Greater than zero**
  - Upon failure
2.44.6.1 Warning About Exit Status with csh

If a job is run in csh and a .logout file exists in the home directory in which the job executes, the exit status of the job is that of the .logout script, not the job script. This may impact any inter-job dependencies.

2.44.7 See Also

2.45  qdel

Deletes PBS jobs

2.45.1  Synopsis

qdel [ -x ] [ -Wforce| -Ws suppress_email=<N> ] <job ID> [<job ID> ...]
qdel --version

2.45.2  Description

The qdel command deletes jobs in the order given, whether they are at the local server or at a remote server.

2.45.2.1  Usage

The qdel command is used without options to delete queued, running, held, or suspended jobs, while the -x option gives it the additional capacity to delete finished or moved jobs. With the -x option, this command can be used on finished and moved jobs, in addition to queued, running, held, or suspended jobs.

When this command is used without the -x option, if job history is enabled, the deleted job's history is retained. The -x option is used to additionally remove the history of the job being deleted.

If someone other than the job's owner deletes the job, mail is sent to the job's owner, or to a list of mail recipients if specified during qsub. See "qsub" on page 207.

If the job is in the process of provisioning, it can be deleted only by using the -W force option.

2.45.2.2  How Behavior of qdel Command Can Be Affected

The server's default_qdel_arguments attribute may affect the behavior of the qdel command. This attribute is settable by the administrator via the qmgr command. The attribute may be set to "-W suppress_email=<N>". The server attribute is overridden by command-line arguments. See section 6.6, “Server Attributes”, on page 295.

2.45.2.3  Sequence of Events

1. The job's running processes are killed.
2. The epilogue runs.
3. Files that were staged in are staged out. This includes standard out (.o) and standard error (.e) files.
4. Files that were staged in or out are deleted.
5. The job's temp directory is removed.
6. The job is removed from the MoM(s) and the server.

2.45.2.4  Required Privilege

A PBS job may be deleted by its owner, an Operator, or the administrator. The server deletes a PBS job by sending a SIGTERM signal, then, if there are remaining processes, a SIGKILL signal.
2.45.3 Options to qdel

(no options)
Can delete queued, running, held, or suspended jobs. Does not delete job history for specified job(s).

-W force
Deletes the job whether or not the job’s execution host is reachable. Deletes the job whether or not the job is in the process of provisioning. Cannot be used with the -Ws suppress_email option.
If the server can contact the MoM, this option is ignored; the server allows the job to be deleted normally. If the server cannot contact the MoM or the job is in the E state, the server deletes its information about the job.

-W suppress_email=<N>
Sets limit on number of emails sent when deleting multiple jobs or subjobs.

• If \( N \geq 1 \) and \( N \) or more job IDs are given, \( N \) emails are sent.
• If \( N \geq 1 \) and less than \( N \) job identifiers are given, the number of emails is the same as the number of jobs.
• If \( N = 0 \), this option is ignored.
• If \( N = -1 \), no mail is sent.
Note that there is no space between “W” and “suppress_email”.
The \( N \) argument is an integer.
Cannot be used with -Wforce option.

-x
Can delete running, queued, suspended, held, finished, or moved jobs. Deletes job history for the specified job(s).

--version
The qdel command returns its PBS version information and exits. This option can only be used alone.

2.45.4 Operands

The qdel command accepts one or more space-separated job ID operands. These operands can be job identifiers, job array identifiers, subjob identifiers, or subjob range identifiers.

Job IDs have the form:
\(<sequence number>[.<server name>][@<server name>]\)

Job arrays have the form:
\(<sequence number>[][][.<server name>][@<server name>]\)

Subjobs have the form:
\(<sequence number>[<index>][.<server name>][@<server name>]\)

Ranges of subjobs have the form:
\(<sequence number>[<first>-<last>][.<server name>][@<server name>]\)

Job array identifiers must be enclosed in double quotes for some shells.

2.45.5 Standard Error

The qdel command writes a diagnostic message to standard error for each error occurrence.
2.45.6 **Exit Status**

Zero
   Upon successful processing of input
Greater than zero
   Upon error

2.45.7 **See Also**

The PBS Professional User's Guide, the PBS Professional Administrator's Guide
Chapter 2  PBS Commands

2.46  qdisable

Prevents a queue from accepting jobs

2.46.1  Synopsis

qdisable <destination> [ <destination> ... ]
qdisable --version

2.46.2  Description

The qdisable command prevents a queue from accepting batch jobs. Sets the value of the queue’s enabled attribute to False. If the command is accepted, the queue no longer accepts Queue Job requests. Jobs already in the queue continue to be processed. You can use this to drain a queue of jobs.

2.46.2.1  Required Permission

In order to execute qdisable, the user must have PBS Operator or Manager privilege.

2.46.3  Options

  --version
  
  The qdisable command returns its PBS version information and exits. This option can only be used alone.

2.46.4  Operands

The qdisable command accepts one or more space-separated destination operands. The operands take any of the following forms:
<queue name>

  Prevents specified queue at default server from accepting jobs.

@<server name>

  Prevents all queues at specified server from accepting jobs.

<queue name>@@<server name>

  Prevents specified queue at specified server from accepting jobs.

To prevent all queues at the default server from accepting jobs, use the qmgr command:

  qmgr: set queue @default enabled=false

2.46.5  Standard Error

The qdisable command writes a diagnostic message to standard error for each error occurrence.
2.46.6 Exit Status

Zero
   Upon successful processing of all the operands
Greater than zero
   If the qdisable command fails to process any operand

2.46.7 See Also

The PBS Professional Administrator’s Guide, "qmgr" on page 146, "qenable" on page 142
2.47  qenable

Allows a queue to accept jobs

2.47.1  Synopsis

qenable <destination> [ <destination> ...]
qenable --version

2.47.2  Description

The qenable command allows a queue to accept batch jobs. Sets the value of the queue’s enabled attribute to True. If the command is accepted, the destination accepts Queue Job requests.

2.47.2.1  Required Privilege

In order to execute qenable, the user must have PBS Operator or Manager privilege.

2.47.3  Options

--version

The qenable command returns its PBS version information and exits. This option can only be used alone.

2.47.4  Operands

The qenable command accepts one or more space-separated destination operands. The operands take any of the following forms:

<queue name>

Allows specified queue at default server to accept jobs.

@<server name>

Allows all queues at specified server to accept jobs.

<queue name>@<server name>

Allows specified queue at specified server to accept jobs.

To allow all queues at the default server to accept jobs, use the qmgr command:

qmgr: set queue @default enabled=true

2.47.5  Standard Error

The qenable command writes a diagnostic message to standard error for each error occurrence.

2.47.6  Exit Status

Zero

Upon successful processing of all the operands
Greater than zero
  If the qenable command fails to process any operand

2.47.7  See Also

The PBS Professional Administrator’s Guide, "qmgr" on page 146, "qdisable" on page 140
2.48 qhold

Holds PBS batch jobs

2.48.1 Synopsis

qhold [-h <hold list>] <job ID> [<job ID> ...]
qhold --version

2.48.2 Description

Places one or more holds on a job. A job that has a hold is not eligible for execution. Can be used on jobs and job arrays, but not on subjobs or ranges of subjobs.

If a job identified by job ID is in the queued, held, or waiting states, all that occurs is that the hold type is added to the job. The job is then put into the held state if it resides in an execution queue.

If the job is running, the result of the qhold command depends upon whether the job can be checkpointed. The job can be checkpointed if the OS supports checkpointing, or if the application being checkpointed supports checkpointing. See the PBS Professional Administrator’s Guide. If the job can be checkpointed, the following happens:

• The job is checkpointed and its execution is interrupted.
• The resources assigned to the job are released.
• The job is placed in the held state in the execution queue.
• The job’s Hold_Types attribute is set to u for user hold.

If checkpoint / restart is not supported, qhold simply sets the job’s Hold_Types attribute to u. The job continues to execute.

A job’s dependency places a system hold on the job. When the dependency is satisfied, the system hold is removed. This system hold is the same as the one set by an administrator. If the administrator sets a system hold on a job with a dependency, when the dependency is satisfied, the job becomes eligible for execution.

If the job is in the process of provisioning, it cannot be held.

A hold on a job can be released by the administrator, a Manager, an Operator, or the job owner, when the job reaches the time set in its Execution_Time attribute, or when a dependency clears. See “qrls” on page 175.

2.48.2.1 Effect of Privilege on Behavior

The following table shows the holds and the privilege required to set each:

<table>
<thead>
<tr>
<th>Hold Type</th>
<th>Meaning</th>
<th>Who Can Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>User</td>
<td>Job owner, Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>o</td>
<td>Other</td>
<td>Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>s</td>
<td>System</td>
<td>Manager, administrator, root, PBS (dependency)</td>
</tr>
<tr>
<td>n</td>
<td>None</td>
<td>Job owner, Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>p</td>
<td>Bad password</td>
<td>Administrator, root</td>
</tr>
</tbody>
</table>
2.48.3 Options to \texttt{qhold}

(no options)
Same as \texttt{-h u}. Applies the \textit{user} hold to the specified job(s).

-\texttt{h <hold list>}
Types of holds to be placed on the job(s).
The \textit{hold list} argument is a string consisting of one or more of the letters "u", "o", or "s" in any combination, or one of the letters "n" or "p".

--\texttt{version}
The \texttt{qhold} command returns its PBS version information and exits. This option can only be used alone.

2.48.4 Operands

The \texttt{qhold} command can be used on jobs and job arrays, but not on subjobs or ranges of subjobs. The \texttt{qhold} command accepts one or more job IDs in the form:

<sequence number>[.<server name>][@<server name>]
<sequence number>[].<server name>[@<server name>]

Note that some shells require that you enclose a job array identifier in double quotes.

2.48.5 Standard Error

The \texttt{qhold} command writes a diagnostic message to standard error for each error occurrence.

2.48.6 Exit Status

Zero
Upon successful processing of all operands

Greater than zero
If the \texttt{qhold} command fails to process any operand

2.48.7 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide, "qrls" on page 175
2.49 qmgr

Administrator’s command interface for managing PBS

2.49.1 Synopsis

At shell command line:

qmgr -c '<directive> [-a] [-e] [-n] [-z]'
qmgr -c 'help [<help option>]'
qmgr <return>
qmgr --version

In qmgr session:

<directive> [-a] [-e] [-n] [-z]
help <help option>

2.49.2 Description

The PBS manager command, qmgr, provides a command-line interface to parts of PBS. The qmgr command is used to create or delete queues, vnodes, resources, and hooks, to set or change vnode, queue, hook, server, or scheduler attributes and resources, and to view information about hooks, queues, vnodes, resource definitions, the server, and schedulers.

For a list of quick summaries of information about syntax, commands, attributes, operators, names, and values, type “help” or “?” at the qmgr prompt. See section 2.49.1, “Printing Usage Information”, on page 165.

2.49.2.1 Modes of Operation

• When you type qmgr -c '<directive>', qmgr performs its task and then exits.
• When you type qmgr <return>, qmgr starts a session and presents you with its command line prompt. The qmgr command then reads directives etc. from standard input; see section 2.49.4.1, “Directive Syntax”, on page 148. You can edit the command line; see section 2.49.2.4, “Reusing and Editing the qmgr Command Line”, on page 147.

For a qmgr prompt, type:

qmgr <return>

You will see the qmgr prompt:

Qmgr:

2.49.2.2 Required Privilege

The qmgr command requires different levels of privilege depending on the operation to be performed.

All users can list or print attributes except for hook attributes.

PBS Operator or Manager privilege is required in order to set or change vnode, queue, server, or scheduler attributes. PBS Manager privilege is required in order to create or delete queues, vnodes, and resources.

Under Linux, root privilege is required in order to create hooks, or operate on hooks or the job_sort_formula server attribute. Under Windows, this must be done from the installation account.
For domained environments, the installation account must be a local account that is a member of the local Administrators group on the local computer. For standalone environments, the installation account must be a local account that is a member of the local Administrators group on the local computer.

Users without manager or operator privilege cannot view custom resources or resource definitions which were created to be invisible to users.

### 2.49.2.3 When To Run `qmgr` At Server Host

When operating on hooks or on the `job_sort_formula` server attribute, the `qmgr` command must be run at the server host.

### 2.49.2.4 Reusing and Editing the `qmgr` Command Line

You can reuse or edit `qmgr` command lines. The `qmgr` command maintains a history of commands entered, up to a maximum of 500. You can use the `history` command to see a numbered list of commands, and the `!n` command to execute the line whose number is `n`. You must not put any spaces between the bang (`"!"`) and the number. For example, to execute the 123rd command, type the following:

```
!123
```

You can see the last `m` commands by typing `history m`. For example, to see the last 6 commands, type the following:

```
history 6
```

You can use the up and down arrows to navigate through the command history list, and the left and right arrows to navigate within a command line. Within a command line, you can use `emacs` commands to move forward and backward, and delete characters.

You can edit the `qmgr` command line using the backspace and delete keys, and you can insert characters anywhere in a command line.

History is maintained across `qmgr` sessions, so that if you start `qmgr`, then exit, then restart it, you can reuse your commands from the previous session. If you exit `qmgr` and then restart it, the command lines are renumbered.

If you enter the same command line more than once in a row, only one occurrence is recorded in the history. If you enter the same command line multiple times, but intersperse other command lines after each line, each occurrence is recorded.

Each user’s history is unique to that user on that host.

In the case where an account runs concurrent sessions, the most recent logout of a session overwrites history from previous logouts. For example, if two people are both logged in as root and using `qmgr`, the second person to log out overwrites the history file.

#### 2.49.2.4.i The `qmgr` History File

The `qmgr` command stores and retrieves its history. First, it tries to write its history in the `${HOME}/.pbs_qmgr_history` file. If this file or directory location is not writable, the command stores its history in `${PBS_HOME}/spool/.pbs_qmgr_history_<user name>`. If this file is also not writable, the following happens:

- The `qmgr` command prints error messages once at `qmgr` startup
- The `qmgr` command cannot provide history across `qmgr` sessions
2.49.3 Options to qmgr

The following table lists the options to qmgr:

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;return&gt;</td>
<td>Starts a qmgr session and presents user with qmgr prompt</td>
</tr>
<tr>
<td>-a</td>
<td>Aborts qmgr on any syntax errors or any requests rejected by a server.</td>
</tr>
<tr>
<td>-c '&lt;directive&gt;'</td>
<td>Executes a single command (directive) and exit qmgr. The directive must be enclosed in single or double quote marks, for example: qmgr -c &quot;print server&quot;</td>
</tr>
<tr>
<td>-c 'help [&lt;help option&gt;]'</td>
<td>Prints out usage information. See &quot;Printing Usage Information&quot; on page 165</td>
</tr>
<tr>
<td>-e</td>
<td>Echoes all commands to standard output</td>
</tr>
<tr>
<td>-n</td>
<td>No commands are executed; syntax checking only is performed</td>
</tr>
<tr>
<td>-z</td>
<td>No errors are written to standard error</td>
</tr>
<tr>
<td>--version</td>
<td>The qmgr command returns its PBS version information and exits. This option can only be used alone</td>
</tr>
</tbody>
</table>

2.49.4 Directives

A qmgr directive is a command together with the object(s) to be operated on, the attribute(s) belonging to the object that is to be changed, the operator, and the value(s) the attribute(s) will take. In the case of resources, you can set the type and/or flag(s).

2.49.4.1 Directive Syntax

A directive is terminated by a newline or a semicolon (“;”). Multiple directives may be entered on a single line. A directive may extend across lines by escaping the newline with a backslash (“\”).

Comments begin with the “#” character and continue to the end of the line. Comments and blank lines are ignored by qmgr.

2.49.4.1.i Server, Scheduler, Queue, Vnode Directives

Syntax for operating on servers, schedulers, queues, and vnodes:

<command> <object type> [<object name(s)>] [<attribute> <operator> <value> [, <attribute> <operator> <value> ...]]

For information about attributes, see Chapter 6, "Attributes", on page 291.

2.49.4.1.ii Resource Directives

Syntax for operating on resources:

<command> <resource name> [<resource name> ...] [type = <type>, [flag = <flag(s)>]]

For information about resources, see "Using PBS Resources" on page 225 in the PBS Professional Administrator’s Guide and Chapter 5, "List of Built-in Resources", on page 277.
2.49.4.1.iii Hook Directives

Syntax for operating on site-defined hooks:

*import hook <hook name> application/x-python <content-encoding> (<input file> | |-)*
*import hook <hook name> application/x-config <content-encoding> (<input file> | |-)*
*export hook <hook name> <content-type> <content-encoding>* > [<output file>]

Syntax for operating on built-in hooks:

*import pbshook <hook name> application/x-config <content-encoding> (<input file> | -)*
*export pbshook <hook name> <content-type> <content-encoding>* > [<output file>]

2.49.4.2 Using Directives

You can use a directive from the shell command line or from within the qmgr session.

• To use a directive from the command line, enclose the command and its arguments in single or double quotes.
  
  `qmgr -c 'command <command arguments>'`

  For example, to have qmgr print server information and exit:

  `qmgr -c "print server"

• To use a directive from within the qmgr session, first start qmgr:

  `qmgr <return>`

  The qmgr session presents a qmgr prompt:

  `Qmgr:`

  At the qmgr prompt, enter the directive (a command and its arguments). For example, to enter the same "print server" directive:

  `Qmgr: print server`

2.49.4.3 Commands Used in Directives

Commands can be abbreviated to their minimum unambiguous form. Commands apply to all target objects unless explicitly limited. The following table lists the commands, briefly tells what they do, and gives a link to a full description:

<table>
<thead>
<tr>
<th>Command</th>
<th>Abbr.</th>
<th>Effect</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>a</td>
<td>Specifies active objects</td>
<td>See section 2.49.6.1, “Making Objects Active”, on page 153</td>
</tr>
<tr>
<td>create</td>
<td>c</td>
<td>Creates object</td>
<td>See section 2.49.6.2, “Creating Objects (Server, Scheduler, Vnode, Queue, Hook)”, on page 154</td>
</tr>
<tr>
<td>delete</td>
<td>d</td>
<td>Deletes object</td>
<td>See section 2.49.6.3, “Deleting Objects”, on page 154</td>
</tr>
<tr>
<td>export</td>
<td>e</td>
<td>Exports hook or hook configuration file</td>
<td>See section 2.49.10.6, “Exporting Hooks”, on page 164 and section 2.49.10.5.ii, “Exporting Configuration Files”, on page 163</td>
</tr>
<tr>
<td>exit</td>
<td></td>
<td>Exits (quits) the qmgr session</td>
<td></td>
</tr>
<tr>
<td>help or ?</td>
<td>h, ?</td>
<td>Prints usage to stdout</td>
<td>See section 2.49.11, “Printing Usage Information”, on page 165</td>
</tr>
</tbody>
</table>
Chapter 2

PBS Commands

### 2.49.5 Arguments to Directive Commands

#### 2.49.5.1 Object Arguments to Directive Commands

The `qmgr` command can operate on objects (servers, schedulers, queues, vnodes, resources, hooks, and built-in hooks). Each of these can be abbreviated inside a directive. The following table lists the objects and their abbreviations:

<table>
<thead>
<tr>
<th>Object Name</th>
<th>Abbr.</th>
<th>Object</th>
<th>Can Be Created/Deleted By:</th>
<th>Can Be Modified By:</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>s</td>
<td>server</td>
<td>No one (created at installation)</td>
<td>Administrator, Operator, Manager</td>
</tr>
<tr>
<td>sched</td>
<td>sc</td>
<td>default scheduler</td>
<td>No one (created at installation)</td>
<td>Administrator, Operator, Manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>multisched</td>
<td>Administrator, Manager</td>
<td>Administrator, Operator, Manager</td>
</tr>
<tr>
<td>queue</td>
<td>q</td>
<td>queue</td>
<td>Administrator, Operator, Manager</td>
<td>Administrator, Operator, Manager</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
<td>vnode</td>
<td>Administrator, Operator, Manager</td>
<td>Administrator, Operator, Manager</td>
</tr>
<tr>
<td>resource</td>
<td>r</td>
<td>resource</td>
<td>Administrator, Manager</td>
<td>Administrator, Manager</td>
</tr>
<tr>
<td>hook</td>
<td>h</td>
<td>hook</td>
<td>Linux: root&lt;br&gt;Windows: installation account</td>
<td>Linux: root&lt;br&gt;Windows: installation account</td>
</tr>
<tr>
<td>pbshook</td>
<td>p</td>
<td>built-in hook</td>
<td>No one (created at installation)</td>
<td>Linux: root&lt;br&gt;Windows: installation account</td>
</tr>
</tbody>
</table>
2.49.5.1.i Specifying Active Server

The `qmgr` command operates on objects (queues, vnodes, etc.) at the active server. There is always at least one active server; the default server is the active server unless other servers have been made active. The default server is the server managing the host where the `qmgr` command runs, meaning it is the server specified in that host’s `pbs.conf` file. Server names have the following format:

\(<hostname>[:<port number>]\)

where `hostname` is the fully-qualified domain name of the host on which the server is running and `port number` is the port number to which to connect. If `port number` is not specified, the default port number, 15001, is used.

- To specify the default server:
  \(@\text{default}\\n\)

- To specify a named server:
  \(@<\text{server name}>\\n\)

- To specify all active servers:
  \(@\text{active}\\n\)

2.49.5.1.ii Using Lists of Object Names

In a `qmgr` directive, `object name(s)` is a list of one or more names of specific objects. The administrator specifies the name of an object when creating the object. The name list is in the form:

\(<object name>[@<server>] [,<object name>[@<server>] ...]\)

where `server` is replaced in the directive with “\text{default}”, “\text{active}”, or the name of the server. The name list must conform to the following:

- There must be no space between the object name and the `@` sign.
- Name lists must not contain white space between entries.
- All objects in a list must be of the same type.
- Node attributes cannot be used as vnode names.

2.49.5.1.iii Specifying Object Type and Name

You can specify objects in the following ways:

- To act on the active objects of the named type, at the active server:
  \(<\text{object type}>\\n\)
  For example, to list all active vnodes, along with their attributes, at the active server:

  **Qmgr:** list node

- To act on the active objects of the named type, at a specified server:
  \(<\text{object type}> @<\text{server name}>\) (note space before `@` sign)
  For example, to list all active vnodes at the default server, along with their attributes:

  **Qmgr:** list node @default

  For example, to print out all queues at the default server, along with their attributes:

  **Qmgr:** list queue @default

- To act on a specific named object:
  \(<\text{object type}> <\text{object name}>\)
  For example, to list Node1 and its attributes:

  **Qmgr:** list node Node1
To list queues workq, slowq, and fastq at the active server:

```
Qmgr: list queue workq,slowq,fastq
```

• To act on the named object at the specified server:

```
<object type> <object name>@<server name>
```

For example, to list Node1 at the default server, along with the attributes of Node1:

```
Qmgr: list node Node1@default
```

To list queues Queue1 at the default server, Queue2 at Server2, and Queue3 at the active server:

```
Qmgr: list queue Queue1@default,Queue2@Server2,Queue3@active
```

### 2.49.5.2 Operators in Directive Commands

In a `qmgr` directive, `operator` is the operation to be performed with the attribute and its value. Operators are listed here:

**Table 2-17: Operators in Directive Commands**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Sets the value of the attribute or resource. If the attribute or resource has an existing value, the current value is replaced with the new value.</td>
</tr>
<tr>
<td>+=</td>
<td>Increases the current value of the attribute or resource by the amount in the new value. When used for a string array, adds the new value as another string after a comma.</td>
</tr>
<tr>
<td>-=</td>
<td>Decreases the current value of the attribute or resource by the specified amount. When used for a string array, removes the first matching string.</td>
</tr>
</tbody>
</table>

Example 2-4: Set routing destination for queue Queue1 to be Dest1:

```
Qmgr: set queue route_destinations = Dest1
```

Example 2-5: Add new routing destination for queue Queue1:

```
Qmgr: set queue route_destinations += Dest2
```

Example 2-6: Remove new routing destination for queue Queue1:

```
Qmgr: set queue route_destinations -= Dest2
```

When setting numerical resource values, you can use only the equal sign (“=”).

### 2.49.5.3 Windows Requirements For Directive Arguments

Under Windows, use double quotes when specifying arguments to `qmgr`. For example:

```
Qmgr: import hook hook1 application/x-python default "\Documents and Settings\pbsuser1\hook1.py"
```

or

```
qumgr -c 'import hook hook1 application/x-python default "\Documents and Settings\pbsuser1\hook1.py"'
```
2.49.6 Operating on Objects (Server, Scheduler, Vnode, Queue, Hook)

2.49.6.1 Making Objects Active

Making objects active is a way to set up a list of objects, all of the same type, on which you can then use a single command. For example, if you are going to set the same attribute to the same value on several vnodes, you can make all of the target vnodes active before using a single command to set the attribute value, instead of having to give the command once for each vnode. You can make any type of object active except for resources or hooks.

When an object is active, it is acted upon when you specify its type but do not specify names. When you specify any object names in a directive, active objects are not operated on unless they are named in the directive.

You can specify a list of active objects for each type of object. You can have active objects of multiple types at the same time. The active objects of one type have no effect on whether objects of another type are active.

Objects are active only until the `qmgr` command is exited, so this feature can be used only at the `qmgr` prompt.

Each time you make any objects active at a given server, that list of objects replaces any active objects of the same kind at that server. For example, if you have four queues at a particular server, and you make Q1 and Q2 active, then later make Q3 and Q4 active, the result is that Q3 and Q4 are the only active queues.

You can make different objects be active at different servers simultaneously. For example, you can set vnodes N1 and N2 at the default server, and vnodes N3 and N4 at server Server2 to be active at the same time.

To make all objects inactive, quit `qmgr`. When you quit `qmgr`, any object that was active is no longer active.

2.49.6.1.i Using the `active` Command

- To make the named object(s) of the specified type active:

  ```active <object type> [<object name>,<object name> ...]```

  Example: To make queue Queue1 active:

  ```Qmgr: active queue Queue1```

  Example: To make queues Queue1 and Queue2 at the active server be active, then enable them:

  ```Qmgr: active queue Queue1,Queue2```
  ```Qmgr: set queue enabled=True```

  Example: To make queue Queue1 at the default server and queue Queue2 at Server2 be active:

  ```Qmgr: active queue Queue1@default,Queue2@Server2```

  Example: To make vnodes N1, N2, N3, and N4 active, and then give them all the same value for their `max_running` attribute:

  ```Qmgr: active node N1,N2,N3,N4```
  ```Qmgr: set node max_running = 2```

- To make all object(s) of the specified type at the specified server active:

  ```active <object type> @<server name>```  (note space before @ sign)

  Example: To make all queues at the default server active:

  ```Qmgr: active queue @default```

  Example: To make all vnodes at server Server2 active:

  ```Qmgr: active node @Server2```

- To report which objects of the specified type are active:

  ```active <object type>```

  The `qmgr` command prints a list of names of active objects of the specified type to `stdout`.  


2.49.6.2 Creating Objects (Server, Scheduler, Vnode, Queue, Hook)

- To create one new object of the specified type for each name, and give it the specified name:

  \[ \text{create } \langle \text{object type} \rangle \ [\langle \text{object name} \rangle \ [\langle \text{object name} \rangle \ldots] \ [\langle \text{attribute} \rangle = \langle \text{value} \rangle \ [\langle \text{attribute} \rangle = \langle \text{value} \rangle \ldots] \]

  Can be used only with multischeds, queues, vnodes, resources, and hooks. Cannot be used with built-in hooks.

  For example, to create a multisched named multisched_1 at the active server:
  ```bash
  Qmgr: create sched multisched_1
  ```

  For example, to create a queue named Q1 at the active server:
  ```bash
  Qmgr: create queue Q1
  ```

  For example, to create a vnode named N1 and a vnode named N2:
  ```bash
  Qmgr: create node N1,N2
  ```

  For example, to create queue Queue1 at the default server and queue Queue2 at Server2:
  ```bash
  Qmgr: create queue Queue1@default,Queue2@Server2
  ```

  For example, to create vnodes named N1, N2, N3, and N4 at the active server, and to set their `Mom` attribute to `Host1` and their `max_running` attribute to 1:
  ```bash
  Qmgr: create node N1,N2,N3,N4 Mom=Host1, max_running = 1
  ```

  To create a host-level consumable string resource named “foo”:
  ```bash
  Qmgr: qmgr -c "create resource foo type=string,flag=nh"
  ```

  All objects of the same type at a server must have unique names. For example, each queue at server Server1 must have a unique name. Objects at one server can have the same name as objects at another server.

  You can create multiple objects of the same type with a single command. You cannot create multiple types of objects in a single command.

  To create multiple resources of the same type and flag, separate each resource name with a comma:
  ```bash
  qmgr -c "create resource <resource>[,<resource> ...] type=<type>,flag=<flag(s)>"
  ```

2.49.6.2.i Examples of Creating Objects

Example 2-7: Create queue:
  ```bash
  create queue fast priority=10,queue_type=e,enabled = true,max_running=0
  ```

Example 2-8: Create queue, set resources:
  ```bash
  create queue little
  set queue little resources_max.mem=8mw,resources_max.cput=10
  ```

2.49.6.3 Deleting Objects

- To delete the named object(s):

  \[ \text{delete } \langle \text{object type} \rangle \ [\langle \text{object name} \rangle \ [\langle \text{object name} \rangle \ldots] \]

  When you delete more than one object, do not put a space after a comma.

  Can be used only with queues, vnodes, resources, and hooks. Cannot be used with built-in hooks.

  For example, to delete queue Q1 at the active server:
  ```bash
  Qmgr: delete queue Q1
  ```

  For example, to delete vnodes N1 and N2 at the active server:
  ```bash
  Qmgr: delete node N1,N2
  ```
For example, to delete queue Queue1 at the default server and queue Queue2 at Server2:

```
Qmgr: delete queue Queue1@default,Queue2@Server2
```

For example, to delete resource “foo” at the active server:

```
Qmgr: delete resource foo
```

- To delete the active objects of the specified type:
  
  ```
delete <object type>
  ```

  For example, to delete the active queues:

  ```
Qmgr: delete queue
  ```

- To delete the active objects of the specified type at the specified server:
  
  ```
delete <object type> @<server name>
  ```

  For example, to delete the active queues at server Server2:

  ```
Qmgr: delete queue @Server2
  ```

You can delete multiple objects of the same type with a single command. You cannot delete multiple types of objects in a single command. To delete multiple resources, separate the resource names with commas.

For example:

```
Qmgr: delete resource r1,r2
```

You cannot delete a resource that is requested by a job or reservation, or that is set on a server, queue, or vnode.

## 2.49.7 Operating on Attributes and Resources

You can specify attributes and resources for named objects or for all objects of a type.

### 2.49.7.1 Setting Attribute and Resource Values

- To set the value of the specified attribute(s) for the named object(s):
  
  ```
set <object type> <object name> [... <object name> ...] <attribute> = <value> [,<attribute> = <value> ...]
  ```

  Each specified attribute is set for each named object, so if you specify three attributes and two objects, both objects get all three attributes set.

- To set the attribute value for all active objects when there are active objects of the type specified:
  
  ```
set <object type> <attribute> = <value>
  ```

- To set the attribute value for all active objects at the specified server when there are active objects of the type specified:
  
  ```
set <object type> @<server name> <attribute> = <value>
  ```

  For example, to set the amount of memory on a vnode:

  ```
Qmgr: set node Vnode1 resources_available.mem = 2mb
  ```

If the attribute is one which describes a set of resources such as `resources_available`, `resources_default`, `resources_max`, `resources_used`, etc., the attribute is specified in the form:

```
<attribute name>.<resource name>
```

You can have spaces between `attribute=value` pairs.
2.49.7.1.i  Examples of Setting Attribute Values

Example 2-9: Increase limit on queue:

```
set queue fast max_running +=2
```

Example 2-10: Set software resource on mynode:

```
set node mynode resources_available.software = "myapp=/tmp/foo"
```

Example 2-11: Set limit on queue:

```
set queue max_running = 10
```

Example 2-12: Set vnode offline:

```
set node state = "offline"
```

2.49.7.2  Unsetting Attribute and Resource Values

You can use the `qmgr` command to unset attributes of any object, except for the type attribute of a built-in hook.

- To unset the value of the specified attributes of the named object(s):
  
  ```
  unset <object type> <object name>[, <object name> ...] <attribute>[, <attribute> ...]
  ```

- To unset the value of specified attributes of active objects:
  
  ```
  unset <object type> <attribute>[, <attribute> ...]
  ```

- To unset the value of specified attributes of the named object:
  
  ```
  unset <object type> <object name> <attribute>[, <attribute> ...]
  ```

- To unset the value of specified attributes of the named object:
  
  ```
  unset <object type> @<server name> <attribute>[, <attribute> ...]
  ```

2.49.7.2.i  Example of Unsetting Attribute Value

Example 2-13: Unset limit on queue

```
unset queue fast max_running
```

2.49.7.3  Caveats and Restrictions for Setting Attribute and Resource Values

- If the value includes whitespace, commas or other special characters, such as the # character, the value string must be enclosed in single or double quotes. For example:

  ```
  Qmgr: set node Vnode1 comment="Node will be taken offline Friday at 1:00 for memory upgrade."
  ```

- You can set or unset attribute values for only one type of object in each command.

- You can use the `qmgr` command to set attributes of any object, except for the type attribute of a built-in hook.

- You can have spaces between attribute names.

- Attribute and resource values must conform to the format for the attribute or resource type. Each attribute’s type is listed in Chapter 6, "Attributes", on page 291. Each format is described in Chapter 7, "Formats", on page 363.

- Most of a vnode’s attributes may be set using `qmgr`. However, some must be set on the individual execution host in local vnode definition files, NOT by using `qmgr`. See "Choosing Configuration Method" on page 42 in the PBS Professional Administrator’s Guide.

2.49.7.4  Setting Resource Type and Flag(s)

You can use the `qmgr` command to set or unset the type and flag(s) for resources.
Resource types can be the following; see Chapter 7, "Formats", on page 363:

- string
- boolean
- string_array
- long
- size
- float

To set a resource type:

\[ \text{set resource } <\text{resource name}> \text{ type } = <\text{type}> \]

Sets the type of the named resource to the specified type. For example:

\[ \text{Qmgr: qmgr -c } \text{"set resource foo type=string_array"} \]

### 2.49.7.4.i Resource Accumulation Flags

The resource accumulation flag for a resource can be one of the following:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(no flags)</td>
<td>Indicates a queue-level or server-level resource that is not consumable.</td>
</tr>
<tr>
<td>( fh )</td>
<td>The amount is consumable at the host level for only the first vnode allocated to the job (vnode with first task.) Must be consumable or time-based. Cannot be used with Boolean or string resources. This flag specifies that the resource is accumulated at the first vnode, meaning that the value of ( \text{resources Assigned.&lt;resource&gt; } ) is incremented only at the first vnode when a job is allocated this resource or when a reservation requesting this resource on this vnode starts.</td>
</tr>
</tbody>
</table>
Table 2-18: Resource Accumulation Flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>Indicates a host-level resource. Used alone, means that the resource is not consumable. Required for any resource that will be used inside a select statement. This flag selects hardware. This flag indicates that the resource must be requested inside of a select statement. Example: for a Boolean resource named &quot;green&quot;:&lt;br&gt;green type=boolean, flag=h</td>
</tr>
<tr>
<td>nh</td>
<td>The amount is consumable at the host level, for all vnodes assigned to the job. Must be consumable or time-based. Cannot be used with Boolean or string resources. This flag specifies that the resource is accumulated at the vnode level, meaning that the value of resources_assigned.&lt;resource&gt; is incremented at relevant vnodes when a job is allocated this resource or when a reservation requesting this resource on this vnode starts. This flag is not used with dynamic consumable resources. A scheduler will not oversubscribe dynamic consumable resources.</td>
</tr>
<tr>
<td>q</td>
<td>The amount is consumable at the queue and server level. When a job is assigned one unit of a resource with this flag, the resources_assigned.&lt;resource&gt; attribute at the server and any queue is incremented by one. Must be consumable or time-based. This flag specifies that the resource is accumulated at the queue and server level, meaning that the value of resources_assigned.&lt;resource&gt; is incremented at each queue and at the server when a job is allocated this resource. When a reservation starts, allocated resources are added to the server’s resources_assigned attribute. This flag is not used with dynamic consumable resources. A scheduler will not oversubscribe dynamic consumable resources.</td>
</tr>
</tbody>
</table>

See "Resource Accumulation Flags" on page 253 in the PBS Professional Administrator’s Guide.

2.49.7.4.ii Resource Permission Flags

The permission flag for a resource can be one of the following:

Table 2-19: Resource Permission Flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(no flag)</td>
<td>Users can view and request the resource, and qalter a resource request for this resource.</td>
</tr>
<tr>
<td>i</td>
<td>“Invisible”. Users cannot view or request the resource. Users cannot qalter a resource request for this resource.</td>
</tr>
<tr>
<td>r</td>
<td>“Read only”. Users can view the resource, but cannot request it or qalter a resource request for this resource.</td>
</tr>
</tbody>
</table>


To set resource flags, concatenate the flags you want without spaces or commas.

- To set the flag(s) of the named resource to the specified flag(s):
  
  set resource <resource name> flag = <flag(s)>
For example:

```
qmgr -c "set resource foo flag=nhi"
```

- To set both type and flag(s):

```
set resource <resource name> type=<type>, flag = <flag(s)>
```

Sets the type and flag(s) of the named resource to the specified type and flag(s). For example:

```
qmgr -c "set resource foo type=long,flag=nhi"
```

You can set multiple resources by separating the names with commas. For example:

```
qmgr -c "set resource r1,r2 type=long"
```

You cannot set the type for a resource that is requested by a job or reservation, or set on a server, queue, or vnode.

You cannot set the flag(s) to h, nh, fh, or q for a resource that is requested by a job or reservation.

### 2.49.7.5 Unsetting Resource Flag(s)

You can use the `qmgr` command to unset the flag(s) for resources.

- To unset the flag(s) of the named resource:

```
unset resource <resource name> flag
```

For example:

```
qmgr -c "unset resource foo flag"
```

You can unset the flag(s) of multiple resources by separating the resource names with commas. For example:

```
qmgr -c "unset resource r1,r2 flag"
```

You cannot unset the type for a resource.

You cannot unset the flag(s) for a resource that is requested by a job or reservation, or set on any server, queue, or vnode.

### 2.49.8 Viewing Object, Attribute, and Resource Information

#### 2.49.8.1 Listing Objects and Their Attributes

You can use the `qmgr` command to list attributes of any object, including attributes at their default values.

- To list the attributes, with associated values, of the named object(s):

```
list <object type> <object name>[,<object name> ...]
```

- To list values of the specified attributes of the named object:

```
list <object type> <object name> <attribute name>[, <attribute name>]...
```

- To list attributes, with associated values, of active objects of the specified type at the active server:

```
list <object type>
```

- To list all objects of the specified type at the specified server, with their attributes and the values associated with the attributes:

```
list <object type> @<server name>
```

- To list attributes of the active server:

```
list server
```

If no server other than the default server has been made active, lists attributes of the default server (it is the active server).

- To list attributes of the specified server:
Chapter 2  

PBS Commands

list server <server name>

• To list attributes of all schedulers:
  list sched

• To list attributes of the specified scheduler:
  list sched <scheduler name>

• To list all hooks, along with their attributes:
  list hook

• To list attributes of the specified hook:
  list hook <hook name>

2.49.8.1.i  Examples of Listing Objects and Their Attributes

Example 2-14: List serverA’s schedulers’ attributes:
  list sched @serverA

Example 2-15: List attributes for default server’s scheduler(s):
  l sched @default

Example 2-16: List PBS version for default server’s scheduler(s):
  l sched @default pbs_version

Example 2-17: List queues at a specified server:
  list queue @server1

2.49.8.2  Listing Resource Definitions

You can use the qmgr list and print commands to list resource definitions showing resource name, type, and flag(s).

• To list the name, type, and flag(s) of the named resource(s):
  list resource <resource name>[,<resource name> ...]
  or
  print resource <resource name>[,<resource name> ...]

• To list name, type, and flag(s) of custom resources only:
  list resource
  or
  print resource
  or
  print server (note that this also prints information for the active server)

• To list all custom resources at the specified server, with their names, types, and flags:
  list resource @<server name>
  or
  print resource @<server name>

When used by a non-privileged user, qmgr prints only resource definitions for resources that are visible to non-privileged users (those that do not have the i flag set).
2.49.8.3 Printing Creation and Configuration Commands

For printing the creation commands for any object except for a built-in hook.

- To print out the commands to create the named object(s) and set their attributes to their current values:
  \[ \text{print } \text{<object type>} \text{<object name>}, \text{<object name>} ...] \]
  where object name follows the name rules in section 2.49.5.1.ii, “Using Lists of Object Names”, on page 151.

- To print out the commands to create the named object and set its attributes to their current values:
  \[ \text{print } \text{<object type>} \text{<object name> } \text{[<attribute name>], <attribute name>]} ...] \]
  where object name follows the name rules in section 2.49.5.1.ii, “Using Lists of Object Names”, on page 151.

- To print out the commands to create and configure the active objects of the named type:
  \[ \text{print } \text{<object type>} \]

- To print out the commands to create and configure all of the objects of the specified type at the specified server:
  \[ \text{print } \text{<object type>} \text{<server name>} \]

- To print out the commands to create each queue, set the attributes of each queue to their current values, and set the attributes of the server to their current values:
  \[ \text{print server} \]
  This is used for the server and queues, but not hooks.
  Prints information for the active server. If there is no active server, prints information for the default server.

- To print out the creation commands for all schedulers:
  \[ \text{print sched} \]

- To print out the creation commands for the specified scheduler:
  \[ \text{print sched } \text{<scheduler name>} \]

2.49.8.4 Caveats for Viewing Information

Some attributes whose values are unset do not appear in the output of the \texttt{qmgr} command.
Definitions for built-in resources do not appear in the output of the \texttt{qmgr} command.

When a non-privileged user prints resource definitions, \texttt{qmgr} prints only resource definitions for resources that are visible to non-privileged users (those that do not have the \texttt{i} flag set).

2.49.9 Saving and Re-creating Server and Queue Information

To save and recreate server and queue configuration, print the configuration information to a file, then read it back in later. For example, to save your configuration:

\[
\begin{align*}
\# \texttt{qmgr -c "print server" > savedsettings} \\
\text{or} \\
\texttt{Qmgr: print server > savedsettings}
\end{align*}
\]

When re-creating queue and server configuration, read the commands back into \texttt{qmgr}. For example:

\[
\begin{align*}
\texttt{qmgr < savedsettings}
\end{align*}
\]
2.49.10 Operating on Hooks

2.49.10.1 Creating Hooks

- To create a hook:
  
  \texttt{Qmgr: create hook <hook name>}

  For example:
  
  \texttt{Qmgr: create hook my\_hook}

2.49.10.2 Deleting Hooks

- To delete a hook:
  
  \texttt{Qmgr: delete hook <hook name>}

  For example:
  
  \texttt{Qmgr: delete hook my\_hook}

2.49.10.3 Setting and Unsetting Hook Attributes

- To set a hook attribute:
  
  \texttt{Qmgr: set hook <hook name> <attribute> = <value>}

- To unset a hook attribute:
  
  \texttt{Qmgr: unset hook <hook name> <attribute>}

Example 2-18: Unset hook1’s \texttt{alarm} attribute, causing hook1’s \texttt{alarm} to revert to its default value of 30 seconds:

  \texttt{Qmgr: unset hook hook1 alarm}

2.49.10.4 Importing Hooks

For importing the contents of a hook. Cannot be used with built-in hooks.

To import a hook, you import the contents of a hook script into the hook. You must specify a filename that is locally accessible to \texttt{qmgr} and the PBS server.

Format for importing a site-defined hook:

\texttt{import hook <hook name> application/x-python <content encoding> {<input file> | -}}

This uses the contents of \texttt{input file} or \texttt{stdin (-)} as the contents of hook \texttt{hook name}.

- The \texttt{input file} or \texttt{stdin (-)} data must have a format of \texttt{content type} and must be encoded with \texttt{content encoding}.
- The allowed values for \texttt{content encoding} are “default” (7bit) and “base64”.
- If the source of input is \texttt{stdin (-)} and \texttt{content encoding} is “default”, \texttt{qmgr} expects the input data to be terminated by \texttt{EOF}.
- If the source of input is \texttt{stdin (-)} and \texttt{content encoding} is “base64”, \texttt{qmgr} expects input data to be terminated by a blank line.
- \texttt{input file} must be locally accessible to both \texttt{qmgr} and the requested batch server.
- A relative path \texttt{input file} is relative to the directory where \texttt{qmgr} was executed.
- If a hook already has a content script, that is overwritten by this import call.
- If the name in \texttt{input file} contains spaces as are used in Windows filenames, \texttt{input file} must be quoted.

There is no restriction on the size of the hook script.
2.49.10.4.i  Examples of Importing Hooks

Example 2-19: Given a Python script in ASCII text file "hello.py", use its contents as the script contents of hook1:

```bash
# cat hello.py
import pbs
pbs.event().job.comment="Hello, world"
# qmgr -c 'import hook hook1 application/x-python default hello.py'
```

Example 2-20: Given a base64-encoded file "hello.py.b64", qmgr unencodes the file's contents, and then makes this the script contents of hook1:

```bash
# cat hello.py.b64
CgF3bnQgImhlbGxvLCB3b3JsZCIK
# qmgr -c 'import hook hook1 application/x-python base64 hello.py.b64'
```

Example 2-21: To create a provisioning hook called Provision_Hook, and import the ASCII hook script called "master_provision.py" located in /root/data/:

```bash
Qmgr: create hook Provision_Hook
Qmgr: import hook Provision_Hook application/x-python default /root/data/master_provision.py
```

2.49.10.5  Importing and Exporting Hook Configuration Files

2.49.10.5.i  Importing Configuration Files

For importing the contents of a site-defined or built-in hook configuration file. To import a hook configuration file, you import the contents of a file to a hook. You must specify a filename that is locally accessible to qmgr and the PBS server.

Format for importing a site-defined hook configuration file:

```bash
import hook <hook name> application/x-config <content encoding> {<config file>|-}
```

Format for importing a built-in hook configuration file:

```bash
import pbshook <hook name> application/x-config <content encoding> {<config file>|-}
```

This uses the contents of `config file` or `stdin` (-) as the contents of the configuration file for hook `hook name`.

- The `config file` or `stdin` (-) data must have a format of `content-type` and must be encoded with `content encoding`.
- The allowed values for `content encoding` are "default" (7bit) and "base64".
- If the source of input is `stdin` (-) and `content encoding` is "default", qmgr expects the input data to be terminated by EOF.
- If the source of input is `stdin` (-) and `content encoding` is "base64", qmgr expects input data to be terminated by a blank line.
- `config file` must be locally accessible to both qmgr and the requested batch server.
- A relative path `config file` is relative to the directory where qmgr was executed.
- If a hook already has a configuration file, that file is overwritten by this import call.
- If the name in `config file` contains spaces as are used in Windows filenames, `input file` must be quoted.

There is no restriction on the size of the hook configuration file.

2.49.10.5.ii  Exporting Configuration Files

Format for exporting a site-defined hook configuration file:

```bash
qmgr -c "export hook <hook name> application/x-config default" > {<config file>|-}
```
Format for exporting a built-in hook configuration file:

```
qmgr -c "export pbshook <hook name> application/x-config default" > {<config file>|-}
```

### 2.49.10.5.iii Hook Configuration File Format

PBS supports several file formats for configuration files. The format of the file is specified in its suffix. Formats can be any of the following:

- `.ini`
- `.json`
- `.py` (Python)
- `.txt` (generic, no special format)
- `.xml`

- No suffix: treat the input file as if it is a `.txt` file
- The dash (`-`) symbol: configuration file content is taken from `STDIN`. The content is treated as if it is a `.txt` file.

Example 2-22: To import a configuration file in `.json` format:

```
# qmgr -c "import hook my_hook application/x-config default my_input_file.json"
```

### 2.49.10.6 Exporting Hooks

For exporting the contents of a hook. Cannot be used with built-in hooks.

Format for exporting a hook:

```
qmgr -c "export hook <hook name> <content type> <content encoding>" > [output file]
```

This dumps the script contents of hook `hook name` into `output file`, or `stdout` if `output file` is not specified.

- The resulting `output file` or `stdout` data is of `content type` and `content encoding`.
- The only `content type` currently supported is `"application/x-python"`.
- The allowed values for `content encoding` are `"default"` (7bit) and `"base64"`.
- `output file` must be a path that can be created by `qmgr`.
- Any relative path `output file` is relative to the directory where `qmgr` was executed.
- If `output file` already exists it is overwritten. If PBS is unable to overwrite the file due to ownership or permission problems, an error message is displayed in `stderr`.
- If the `output file` name contains spaces like the ones used in Windows file names, `output file` must be enclosed in quotes.

#### 2.49.10.6.i Examples of Exporting Hooks

Example 2-23: Dump hook1’s script contents directly into a file "hello.py.out"

```
# qmgr -c "export hook hook1 application/x-python default" > hello.py
# cat hello.py
import pbs
pbs.event().job.comment="Hello, world"
```

Example 2-24: To dump the script contents of a hook 'hook1' into a file in “My Hooks\hook1.py”:

```
qmgr -c "export hook hook1 application/x-python default" > "My Hooks\hook1.py"
```

### 2.49.10.7 Printing Hook Information

- To print out the commands to create and configure all hooks, including their configuration files:
print hook

- To print out the commands to create and configure the specified hook, including its configuration file:
  
  `print hook <hook name>`

### 2.49.10.8 Saving and Re-creating Hook Information

You can save creation and configuration information for all hooks. For example:

```bash
# qmgr -c "print hook" > hook.qmgr
```

You can re-create all hooks and their configuration files. For example:

```bash
# qmgr < hook.qmgr
```

### 2.49.10.9 Restrictions on Built-in Hooks

You cannot do the following with built-in hooks:

- Import a built-in hook
- Export a built-in hook
- Print creation commands for a built-in hook
- Create a built-in hook
- Delete a built-in hook
- Set the `type` attribute for a built-in hook

### 2.49.11 Printing Usage Information

You use the `help` command or a question mark ("?") to invoke the `qmgr` built-in help function. You can request usage information for any of the `qmgr` commands, and for topics including attributes, operators, names, and values.

- To print out usage information for the specified command or topic:
  
  `Qmgr: help [<command or topic>]`

  or

  `Qmgr: ? [<command or topic>]`

  For example, to print usage information for the `set` command:

  ```bash
  qmgr
  Qmgr: help set
  Syntax: set object [name][,name...] attribute[.resource] OP value
  ```

### 2.49.12 Standard Input

When you start a `qmgr` session, the `qmgr` command reads standard input for directives until it reaches end-of-file, or it reads the `exit` or `quit` command.

### 2.49.13 Standard Output

When you start a `qmgr` session, and standard output is connected to a terminal, `qmgr` writes a command prompt to standard output.

If you specify the `-e` option, `qmgr` echoes the directives it reads from standard input to standard output.
2.49.14 Standard Error

If you do not specify the -z option, the qmgr command writes a diagnostic message to standard error for each error occurrence.

2.49.15 Exit Status

0  
Success
1  
Error in parsing
2  
Error in execution
3  
Error connecting to server
4  
Error making object active
5  
Memory allocation error

2.49.16 See Also

The PBS Professional Administrator’s Guide, Chapter 6, "Attributes", on page 291, Chapter 5, "List of Built-in Resources", on page 277
2.50  qmove

Moves a PBS job from one queue to another

2.50.1  Synopsis

qmove <destination> <job ID> [Job ID> ...]  
qmove --version

2.50.2  Description

Moves a job from one queue to another.  The behavior of the qmove command may be affected by any site hooks. Site hooks can modify the job’s attributes, change its routing, etc.

2.50.2.1  Restrictions

The qmove command can be used on job arrays, but not on subjobs or ranges of subjobs.  Job arrays can only be moved from one server to another if they are in the ‘Q’, ‘H’, or ‘W’ states, and only if there are no running subjobs.  The state of the job array is preserved, and the job array will run to completion on the new server.  A job in the Running, Transiting, or Exiting state cannot be moved.  A job in the process of provisioning cannot be moved.

2.50.2.2  Effect of Privilege on Behavior

An unprivileged user can use the qmove command to move a job only when the move would not violate queue restrictions.  A privileged user (root, Manager, Operator) can use the qmove command to move a job under some circumstances where an unprivileged user cannot.  The following restrictions apply only to unprivileged users:

•  The queue must be enabled
•  Moving the job into the queue must not exceed the queue’s limits for jobs or resources
•  If the job is an array job, the size of the job array must not exceed the queue’s max_array_size
•  The queue cannot have its from_route_only attribute set to True (accepting jobs only from routing queues)

2.50.3  Options

--version

The qmove command returns its PBS version information and exits.  This option can only be used alone.

2.50.4  Operands

destination

Where job(s) are to end up. First operand. Syntax:

<queue name>

Moves the job(s) into the specified queue at the job’s current server.

@<server name>
Chapter 2  

PBS Commands

Moves the job(s) into the default queue at specified server.

<queue name>@<server name>

Moves the job(s) into the specified queue at the specified server.

See Chapter 7, "Formats", on page 363 for destination identifier formats.

job ID

Job(s) and/or job array(s) to be moved to the new destination. The qmove command accepts one or more job ID operands of the form:

<sequence number>[.<server name>]@<server name>
<sequence number>[@<server name>]
<sequence number>[@<server name>]

Note that some shells require that you enclose a job array identifier in double quotes.

2.50.5  Standard Error

The qmove command writes a diagnostic messages to standard error for each error occurrence.

2.50.6  Exit Status

Zero

Upon successful processing of all the operands presented to the qmove command.

Greater than zero

If the qmove command fails to process any operand.

2.50.7  See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide
2.51 qmsg

Writes message string into one or more job output files

2.51.1 Synopsis

qmsg [-E] [-O] <message string> <job ID> [<job ID> ...]
qmsg --version

2.51.2 Description

Writes a message string into one or more output files of the job. Typically this is done to leave an informative message in the output of the job. Also called “sending a message to a job”.

The qmsg command writes messages into the files of jobs by sending a Message Job batch request to the batch server that owns the job. The qmsg command does not directly write the message into the files of the job.

The qmsg command cannot be used on job arrays, subjobs, or ranges of subjobs.

2.51.3 Options

-E
   The message is written to the standard error of each job.

-O
   The message is written to the standard output of each job.

--version
   The qmsg command returns its PBS version information and exits. This option can only be used alone.

(no options)
   The message is written to the standard error of each job.

2.51.4 Operands

message string
   The message to be written. String. First operand. If the string contains blanks, the string must be quoted. If the final character of the string is not a newline, a newline character is added when written to the job’s file.

job ID
   The job(s) to receive the message string. This operand follows the message string operand. Cannot be a job array, subjob, or range of subjobs. The qmsg command accepts one or more job ID operands of the form:
      <sequence number>[.<server name>][@<server name>]

2.51.5 Standard Error

The qmsg command writes a diagnostic message to standard error for each error occurrence.
2.51.6 Exit Status

Zero
Upon successful processing of all the operands presented to the qmsg command.

Greater than zero
If the qmsg command fails to process any operand.

2.51.7 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide
2.52 qorder

Swaps queue positions of two PBS jobs

2.52.1 Synopsis

qorder <job ID> <job ID>
qorder --version

2.52.2 Description

Exchanges positions in queue(s) of two jobs, whether in the same or different queue(s).
No attribute of either job, e.g. priority, is changed. The impact of interchanging the order within or between queues is dependent on local job scheduling policy; contact your systems administrator.

2.52.2.1 Restrictions

• A job in the running state cannot be reordered.
• The qorder command can be used on job arrays, but not on subjobs or ranges of subjobs.
• The two jobs must be located at the same server.

2.52.2.2 Effect of Privilege on Behavior

For an unprivileged user to reorder jobs, both jobs must be owned by the user. A privileged user (Manager, Operator) can reorder any jobs.

2.52.3 Options

--version
The qorder command returns its PBS version information and exits. This option can only be used alone.

2.52.4 Operands

Both operands are job IDs which specify the jobs to be exchanged. The qorder command accepts two job ID operands of the form:

<sequence number>[.<server name>][@<server name>]
<sequence number>[]<.server name>[@<server name>]

If you specify the server for both jobs, they must be at the same server.
Note that some shells require that you enclose a job array identifier in double quotes.

2.52.5 Standard Error

The qorder command writes diagnostic messages to standard error for each error occurrence.
2.52.6 Exit Status

Zero
   Upon successful processing of all the operands presented to the qorder command
Greater than zero
   If the qorder command fails to process any operand

2.52.7 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide
2.53 qrerun

Requeues a PBS job

2.53.1 Synopsis

qrerun [-W force] <job ID> [<job ID> ...]
qrerun --version

2.53.2 Description

If possible, kills the specified job(s), then requeues each job in the execution queue from which it was run.

The qrerun command can be used on jobs, job arrays, subjobs, and ranges of subjobs. If you give a job array identifier as an argument, the job array is returned to its initial state at submission time, or to its altered state if it has been qaltered. All of that job array’s subjobs are requeued, which includes those that are currently running, and those that are completed and deleted. If you give a subjob or range as an argument, those subjobs are requeued.

2.53.2.1 Restrictions

If a job is marked as not rerunnable, qrerun neither kills nor requeues the job. See the -r option for the qsub and qalter commands, and the Rerunable job attribute.

The qrerun command cannot requeue a job or subjob which is not running, is held, or is suspended.

2.53.2.2 Required Privilege

PBS Manager or Operator privilege is required to use this command.

2.53.3 Options

- W force
  The job is to be requeued even if the vnode on which the job is executing is unreachable, or if the job’s substate is provisioning.

--version
  The qrerun command returns its PBS version information and exits. This option can only be used alone.

2.53.4 Operands

The qrerun command accepts one or more job ID operands of the form:

<sequence number>[.<server name>][@<server name>]

Note that some shells require that you enclose a job array identifier in double quotes.
2.53.5 Standard Error

The `qrerun` command writes a diagnostic message to standard error for each error occurrence.

2.53.6 Exit Status

Zero
   Upon successful processing of all operands
Greater than zero
   Upon failure to process any operand

2.53.7 See Also

2.54 qrls

Releases holds on PBS jobs

2.54.1 Synopsis

```
qrls [-h <hold list>] <job ID> [<job ID> ...]
qrls --version
```

2.54.2 Description

The `qrls` command releases or removes holds on batch jobs or job arrays, but not on subjobs or ranges of subjobs.

A job may have one or more types of holds which make the job ineligible for execution.

When you `qrls` a job whose `Execution_Time` attribute is not set to a time in the future, the job changes to the `queued` state. If `Execution_Time` is in the future, the job changes to the `waiting` state.

Holds can be set by the owner, an Operator, or Manager, when a job has a dependency, or when a job has its `Execution_Time` attribute set to a time in the future. See "qhold" on page 144.

2.54.2.1 Effect of Privilege on Behavior

The following table shows the holds and the privilege required to release each:

<table>
<thead>
<tr>
<th>Hold Type</th>
<th>Meaning</th>
<th>Privilege Required to Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>User</td>
<td>Job owner, Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>o</td>
<td>Other</td>
<td>Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>s</td>
<td>System</td>
<td>Manager, administrator, root, PBS (dependency)</td>
</tr>
<tr>
<td>n</td>
<td>None</td>
<td>Job owner, Operator, Manager, administrator, root</td>
</tr>
<tr>
<td>p</td>
<td>Bad password</td>
<td>Administrator, root</td>
</tr>
</tbody>
</table>

If you try to release a hold for which the you do not have privilege, the entire request is rejected, and no holds are released.

2.54.3 Options

(no options)

- Defaults to `-h u`, removing `user` hold.

- `-h <hold list>`
  Types of hold to be released for the jobs. The `hold list` option argument is a string consisting of one or more of the letters `u`, `o`, or `s` in any combination, or one of the letters `n` or `p`.

- `--version`
  The `qrls` command returns its PBS version information and exits. This option can only be used alone.
2.54.4 Operands

The `qrls` command can be used on jobs and job arrays, but not on subjobs or ranges of subjobs. The `qrls` command accepts one or more job ID operands of the form:

\<sequence number>\.[\<server name>]\[@\<server name]\]

\<sequence number\>\[\]\.[\<server name>]\[@\<server name]\]

Note that some shells require that you enclose a job array identifier in double quotes.

2.54.5 Standard Error

The `qrls` command writes a diagnostic message to standard error for each error occurrence.

2.54.6 Exit Status

Zero

Upon successful processing of all the operands presented to the `qrls` command

Greater than zero

If the `qrls` command fails to process any operand

2.54.7 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide, "qhold" on page 144
2.55 qrun

Runs a PBS job immediately

2.55.1 Synopsis

qrun [-a] [-H <vnode specification>] [<job ID>] [<job ID> ...]
qrun [-a] [-H -] [<job ID>] [<job ID> ...]
qrun --version

2.55.2 Description

Forces a job to run, regardless of scheduling position or resource requirements.

The qrun command can be used on jobs, subjobs, or ranges of subjobs, but not on job arrays. When it is used on a range of subjobs, the non-running subjobs in that range are run.

When preemption is enabled, a scheduler preempts other jobs in order to run this job. Running a job via qrun gives the job higher preemption priority than any of the priorities defined in the preempt_prio scheduler parameter. See "Using Preemption" on page 179 in the PBS Professional Administrator’s Guide.

2.55.2.1 Required Privilege

In order to execute qrun, you must have PBS Operator or Manager privilege.

2.55.2.2 Caveats for qrun

- The job is run without respect for limits, primetime, or dedicated time.
- If you use a -H <vnode specification> option to run a job, but specify insufficient vnodes or resources, the job may not run correctly. Avoid using this option unless you are sure.
- If you don’t use the -H option, the job must be in the Queued state and reside in an execution queue.
- If you do use the -H option, the job must be in the Queued or Suspended state and reside in an execution queue.
- The qrun command cannot be used on a job that is in the process of provisioning.
- If you use the -H option, all schedulers are bypassed, and partition boundaries are ignored.

2.55.3 Options to qrun

-a

The qrun command exits before the job actually starts execution.
(no -H option)

The job is run immediately regardless of scheduling policy as long as the following are true:

• The queue in which the job resides is an execution queue.
• Either the resources required by the job are available, or preemption is enabled and the required resources can be made available by preempting jobs that are running.

The qrun command by itself, with no -H option, overrides the following:

• Limits on resource usage by users, groups, and projects
• Limits on the number of jobs that can be run at a vnode
• Boundaries between primetime and non-primetime, specified in backfill_prime
• Whether the job is in a primetime queue: you can run a job in a primetime slot even when it’s not prime-time, or vice versa. Primetime boundaries are not honored.
• Dedicated time: you can run a job in a dedicated time slot, even if it’s not in a dedicated time queue, and vice versa. However, dedicated time boundaries are still honored.

The qrun command by itself, with no -H option, does not override the following:

• Server and queue resource usage limits

(with -H option)

Do NOT use this option unless you know exactly what you are doing.

With the -H option, all scheduling policies are bypassed and the job is run directly. The job is run immediately on the named or previously-assigned vnodes, regardless of current usage on those vnodes or which scheduler manages those vnodes, with the exception of vnode state. The job is not run and the qrun request is rejected if any named vnode is down, already allocated exclusively, or would need to be allocated exclusively and another job is already running on the vnode. The job is run if the vnode is offline.

The -H option runs jobs that are queued or suspended.

If the qrun -H command is used on a job that requests an AOE, and that AOE is not instantiated on those vnodes, the vnodes are provisioned with the AOE.

If the job requests an AOE, and that AOE is not available on the specified vnodes, the job is held.

-H <vnode specification without resources>

The vnode specification without resources has this format:

(<vchunk>)[+(<vchunk>) ...]

where vchunk has the format

<vnode name>[+(vnode name) ...]

Example:

-H (VnodeA+VnodeB)+(VnodeC)

PBS applies one requested chunk from the job’s selection directive in round-robin fashion to each vchunk in the list. Each vchunk must be sufficient to run the job’s corresponding chunk, otherwise the job may not execute correctly.
-H <vnode specification with resources>
The vnode specification with resources has this format:

\[(<vchunk>)[+(<vchunk>) ...]\]
where vchunk has the format

\[<vnode name>:<vnode resources>[+<vnode name>:<vnode resources> ...]\]
and where vnode resources has the format

\[<resource name>=<value>[::<resource name>=<value> ...]\]
Example:

-\(H\) (VnodeA:mem=100kb:ncpus=1) +(VnodeB:mem=100kb:ncpus=2+VnodeC:mem=100kb)

PBS creates a new selection directive from the vnode specification with resources, using it instead of the original specification from the user. Any single resource specification results in the job’s original selection directive being ignored. Each vchunk must be sufficient to run the job’s corresponding chunk, otherwise the job may not execute correctly.

If the job being run requests -l place=exclhost, take extra care to satisfy the exclhost request. Make sure that if any vnodes are from a multi-vnoded host, all vnodes from that host are allocated. Otherwise those vnodes can be allocated to other jobs.

-H
Runs the job on the set of resources to which it is already assigned. You can run a job on the set of resources already assigned to the job, without having to list the resources, by using the - (dash) argument to the -H option.

--version
The qrun command returns its PBS version information and exits. This option can only be used alone.

2.55.4 Operands

Job ID
The qrun command accepts a list of job IDs, of the form:

<sequence number>[.<server name>][@<server name>]
<sequence number>[<index>][.<server name>][@<server name>]
<sequence number>[<index start>-<index end>][.<server name>][@<server name>]
Note that some shells require that you enclose a job array identifier in double quotes.
The vnode specification without resources has this format:

\[(<vchunk>)[+(<vchunk>) ...]\]

where vchunk has the format

\[<vnode name>[+<vnode name> ...]\]

Example:

\[-H (VnodeA+VnodeB)+(VnodeC)\]

The vnode specification with resources has this format:

\[(<vchunk>)[+(<vchunk>) ...]\]

where vchunk has the format

\[<vnode name>:<vnode resources>[+<vnode name>:<vnode resources> ...]\]

and where vnode resources has the format

\[<resource name>=<value>[:<resource name>=<value> ...]\]

Example:

\[-H (VnodeA:mem=100kb:ncpus=1) +(VnodeB:mem=100kb:ncpus=2+VnodeC:mem=100kb)\]

A vnode name is the name of the vnode, not the name of the host.

### 2.55.5 Standard Error

The `qrun` command writes a diagnostic message to standard error for each error occurrence.

### 2.55.6 Exit Status

- **Zero**
  - On success
- **Greater than zero**
  - If the `qrun` command fails to process any operand

### 2.55.7 See Also

The PBS Professional Administrator’s Guide
2.56 qselect

Selects specified PBS jobs

2.56.1 Synopsis


qselect --version

2.56.2 Description

The qselect command lists those jobs that meet the specified selection criteria. You can compare certain job attribute values to specified values using a comparison operator shown as op in the option description.

You can select jobs, job arrays, or subjobs. You can select jobs from one server per call to the command.

Each option acts as a filter restricting which jobs are listed.

You can select jobs according to the values of some of the resources in the Resource_List job attribute. You can also select jobs according the selection directive (although because this is a string, you can only check for equality or inequality.)

Jobs that are finished or moved are listed only when the -x or -H options are used. Otherwise, job selection is limited to queued and running jobs.

2.56.2.1 Comparison Operations

You can select jobs by comparing the values of certain job attributes to values you specify. The following table lists the comparison operations you can use:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Type of Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>.eq.</td>
<td>The value of the job attribute is equal to the value of the option argument.</td>
</tr>
<tr>
<td>.ne.</td>
<td>The value of the job attribute is not equal to the value of the option argument.</td>
</tr>
<tr>
<td>.ge.</td>
<td>The value of the job attribute is greater than or equal to the value of the option argument.</td>
</tr>
<tr>
<td>.gt.</td>
<td>The value of the job attribute is greater than the value of the option argument.</td>
</tr>
<tr>
<td>.le.</td>
<td>The value of the job attribute is less than or equal to the value of the option argument.</td>
</tr>
<tr>
<td>.lt.</td>
<td>The value of the job attribute is less than the value of the option argument.</td>
</tr>
</tbody>
</table>

For example, to select jobs whose Priority attribute has a value greater than 5:

```
qselect -p.gt.5
```

Where an optional comparison is not specified, the comparison operation defaults to .eq, meaning PBS checks whether the value of the attribute is equal to the option argument.
2.56.2.2 Required Permissions

When selecting jobs according to resource values, users without operator or manager privilege cannot specify custom resources which were created to be invisible to unprivileged users.

2.56.3 Options to qselect

(no options)

Lists all jobs at the server which the user is authorized to list (query status of).

-a [<op>] <date and time>

Deprecated. Restricts selection to those jobs whose Execution_Time attribute qualifies when compared to the date and time argument. You can select a range of execution times by using this option twice, to compare to a minimum time and a maximum time.

The date and time argument has the format:

[[CC]YY]MMDDhhmm[.SS]

where MM is the two digits for the month, DD is the day of the month, hh is the hour, mm is the minute, and the optional SS is the seconds. CC is the century and YY the year.

-A <account string>

Restricts selection to jobs whose Account_Name attribute matches the specified account string.

-c [<op>] <interval>

Restricts selection to jobs whose Checkpoint interval attribute meets the comparison criteria.

The interval argument can take one of the following values:

c

c=<minutes>

n

s

w

w=<minutes>

We give the range of interval values for the Checkpoint attribute the following ordered relationship:

n > s > c=<minutes> > c > u

(Information about w and w=<minutes> is not available.)

For an interval value of “u”, only “.eq,” and “.ne,” are valid.
-h <hold list>
Restricts the selection of jobs to those with a specific set of hold types. The holds in the Hold_Types job attribute must be the same as those in the hold list argument, but can be in a different order.

The hold list argument is a string consisting of the single letter n, or one or more of the letters u, o, p, or s in any combination. If letters are duplicated, they are treated as if they occurred once. The letters represent the hold types:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Hold Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>None</td>
</tr>
<tr>
<td>u</td>
<td>User</td>
</tr>
<tr>
<td>o</td>
<td>Other</td>
</tr>
<tr>
<td>p</td>
<td>Bad password</td>
</tr>
<tr>
<td>s</td>
<td>System</td>
</tr>
</tbody>
</table>

-H
Restricts selection to finished and moved jobs.

-J
Limits selection to job arrays only.

-I <resource list>
Restricts selection of jobs to those with specified resource amounts. Resource must be job-wide, or be mem, ncpus, or vmem.

The resource list is in the following format:
<resource name> <op> <value>[,<resource name> <op> <value> ...]

You must specify op, and you can use any of the comparison operators.

Because resource specifications for chunks using the select statement, and placement using the place statement, are stored as strings, the only useful operators for these are .eq. and .ne.

Unprivileged users cannot specify custom resources which were created to be invisible to unprivileged users.

-N <name>
Restricts selection of jobs to those with the specified value for the Job_Name attribute.

-p [<op>]<priority>
Restricts selection of jobs to those with the specified Priority value(s).

-P <project>
Restricts selection of jobs to those matching the specified value for the project attribute.

Format: Project Name; see "Project Name" on page 367

-q <destination>
Restricts selection to those jobs at the specified destination.

The destination may take one of the following forms:
<queue name>
Restricts selection to the specified queue at the default server.

@<server name>
Restricts selection to the specified server.

<queue name>@<server name>

Restricts selection to the specified queue at the specified server.

If the -q option is not specified, jobs are selected from the default server.

-r <rerun>
Restricts selection of jobs to those with the specified value for the Rerunable attribute. The option argument rerun must be a single character, either y or n.

-s <states>
Restricts job selection to those whose job_state attribute has the specified value(s).

The states argument is a character string consisting of any combination of these characters: B, E, F, H, M, Q, R, S, T, U, W, and X. (A repeated character is accepted, but no additional meaning is assigned to it.)

**Table 2-23: Job States**

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Job array has started execution</td>
</tr>
<tr>
<td>E</td>
<td>The Exiting state</td>
</tr>
<tr>
<td>F</td>
<td>The Finished state</td>
</tr>
<tr>
<td>H</td>
<td>The Held state</td>
</tr>
<tr>
<td>M</td>
<td>The Moved state</td>
</tr>
<tr>
<td>Q</td>
<td>The Queued state</td>
</tr>
<tr>
<td>R</td>
<td>The Running state</td>
</tr>
<tr>
<td>S</td>
<td>The Suspended state</td>
</tr>
<tr>
<td>T</td>
<td>The Transiting state</td>
</tr>
<tr>
<td>U</td>
<td>Job suspended due to workstation user activity</td>
</tr>
<tr>
<td>W</td>
<td>The Waiting state</td>
</tr>
<tr>
<td>X</td>
<td>The eXited state. Subjobs only</td>
</tr>
</tbody>
</table>

Jobs in any of the specified states are selected.

Job arrays are never in states R, S, T, or U. Subjobs may be in those states.
### PBS Commands

**Chapter 2**

- `-t <time option> [<op>] <specified time>`

Jobs are selected according to one of their time-based attributes. The **time option** specifies which time-based attribute is tested. You give the **specified time** in **datetime format**. See [Chapter 7, "Formats", on page 363](#).

The **time option** is one of the following:

**Table 2-24: Sub-options to the -t Option**

<table>
<thead>
<tr>
<th>Time Option</th>
<th>Time Attribute</th>
<th>Option Format(s)</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>a</code></td>
<td>Execution_Time</td>
<td><code>Timestamp</code></td>
<td>Time at which the job is eligible for execution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use <strong>datetime</strong> format to specify.</td>
<td></td>
</tr>
<tr>
<td><code>c</code></td>
<td>ctime</td>
<td><code>Timestamp</code></td>
<td>Time at which the job was created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Printed by <strong>qstat</strong> in human-readable <strong>Date</strong> format. Output in hooks as seconds since epoch.</td>
<td></td>
</tr>
<tr>
<td><code>e</code></td>
<td>etime</td>
<td><code>Timestamp</code></td>
<td>Time when job became eligible to run, i.e. was enqueued in an execution queue and was in the “Q” state. Reset when a job moves queues, or is held then released. Not affected by qaltering.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Printed by <strong>qstat</strong> in human-readable <strong>Date</strong> format. Output in hooks as seconds since epoch.</td>
<td></td>
</tr>
<tr>
<td><code>g</code></td>
<td>eligible_time</td>
<td>Use <strong>duration</strong> format to specify.</td>
<td>Amount of eligible time job accrued waiting to run.</td>
</tr>
<tr>
<td><code>m</code></td>
<td>mtime</td>
<td><code>Timestamp</code></td>
<td>Time that the job was last modified, changed state, or changed locations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Printed by <strong>qstat</strong> in human-readable <strong>Date</strong> format. Output in hooks as seconds since epoch.</td>
<td></td>
</tr>
<tr>
<td><code>q</code></td>
<td>qtime</td>
<td><code>Timestamp</code></td>
<td>Time that the job entered the current queue.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Printed by <strong>qstat</strong> in human-readable <strong>Date</strong> format. Output in hooks as seconds since epoch.</td>
<td></td>
</tr>
<tr>
<td><code>s</code></td>
<td>stime</td>
<td><code>Timestamp</code></td>
<td>Time the job started. Updated when job is restarted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Printed by <strong>qstat</strong> in human-readable <strong>Date</strong> format. Output in hooks as seconds since epoch.</td>
<td></td>
</tr>
<tr>
<td><code>t</code></td>
<td>estimated.start_time</td>
<td>Use <strong>datetime</strong> format to specify.</td>
<td>Job’s estimated start time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Printed by <strong>qstat</strong> in human-readable <strong>Date</strong> format. Output in hooks as seconds since epoch.</td>
<td></td>
</tr>
</tbody>
</table>

To bracket a time period, use the `-t` option twice. For example, to select jobs using **stime** between noon and 3 p.m.:

```
qselect -ts.gt.09251200 -ts.lt.09251500
```

**-T**

Limits selection to jobs and subjobs.
-u <user list>
   Restrictions selection to jobs owned by the specified usernames.
   Syntax of user list:
   <username>[@<hostname>],<username>[@<hostname>],...
   Selects jobs which are owned by the listed users at the corresponding hosts. Hostnames may be wildcarded on
   the left end, e.g. "*.nasa.gov". A username without a "[@<hostname>]" is equivalent to "<username>@*",
   meaning that it is valid at any host.

-x
   Selects finished and moved jobs in addition to queued and running jobs.

-X
   Selects completed or deleted subjobs (subjobs in X state).

--version
   The qselect command returns its PBS version information and exits. This option can only be used alone.

2.56.4 Standard Output
PBS writes a list of the selected job IDs to standard output. Each job ID is separated by white space. A job ID can repre-
sent a job, a job array, or a subjob. Each job ID has one of the forms:
<sequence number>.<server name>[@<server name>]
<sequence number>[.]<server name>[@<server name>]
<sequence number>[<index>].<server name>[@<server name>]
@<server name> identifies the server which currently owns the job.

2.56.5 Standard Error
The qselect command writes a diagnostic message to standard error for each error occurrence.

2.56.6 Exit Status
   Zero
   Upon successful processing of all options presented to the qselect command
   Greater than zero
   If the qselect command fails to process any option

2.56.7 See Also
341. Chapter 5, "List of Built-in Resources", on page 277
2.57 qsig

Send signal to PBS job

2.57.1 Synopsis

qsig [-s <signal>] <job ID> [<job ID> ...]
qsig --version

2.57.2 Description

The qsig command sends a signal to all the processes of the specified job(s). The \texttt{qsig} command sends a Signal Job batch request to the server which owns the job.

The \texttt{qsig} command can be used for jobs, job arrays, subjobs, and ranges of subjobs. If it is used on a range of subjobs, the running subjobs in the range are signaled.

Not all signal names are recognized by \texttt{qsig}; if using a signal name does not work, try issuing the signal number instead.

2.57.2.1 Using admin-suspend and admin-resume

If you have a vnode requiring maintenance while remaining powered up, where you don’t want jobs running during the maintenance, you can use the special signals \texttt{admin-suspend} and \texttt{admin-resume} to suspend and resume the jobs on the vnode. When you use \texttt{admin-suspend} on a vnode’s job(s), the vnode goes into the \texttt{maintenance} state, and its scheduler does not schedule jobs on it. You must separately \texttt{admin-suspend} each job on the vnode. When its last \texttt{admin-suspended} job is \texttt{admin-resumed}, a vnode leaves the \texttt{maintenance} state.

2.57.2.2 Restrictions

The request to signal a job is rejected if:

- The user is not authorized to signal the job
- The job is not in the \texttt{running} or \texttt{suspended} state
- The requested signal is not supported by the system upon which the job is executing
- The job is in the process of provisioning
- You attempt to use \texttt{admin-resume} on a job that was \texttt{suspended}
- You attempt to use \texttt{resume} on a job that was \texttt{admin-suspended}

2.57.2.3 Required Privilege

Manager or Operator privilege is required to use the \texttt{admin-suspend}, \texttt{admin-resume}, \texttt{suspend}, or \texttt{resume} signals. Unprivileged users can use other signals.

2.57.3 Options to qsig

(no options)

PBS sends \texttt{SIGTERM} to the job.
-s <signal>
  PBS sends signal signal to the job.

--version
  The qsig command returns its PBS version information and exits. This option can only be used alone.

### 2.57.3.1 Signals

You can send standard signals to a job, or the special signals described below. The signal argument can be in any of the following formats:

- A signal name, e.g. SIGKILL
- A signal name without the SIG prefix, e.g. KILL
- An unsigned signal number, e.g. 9

The signal name SIGNULL is allowed; in this case the server sends the signal 0 to the job, which has no effect.

#### 2.57.3.1.i Special Signals

The following special signals are all lower-case, and have no associated signal number:

- **admin-suspend**
  
  Suspends a job and puts its vnodes into the maintenance state. The job is put into the S state and its processes are suspended. When suspended, a job is not executing and is not charged for walltime.
  
  Syntax: `qsig -s admin-suspend <job ID>`

- **admin-resume**
  
  Resumes a job that was suspended using the admin-suspend signal, without waiting for its scheduler. Cannot be used on jobs that were suspended with the suspend signal.
  
  Syntax: `qsig -s admin-resume <job ID>`

- **suspend**
  
  Suspends specified job(s). Job goes into suspended (S) state. When suspended, a job is not executing and is not charged for walltime.

- **resume**
  
  Marks specified job(s) for resumption by its scheduler when there are sufficient resources. If you use qsig -s resume on a job that was suspended using qsig -s suspend, the job is resumed when there are sufficient resources. Cannot be used on jobs that were suspended with the admin_suspend signal.

### 2.57.4 Operands

The qsig command accepts one or more job ID operands. For a job, this has the form:

```
<sequence number>[@<server name>]
```

For a job array, job ID takes this form:

```
<sequence number>@<server name>
```

Note that some shells require that you enclose a job array identifier in double quotes.

### 2.57.5 Standard Error

The qsig command writes a diagnostic message to standard error for each error occurrence.
2.57.6 Exit Status

Zero
Upon successful processing of all the operands presented to the `qsig` command
Greater than zero
If the `qsig` command fails to process any operand

2.57.7 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide
2.58 qstart

Turns on scheduling or routing for the jobs in a PBS queue

2.58.1 Synopsis

qstart <destination> [ <destination> ... ]
qstart --version

2.58.2 Description

If destination is an execution queue, the qstart command allows a PBS scheduler to schedule jobs residing in the specified queue. If destination is a routing queue, the server can begin routing jobs from that queue. Sets the value of the queue’s started attribute to True.

2.58.2.1 Required Privilege

In order to execute qstart, you must have PBS Operator or Manager privilege.

2.58.3 Options

--version

The qstart command returns its PBS version information and exits. This option can only be used alone.

2.58.4 Operands

The qstart command accepts one or more space-separated destination operands. The operands take one of three forms:

<queue name>

Starts scheduling or routing from the specified queue.

@<server name>

Starts scheduling or routing from all queues at the specified server.

<queue name>@<server name>

Starts scheduling or routing from the specified queue at the specified server.

To start scheduling at all queues at the default server, use the qmgr command:

qmgr: set queue @default started=true

2.58.5 Standard Error

The qstart command writes a diagnostic message to standard error for each error occurrence.
2.58.6 Exit Status

Zero
  Upon successful processing of all the operands presented to the qstart command
Greater than zero
  If the qstart command fails to process any operand

2.58.7 See Also

The PBS Professional Administrator’s Guide, "qmgr" on page 146, "qstop" on page 205
2.59  qstat

Displays status of PBS jobs, queues, or servers

2.59.1  Synopsis

2.59.1.1  Displaying Job Status

Default format:
```
qustat [-E] [-J] [-p] [-t] [-x] [[<job ID> | <destination>]] ...
```

Long format:
```
```

Alternate format:
```
qustat -a [-w] [-H] [-i] [-r] [-E] [-G | -M] [-J] [-n [-I] [-w]] [-s [-I] [-w]] [-t [-T [-w]] [-u <user list>]] [[<job ID> | <destination>]] ...
```

2.59.1.2  Displaying Queue Status

Default format:
```
qustat -Q [<destination>] ...
```

Long format:
```
qustat -Q -f [-F json|dsv [-D <delimiter>]] [-w] [<destination>] ...
```

Alternate format:
```
qustat -q [-G | -M] [<destination>] ...
```

2.59.1.3  Displaying Server Status

Default format:
```
qustat -B [<server name>] ...
```

Long format:
```
qustat -B -f [-F json|dsv [-D <delimiter>]] [-w] [<server name>] ...
```

2.59.1.4  Displaying Version Information

```
qustat --version
```

2.59.2  Description

The `qstat` command displays the status of jobs, queues, or servers, writing the status information to standard output. When displaying job status information, the `qstat` command displays status information about all specified jobs, job arrays, and subjobs. You can specify jobs by ID, or by destination, for example all jobs at a specified queue or server.
2.59.2.1 Display Formats

You can use particular options to display status information in a default format, an alternate format, or a long format. Default and alternate formats display all status information for a job, queue, or server with one line per object, in columns. Long formats display status information showing all attributes, one attribute to a line.

2.59.2.2 Displaying Information for Finished and Moved Jobs

You can display status information for finished and moved jobs by using the -x and -H options.

If your job has been moved to another server through peer scheduling, give the job ID as an argument to `qstat`. If you do not specify the job ID, your job will not appear to exist. For example, your job 123.ServerA is moved to ServerB. In this case, you can use:

```
qstat 123
```

or

```
qstat 123.ServerA
```

Specifying the full job name, including the server, avoids the possibility that `qstat` will report on a job named 123.ServerB that was moved to ServerA.

To list all jobs at ServerB, you can use:

```
qstat @ServerB
```

2.59.2.3 Required Privilege

Users without Manager or Operator privilege cannot view resources or attributes that are invisible to unprivileged users.

2.59.3 Displaying Job Status

2.59.3.1 Job Status in Default Format

Triggers: any of the `-J`, `-p`, `-t`, or `-x` options.

The `qstat` command displays job status in default format when you specify any of the `-J`, `-p`, `-t`, or `-x` options. Jobs are displayed one to a line, with these column headers:

```
Job id   Name       User      Time Use S Queue
-------- ---------- --------- -------- - -----
```

Description of columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job id</td>
<td>The job ID assigned by PBS</td>
</tr>
<tr>
<td>Name</td>
<td>Job name specified by submitter</td>
</tr>
<tr>
<td>User</td>
<td>Username of job owner</td>
</tr>
</tbody>
</table>
Table 2-25: Description of Default Job Status Columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Use</td>
<td>The CPU time used by the job. Before the application has actually started running, for example during stage-in, this field is &quot;0&quot;. At the point where the application starts accumulating \texttt{cput}, this field changes to &quot;00:00:00&quot;. After that, every time the MoM polls for resource usage, the field is updated. The MoM on each execution host polls for the usage of all processes on her host belonging to the job. Usage is summed. The polling interval is short when a job first starts running and lengthens to a maximum of 2 minutes. See &quot;Configuring MoM Polling Cycle&quot; on page 45 in the PBS Professional Administrator's Guide.</td>
</tr>
<tr>
<td>S</td>
<td>The job’s state. See section 8.1, “Job States”, on page 371</td>
</tr>
<tr>
<td>B</td>
<td>Array job has at least one subjob running</td>
</tr>
<tr>
<td>E</td>
<td>Job is exiting after having run</td>
</tr>
<tr>
<td>F</td>
<td>Job is finished</td>
</tr>
<tr>
<td>H</td>
<td>Job is held</td>
</tr>
<tr>
<td>M</td>
<td>Job was moved to another server</td>
</tr>
<tr>
<td>Q</td>
<td>Job is queued</td>
</tr>
<tr>
<td>R</td>
<td>Job is running</td>
</tr>
<tr>
<td>S</td>
<td>Job is suspended</td>
</tr>
<tr>
<td>T</td>
<td>Job is being moved to new location</td>
</tr>
<tr>
<td>U</td>
<td>Cycle-harvesting job is suspended due to keyboard activity</td>
</tr>
<tr>
<td>W</td>
<td>Job is waiting for its submitter-assigned start time to be reached</td>
</tr>
<tr>
<td>X</td>
<td>Subjob has completed execution or has been deleted</td>
</tr>
<tr>
<td>Queue</td>
<td>The queue in which the job resides</td>
</tr>
</tbody>
</table>

2.59.3.2 Job Status in Long Format

Trigger: the -f option.

If you specify the -f (full) option, full job status information for each job is displayed in this order:

- The job ID
- Each job attribute, one to a line
- The job’s submission arguments
- The job’s executable, in JSDL format
- The executable’s argument list, in JSDL format

The job attributes are listed as \texttt{<name>=<value>} pairs. This includes the \texttt{exec_host} and \texttt{exec_vnode} strings. The full output can be very large.

The \texttt{exec_host} string has this format:

\texttt{<host1>/<T1>*<P1>[+<host2>/<T2>*<P2>+... ]}

where

\texttt{T1} is the task slot number (the index) of the job on host1.
$P1$ is the number of processors allocated to the job from host1. The number of processors allocated does not appear if it is 1.

The `exec_vnode` string has the format:

\[(<\text{vnode}1>:\text{ncpus}=<N1>:\text{mem}=<M1>)+(<\text{vnode}2>:\text{ncpus}=<N2>:\text{mem}=<M2>))++\]

where

$N1$ is the number of CPUs allocated to that job on $vnode1$.

$M1$ is the amount of memory allocated to that job on $vnode1$.

### 2.59.3.3 Job Status in Alternate Format

Triggers: any of the `-a`, `-i`, `-G`, `-H`, `-M`, `-n`, `-r`, `-s`, or `-u <user list>` options.

The `qstat` command displays job status in alternate format if you specify any of the `-a`, `-i`, `-G`, `-H`, `-M`, `-n`, `-r`, `-s`, or `-u <user list>` options. Jobs are displayed one to a line. If jobs are running and the `-n` option is specified, or if jobs are finished or moved and the `-H` and `-n` options are specified, there is a second line for the `exec_host` string.

#### 2.59.3.3.i Job Status Alternate Format Output Columns

Alternate format job status output contains the following columns:

<table>
<thead>
<tr>
<th>Req’d</th>
<th>Req’d</th>
<th>Elap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job ID</td>
<td>Username</td>
<td>Queue</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>------</td>
</tr>
</tbody>
</table>

Description of columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job ID</td>
<td>The job ID assigned by PBS</td>
</tr>
<tr>
<td>Username</td>
<td>Username of job owner</td>
</tr>
<tr>
<td>Queue</td>
<td>Queue in which the job resides</td>
</tr>
<tr>
<td>Jobname</td>
<td>Job name specified by submitter</td>
</tr>
<tr>
<td>SessID</td>
<td>Session ID. Appears only if the job is running</td>
</tr>
<tr>
<td>NDS</td>
<td>Number of chunks or vnodes requested by the job</td>
</tr>
<tr>
<td>TSK</td>
<td>Number of CPUs requested by the job</td>
</tr>
<tr>
<td>Req’d Memory</td>
<td>Amount of memory requested by the job</td>
</tr>
<tr>
<td>Req’d Time</td>
<td>If CPU time is requested, this shows CPU time. Otherwise, shows walltime</td>
</tr>
<tr>
<td>S</td>
<td>The job’s state; see &quot;States&quot; on page 371 for states</td>
</tr>
<tr>
<td>Elap Time</td>
<td>If CPU time is requested, this shows CPU time. Otherwise, shows walltime</td>
</tr>
</tbody>
</table>

### 2.59.3.4 Grouping Jobs and Sorting by ID

Trigger: the -E option.
You can use the -E option to sort and group jobs in the output of `qstat`. The -E option groups jobs by server and displays each group by ascending ID. This option also improves `qstat` performance. The following table shows how the -E option affects the behavior of `qstat`:

**Table 2-27: How -E Option Affects qstat Output**

<table>
<thead>
<tr>
<th>How qstat is Used</th>
<th>Result Without -E</th>
<th>Result With -E</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>qstat</code> (no job ID specified)</td>
<td>Queries the default server and displays result</td>
<td>No change in behavior; same as without -E option</td>
</tr>
<tr>
<td><code>qstat &lt;list of job IDs from single server&gt;</code></td>
<td>Displays results in the order specified</td>
<td>Displays results in ascending ID order</td>
</tr>
<tr>
<td><code>qstat &lt;job IDs at multiple servers&gt;</code></td>
<td>Displays results in the order they are specified</td>
<td>Groups jobs by server. Displays each group in ascending order</td>
</tr>
</tbody>
</table>

### 2.59.4 Displaying Queue Status

#### 2.59.4.1 Queue Status in Default Format

Trigger: the -Q option by itself.

The `qstat` command displays queue status in default format if the only option is -Q. Queue status is displayed one queue to a line, with these column headers:

```
Queue       Max  Tot  Ena  Str  Que  Run  Hld  Wat  Trn  Ext  Type
----------- ---- ---- ---- --- ---- ---- ---- ---- ---- ---- ----
```

Description of columns:

**Table 2-28: Description of Default Queue Status Columns**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue</td>
<td>Queue name</td>
</tr>
<tr>
<td>Max</td>
<td>Maximum number of jobs allowed to run concurrently in this queue</td>
</tr>
<tr>
<td>Tot</td>
<td>Total number of jobs in the queue</td>
</tr>
<tr>
<td>Ena</td>
<td>Whether the queue is enabled or disabled</td>
</tr>
<tr>
<td>Str</td>
<td>Whether the queue is started or stopped</td>
</tr>
<tr>
<td>Que</td>
<td>Number of queued jobs</td>
</tr>
<tr>
<td>Run</td>
<td>Number of running jobs</td>
</tr>
<tr>
<td>Hld</td>
<td>Number of held jobs</td>
</tr>
<tr>
<td>Wat</td>
<td>Number of waiting jobs</td>
</tr>
<tr>
<td>Trn</td>
<td>Number of jobs being moved (transiting)</td>
</tr>
<tr>
<td>Ext</td>
<td>Number of exiting jobs</td>
</tr>
<tr>
<td>Type</td>
<td>Type of queue: execution or routing</td>
</tr>
</tbody>
</table>
2.59.4.2 Queue Status in Long Format

Trigger: the -q and -f options together.

If you specify the -f (full) option with the -q option, full queue status information for each queue is displayed starting with the queue name, followed by each attribute, one to a line, as <name> = <value> pairs.

2.59.4.2.i Queue Status: Alternate Format

Triggers: any of the -q, -G, or -M options.

The qstat command displays queue status in the alternate format if you specify any of the -q, -G, or -M options. Queue status is displayed one queue to a line, and the lowest line contains totals for some columns.

These are the alternate format queue status column headers:

Table: Queue Alternate Status Columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue</td>
<td>Queue name</td>
</tr>
<tr>
<td>Memory</td>
<td>Maximum amount of memory that can be requested by a job in this queue</td>
</tr>
<tr>
<td>CPU Time</td>
<td>Maximum amount of CPU time that can be requested by a job in this queue</td>
</tr>
<tr>
<td>Walltime</td>
<td>Maximum amount of walltime that can be requested by a job in this queue</td>
</tr>
<tr>
<td>Node</td>
<td>Maximum number of vnodes that can be requested by a job in this queue</td>
</tr>
<tr>
<td>Run</td>
<td>Number of running and suspended jobs. Lowest row is total number of running and suspended jobs in all the queues shown</td>
</tr>
<tr>
<td>Que</td>
<td>Number of queued, waiting, and held jobs. Lowest row is total number of queued, waiting, and held jobs in all the queues shown</td>
</tr>
<tr>
<td>Lm</td>
<td>Maximum number of jobs allowed to run concurrently in this queue</td>
</tr>
<tr>
<td>State</td>
<td>State of this queue: E (enabled) or D (disabled), and R (running) or S (stopped)</td>
</tr>
</tbody>
</table>

2.59.5 Displaying Server Status

2.59.5.1 Server Status in Default Format:

Trigger: the -B option.

The qstat command displays server status if the only option given is -B.

Column headers for default server status output:

Table: Server Status

<table>
<thead>
<tr>
<th>Server</th>
<th>Max</th>
<th>Tot</th>
<th>Que</th>
<th>Run</th>
<th>Hld</th>
<th>Wat</th>
<th>Trn</th>
<th>Ext</th>
<th>Status</th>
</tr>
</thead>
</table>

Table: Description of Queue Alternate Status Columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue</td>
<td>Queue name</td>
</tr>
<tr>
<td>Memory</td>
<td>Maximum amount of memory that can be requested by a job in this queue</td>
</tr>
<tr>
<td>CPU Time</td>
<td>Maximum amount of CPU time that can be requested by a job in this queue</td>
</tr>
<tr>
<td>Walltime</td>
<td>Maximum amount of walltime that can be requested by a job in this queue</td>
</tr>
<tr>
<td>Node</td>
<td>Maximum number of vnodes that can be requested by a job in this queue</td>
</tr>
<tr>
<td>Run</td>
<td>Number of running and suspended jobs. Lowest row is total number of running and suspended jobs in all the queues shown</td>
</tr>
<tr>
<td>Que</td>
<td>Number of queued, waiting, and held jobs. Lowest row is total number of queued, waiting, and held jobs in all the queues shown</td>
</tr>
<tr>
<td>Lm</td>
<td>Maximum number of jobs allowed to run concurrently in this queue</td>
</tr>
<tr>
<td>State</td>
<td>State of this queue: E (enabled) or D (disabled), and R (running) or S (stopped)</td>
</tr>
</tbody>
</table>
Description of columns:

**Table 2-30: Description of Server Status Default Display Columns**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Name of server</td>
</tr>
<tr>
<td>Max</td>
<td>Maximum number of jobs allowed to be running concurrently on the server</td>
</tr>
<tr>
<td>Tot</td>
<td>Total number of jobs currently managed by the server</td>
</tr>
<tr>
<td>Que</td>
<td>Number of queued jobs</td>
</tr>
<tr>
<td>Run</td>
<td>Number of running jobs</td>
</tr>
<tr>
<td>Hld</td>
<td>Number of held jobs</td>
</tr>
<tr>
<td>Wat</td>
<td>Number of waiting jobs</td>
</tr>
<tr>
<td>Tm</td>
<td>Number of transiting jobs</td>
</tr>
<tr>
<td>Ext</td>
<td>Number of exiting jobs</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the server</td>
</tr>
</tbody>
</table>

### 2.59.5.2 Server Status in Long Format

Trigger: the `-f` option.

If you specify the `-f` (full) option, displays full server status information starting with the server name, followed by each server attribute, one to a line, as `<name> = <value>` pairs. Includes PBS version information.

### 2.59.6 Options to *qstat*

#### 2.59.6.1 Generic Job Status Options

- `-E`  
  Groups jobs by server and displays jobs sorted by ascending ID. When *qstat* is presented with a list of jobs, jobs are grouped by server and each group is displayed by ascending ID. This option also improves *qstat* performance. See section 2.59.3.4, “Grouping Jobs and Sorting by ID”, on page 195.

#### 2.59.6.2 Default Job Status Options

The following options cause job status information to be displayed in default format:

- `-J`  
  Displays status information for job arrays (not subjobs).

- `-t`  
  Displays status information for jobs, job arrays, and subjobs. When used with `-J` option, displays status information for subjobs only.

- `-p`  
  The *Time Use* column is replaced with the percentage completed for the job. For a job array this is the percentage of subjobs completed. For a normal job, it is the percentage of allocated CPU time used.

- `-x`  
  Displays status information for finished and moved jobs in addition to queued and running jobs.
2.59.6.3 Alternate Job Status Options

The following options cause job status information to be displayed in alternate format:

- **-a**
  All queued and running jobs are displayed. If a destination is specified, information for all jobs at that destination is displayed. If a job ID is specified, information about that job is displayed. Always specify this option before the -n or -s options, otherwise they will not take effect.

- **-H**
  Without a job identifier, displays information for all finished or moved jobs. If a job ID is given, displays information for that job regardless of its state. If a destination is specified, displays information for finished or moved jobs, or specified job(s), at destination.

- **-i**
  If a destination is given, information for queued, held or waiting jobs at that destination is displayed. If a job ID is given, information about that job is displayed regardless of its state.

- **-n**
  The exec_host string is listed on the line below the basic information. If the -1 option is given, the exec_host string is listed on the end of the same line. If using the -a option, always specify the -n option after -a, otherwise the -n option does not take effect.

- **-r**
  If a destination is given, information for running or suspended jobs at that destination is displayed. If a job ID is given, information about that job is displayed regardless of its state.

- **-s**
  Any comment added by the administrator or scheduler is shown on the line below the basic information. If the -1 option is given, the comment string is listed on the end of the same line. If using the -a option, always specify the -s option after -a, otherwise the -s option does not take effect.
Displays estimated start time for queued jobs, replacing the *Elap Time* field with the *Est Start Time* field. Jobs with earlier estimated start times are displayed before those with later estimated start times.

Running jobs are displayed before other jobs. Running jobs are sorted by their *stime* attribute (start time). Queued jobs whose estimated start times are unset (*estimated.start_time* = *unset*) are displayed after those with estimated start times, with the unset value shown as a double dash (“--”). Queued jobs with estimated start times in the past are treated as if their estimated start times are unset.

If a job’s estimated start time cannot be calculated, the start time is shown as a question mark (“?”). Time displayed is local to the `qstat` command. Current week begins on Sunday.

The following table shows the format for the *Est Start Time* field when the `-w` option is not used:

**Table 2-31: Format for Estimated Start Time Field without `-w` Option**

<table>
<thead>
<tr>
<th>Format</th>
<th>Job Estimated Start Time</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;HH&gt;:&lt;MM&gt;</code></td>
<td>Today</td>
<td>15:34</td>
</tr>
<tr>
<td><code>&lt;2-letter weekday&gt; &lt;HH&gt;</code></td>
<td>Within 7 days, but after today</td>
<td>We 15</td>
</tr>
<tr>
<td><code>&lt;3-letter month name&gt;</code></td>
<td>This calendar year, but after this week</td>
<td>Feb</td>
</tr>
<tr>
<td><code>&lt;YYYY&gt;</code></td>
<td>Less than or equal to 5 years from today, after this year</td>
<td>2018</td>
</tr>
<tr>
<td><code>&gt;5yrs</code></td>
<td>More than 5 years from today</td>
<td>&gt;5yrs</td>
</tr>
</tbody>
</table>

The following table shows the format for the *Est Start Time* field when the `-w` option is used:

**Table 2-32: Format for Estimated Start Time Field with `-w` Option**

<table>
<thead>
<tr>
<th>Format</th>
<th>Job Estimated Start Time</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Today</em> <code>&lt;HH&gt;:&lt;MM&gt;</code></td>
<td>Today</td>
<td><em>Today</em> 13:34</td>
</tr>
<tr>
<td><code>&lt;Day&gt; &lt;HH&gt;:&lt;MM&gt;</code></td>
<td>This week, but after today</td>
<td>Wed 15:34</td>
</tr>
<tr>
<td><code>&lt;Day&gt; &lt;Month&gt; &lt;Daynum&gt; &lt;HH&gt;:&lt;MM&gt;</code></td>
<td>This year, but after this week</td>
<td>Wed Feb 10 15:34</td>
</tr>
<tr>
<td><code>&lt;Day&gt; &lt;Month&gt; &lt;Daynum&gt; &lt;YYYY&gt; &lt;HH&gt;:&lt;MM&gt;</code></td>
<td>After this year</td>
<td>Wed Feb 10 2011 15:34</td>
</tr>
</tbody>
</table>

When used with the `-t` option, prints the full timezone-qualified start time.

Estimated start time information can be made unavailable to unprivileged users; in this case, the estimated start time appears to be unset.

**-u <user list>**

If a destination is given, status for jobs at that destination owned by users in *user list* is displayed. If a job ID is given, status information for that job is displayed regardless of the job’s ownership.

Format: `<username>[@<hostname>], <username>[@<hostname>], ...` in comma-separated list.

Hostnames may be wildcarded, but not domain names. When no hostname is specified, *username* is for any host.
-w
Allows display of wider fields up to 120 characters. The Job ID column can be up to 30 characters wide. User-
name, Queue, and Jobname can be up to 15 characters wide. SessID can be up to 8 characters wide and
NDS can be up to 4 characters wide. TSK can be up to 5 characters wide. Req d Memory can be 6 characters,
Elap Time can be 5 characters, and S can be only 1 character wide. Can be used only in conjunction with the -a,-n, or -s options. This option is different from the -w option used with -f.

-1
Reformats qstat output to a single line. Can be used only in conjunction with the -n and/or -s options.

2.59.6.4 Queue Status Options

-Q
Displays queue status in default format. Operands must be destinations.

-q
Displays queue status in alternate format. Operands must be destinations.

2.59.6.5 Server Status Options

-B
Display server status. Operands must be names of servers.

2.59.6.6 Job, Queue, and Server Status Options

-f [-w]
Full display. Job, subjob, queue, or server attributes displayed one to a line.

JSON output:
PBS reports resources_used values for resources that are created or set in a hook as JSON strings in the
output of qstat -f.

If MoM returns a JSON object (a Python dictionary), PBS reports the value as a string in single quotes:
resources_used.<resource_name> = '{ <MoM JSON item value>, <MoM JSON item value>, <MoM JSON item
value>, ..}'

Example: MoM returns { "a":1, "b":2, "c":1,"d": 4} for resources_used.foo_str. We get:
resources_used.foo_str='{"a": 1, "b": 2, "c":1,"d": 4}'

If MoM returns a value that is not a valid JSON object, the value is reported verbatim.

Example: MoM returns "hello" for resources_used.foo_str. We get:
resources_used.foo_str="hello"

Optional -w prints each attribute on one unbroken line. Feed characters are converted:
• Newline is converted to backslash concatenated with “n”, resulting in “\n”
• Form feed is converted to backslash concatenated with “f”, resulting in “\f”

This -w is independent of the -w job alternate format output option.
PBS Commands

-F dsv [-D <delimiter>]
Prints output in delimiter-separated value format. The default *delimiter* is a pipe ("|"). You can specify a character or a string *delimiter* using the -D argument to the -F dsv option. For example, to use a comma as the delimiter:

```
qstat -f -F dsv -D,
```

If the delimiter itself appears in a value, it is escaped:

- On Linux, the delimiter is escaped with a backslash ("\").
- On Windows, the delimiter is escaped with a caret ("^\").

Feed characters are converted:

- Newline is converted to backslash concatenated with “n”, resulting in “\n”
- Form feed is converted to backslash concatenated with “f”, resulting in “\f”

A newline separates each job from the next. Using newline as the delimiter leads to undefined behavior.

Example of getting output in delimiter-separated value format:

```
qstat -f -F dsv
```

Job Id: 1.vbox|Job_Name = STDIN|Job_Owner = root@vbox|job_state = Q|queue = workq|server = vbox|Checkpoint = u|ctime = Fri Nov 11 17:57:05 2016|Error_Path = ...

-F json
Prints output in JSON format (http://www.json.org/).
Attribute output is preceded by timestamp, PBS version, and PBS server hostname.

Example:

```
qstat -F json
```

```json
{
  "timestamp":1479277336,
  "pbs_version":"14.1",
  "pbs_server":"vbox",
  "Jobs":{
    "1.vbox":{
      "Job_Name":"STDIN",
      "Job_Owner":"root@vbox",
      "job_state":"Q",
      ...
    }
  }
}
```

-G
Shows size in gigabytes. Triggers alternate format.

-M
Shows size in megawords. A word is considered to be 8 bytes. Triggers alternate format.

2.59.6.7 Version Information

--version
The qstat command returns its PBS version information and exits. This option can only be used alone.
2.59.7 Operands

2.59.7.1 Job Identifier Operands

The job ID is assigned by PBS at submission. Job IDs are used only with job status requests. Status information for specified job(s) is displayed. Formats:

Job ID:
<sequence number> [.<server name>] [@<server name>]

Job array ID:
<sequence number> [] [.<server name>] [@<server name>]

Subjob ID:
<sequence number> [<index>] [.<server name>] [@<server name>]

Range of subjobs:
<sequence number> [<index start> - <index end>] [.<server name>] [@<server name>]

Note that some shells require that you enclose a job array identifier in double quotes.

2.59.7.2 Destination Operands

Name of queue, name of server, or name of queue at a specific server. Formats:

queue name
  Specifies name of queue for job or queue display.
  • When displaying job status, PBS displays status for all jobs in the specified queue at the default server.
  • When displaying queue status, PBS displays status for the specified queue at the default server.

queue name@server name
  Specifies name of queue at server for job or queue display.
  • When displaying job status, PBS displays status for all jobs in the specified queue at the specified server.
  • When displaying queue status, PBS displays status for the specified queue at the specified server.

@server name
  Specifies server name for job or queue display.
  • When displaying job status, PBS displays status for all jobs at all queues at the specified server.
  • When displaying queue status, PBS displays status for all queues at the specified server.

server name
  Specifies server name for server display.
  • When displaying server status (with the -B option) PBS displays status for the specified server.

2.59.8 Standard Error

The qstat command writes a diagnostic message to standard error for each error occurrence.

2.59.9 Exit Status

Zero
  Upon successful processing of all operands
Greater than zero
  If any operands could not be processed

2.59.10 See Also

2.60 qstop

Prevents PBS jobs in the specified queue from being scheduled or routed

2.60.1 Synopsis

qstop <destination> [＜destination＞ ...]
qstop --version

2.60.2 Description

If destination is an execution queue, the qstop command stops a scheduler from scheduling jobs residing in destination. If destination is a routing queue, the server stops routing jobs from that queue. Sets the value of the queue’s started attribute to False.

2.60.2.1 Required Privilege

You must have PBS Operator or Manager privilege to run this command.

2.60.3 Options

   --version

The qstop command returns its PBS version information and exits. This option can only be used alone

2.60.4 Operands

The qstop command accepts one or more space-separated destination operands. The operands take one of three forms:

<queue name>
   Stops scheduling or routing from the specified queue.
@<server name>
   Stops scheduling or routing from all queues at the specified server.
<queue name>@<server name>
   Stops scheduling or routing from the specified queue at the specified server.

To stop scheduling at all queues at the default server, use the qmgr command:

    Qmgr: set queue @default started=false

2.60.5 Standard Error

The qstop command writes a diagnostic message to standard error for each error occurrence.

2.60.6 Exit Status

Zero

Upon successful processing of all operands presented to the qstop command
Greater than zero

If the `qstop` command fails to process any operand

## 2.60.7 See Also

The PBS Professional Administrator’s Guide, "`qmgr` on page 146, "`qstart` on page 190"
2.61 qsub

Submits a job to PBS

2.61.1 Synopsis

```
```

```
qsub --version
```

2.61.2 Description

You use the `qsub` command to submit a batch job to PBS. Submitting a PBS job specifies a task, requests resources, and sets job attributes.

The `qsub` command can read from a job script, from standard input, or from the command line. When the user has submitted the job, PBS returns the job identifier for that job. For a job, this is of the form:

```
<sequence number>.<server name>
```

For an array job, this is of the form:

```
<sequence number>[:].<server name>
```

During execution, jobs can be interactive or non-interactive. Interactive jobs are not rerunnable, and if they are blocking, you cannot use their exit status.

Jobs are run as the user and group who submitted the job.

2.61.2.1 Background Process

By default, on the first invocation, `qsub` spawns a background process to manage communication with the PBS server. Later invocations of `qsub` attempt to communicate with this background process. Under certain circumstances, calls to `qsub` when it uses the background process can result in communication problems. You can prevent `qsub` from spawning a background process by using the `-f` option, although this can degrade performance.

2.61.2.2 Where PBS Puts Job Files

By default, PBS copies the `stdout` and `stderr` files from the job back to the current working directory where the `qsub` command is executed. However, you can specify the output paths using the `-o` and `-e` options. You can also specify which and whether these files should be kept on the execution host via the `-k` option, or deleted, using the `-R` option.

See the `-k`, `-o`, `-e`, and `-R` options, and "Managing Output and Error Files", on page 42 of the PBS Professional User’s Guide.

2.61.2.3 Submitting Jobs By Using Job Scripts

To submit a PBS job by using a script, you specify a job script on the command line:

```
qsub [<options>] <script name>
```
Job scripts are run as the user and group who submitted the job. Job scripts can be written in Python, Linux shells such as `csh` and `sh`, the Windows command batch language, Perl, etc.

A PBS job script consists of the following:

- Optional shell specification
- Any PBS directives
- The user’s tasks: programs, commands, or applications
- Optional comments

Under Windows, comments can contain only ASCII characters. See the PBS Professional User’s Guide.

### 2.61.2.3.i Using Shells and Interpreters

By default, PBS uses your login shell to run your script. You can optionally specify a different shell or interpreter to run your script:

- Via the `-S` option to `qsub`:
  ```bash
  qsub -S <path to shell> <script name>
  ```
  For example:
  ```bash
  qsub -S /bin/bash myscript.sh
  ```
- In the first line of your script. For example:
  ```bash
  cat myscript.sh
  #!/bin/sh
  #PBS -N MyHelloJob
  print "Hello"
  ```

### 2.61.2.3.ii Python Job Scripts

You can use the same Python script under Linux or under Windows, if the script is written to be portable. PBS includes a Python package, allowing Python job scripts to run; you do not need to install Python. You can include PBS directives in a Python job script as you would in a Linux shell script. Python job scripts can access Win32 APIs, including the following modules:

- `Win32api`
- `Win32con`
- `Pywintypes`

Example 2-25: We have a Python job script that includes PBS directives:

```python
#!/usr/bin/python
#PBS -l select=1:ncpus=3:mem=1gb
#PBS -N HelloJob
print "Hello"
```

To run a Python job script under Linux, use the Python path on the execution host:

```bash
qsub -S <Python path on execution host> <script name>
```

For example,

```bash
qsub -S /PBS_EXEC/bin/pbs_python /usr/bin/python
```

To run a Python job script under Windows, use the Python path on the execution host:
 PBS Commands

Chapter 2

2.61.2.3.i Linux Shell Job Scripts

Example 2-26: We have a Linux job script named "weatherscript" for a job named "Weather1" which runs the executable "weathersim" on Linux:

```bash
#!/bin/sh
#PBS -N Weather1
#PBS -l walltime=1:00:00
/weusr/local/weathersim
```

To submit the job, the user types:

```bash
qsub weatherscript <return>
```

2.61.2.3.iv Windows Command Job Scripts

Example 2-27: We have a script named "weather.exe" for a job named "Weather1" which runs under Windows:

```bash
#PBS -N Weather1
#PBS -l walltime=1:00:00
weathersim.exe
```

To submit the job, the user types:

```bash
qsub weather.exe <return>
```

In Windows, if you use notepad to create a job script, the last line does not automatically get newline-terminated. Be sure to put one explicitly, otherwise, PBS job will get the following error message:

```
More?
```

when the Windows command interpreter tries to execute that last line.

2.61.2.4 Submitting Jobs From Standard Input

To submit a PBS job by typing job specifications at the command line, you type:

```bash
qsub [options] [-] <return>
```

then type any directives, then any tasks, followed by:

- Linux: CTRL-D on a line by itself
- Windows: CTRL-Z <return>

to terminate the input.

The qsub command behaves the same both with and without the dash operand.

For example, on Linux:

```bash
qsub <return>
#PBS -N StdInJob
sleep 100
<CTRL-D>
```
2.61.2.5  Submitting Job Directly by Specifying Executable on Command Line

To submit a job directly, you specify the executable on the command line:

```
qsub [<options>] -- <executable> [ <arguments to executable> ] <return>
```

When you run `qsub` this way, it runs the `executable` directly. It does not start a shell, so no shell initialization scripts are run, and execution paths and other environment variables are not set. There is not an easy way to run your command in a different directory. You should make sure that environment variables are set correctly, and you will usually have to specify the full path to the command.

Example 2-28: To run `myprog` with the arguments `a` and `b`:

```
qsub -- myprog a b <return>
```

Example 2-29: To run `myprog` with the arguments `a` and `b`, naming the job “JobA”:

```
qsub -N JobA -- myprog a b <return>
```

2.61.2.6  Requesting Resources and Placing Jobs

Requesting resources includes setting limits on resource usage and controlling how the job is placed on vnodes.

Resources are requested by using the `-l` option, either in job-wide requests using `<resource name>=<value>` pairs, or in chunks inside of selection statements. See Chapter 5, “List of Built-in Resources”, on page 277.

Job-wide `<resource name>=<value>` requests are of the form:

```
-l <resource name>=<value> [...]
```

The selection statement is of the form:

```
-l select=[<N>:]<chunk> [+[<N>:]<chunk> ...]
```

where `N` specifies how many of that chunk, and a `chunk` is of the form:

```
<resource name>=<value>[:<resource name>=<value> ...]
```

You choose how your chunks are placed using the `place statement`. The `place statement` can contain the following elements, in any order:

```
-l place=[<arrangement>][: <sharing> [: <grouping>]]
```

where

- **arrangement**
  - Whether this chunk is willing to share this vnode or host with other chunks from the same job. One of **free** | **pack** | **scatter** | **vscatter**

- **sharing**
  - Whether this this chunk is willing to share this vnode or host with other jobs. One of **excl** | **shared** | **exclhost**

- **grouping**
  - Whether the chunks from this job should be placed on vnodes that all have the same value for a resource. Can have only one instance of `group=<resource name>`

- **free**
  - Place job on any vnode(s).

- **pack**
  - All chunks are taken from one host.
scatter
Only one chunk with any MPI processes is taken from a host. A chunk with no MPI processes may be taken from the same vnode as another chunk.

vscatter
Only one chunk is taken from any vnode. Each chunk must fit on a vnode.

excl
Only this job uses the vnodes chosen.

shared
This job can share the vnodes chosen.

exclhost
The entire host is allocated to the job.

group=<resource name>
Chunks are grouped according to a resource. All vnodes in the group must have a common value for resource, which can be either the built-in resource host or a custom vnode-level resource.

Resource name must be a string or a string array.

The place statement cannot begin with a colon. Colons are delimiters; use them only to separate parts of a place statement, unless they are quoted inside resource values.

Note that vnodes can have sharing attributes that override job placement requests. See section 6.10, “Vnode Attributes”, on page 333.

For more on resources, resource requests, usage limits, and job placement, see "Using PBS Resources" on page 225 in the PBS Professional Administrator’s Guide and "Allocating Resources & Placing Jobs", on page 49 of the PBS Professional User’s Guide.

2.61.2.6.i Caveats for Requesting Resources
Do not mix old-style resource or vnode specifications with the new select and place statements. Do not use one in a job script and the other on the command line. Mixing the two will result in an error.

You cannot submit a job requesting a custom resource which has been created to be invisible or read-only for unprivileged users, regardless of your privilege. A Manager or Operator can use the qalter command to change a job’s request for this kind of custom resource.

2.61.2.7 Setting Attributes
The job submitter sets job attributes by giving options to the qsub command or by using PBS directives. Most qsub options set a job attribute, and have a corresponding PBS directive with the same syntax as the option. Attributes set via command-line options take precedence over those set using PBS directives. See the PBS Professional User’s Guide, or section 6.11, “Job Attributes”, on page 341.

2.61.2.8 Changing qsub Behavior
The behavior of the qsub command may be affected by the server’s default_qsub_arguments attribute. This attribute can set the default for any job attribute. The default_qsub_arguments server attribute is settable by the administrator, and is overridden by command-line arguments and script directives. See section 6.6, “Server Attributes”, on page 295.

The behavior of the qsub command may also be affected by any site hooks. Site hooks can modify the job’s attributes, change its routing, etc.
2.61.3 Options to qsub

-a <date and time>
Point in time after which the job is eligible for execution. Given in pairs of digits. Sets job’s Execution_Time attribute to date and time.
Format: datetime, expressed as [[[CC]YY][MM]DD]hhmm[.SS]
where CC is the century, YY is the year, MM is the month, DD is the day of the month, hh is the hour, mm is the minute, and SS is the seconds.
Each portion of the date defaults to the current date, as long as the next-smaller portion is in the future. For example, if today is the 3rd of the month and the specified day DD is the 5th, the month MM is set to the current month.
If a specified portion has already passed, the next-larger portion is set to one after the current date. For example, if the day DD is not specified, but the hour hh is specified to be 10:00 a.m. and the current time is 11:00 a.m., the day DD is set to tomorrow.

-A <account string>
Accounting string associated with the job. Used for labeling accounting data. Sets job’s Account_Name attribute to account string.
Format: String

-c <checkpoint spec>
Determines when the job will be checkpointed. Sets job’s Checkpoint attribute to checkpoint spec. An $action script is required to checkpoint the job.
The argument checkpoint spec can take one of the following values:

  c
  Checkpoint at intervals, measured in CPU time, set on job’s execution queue. If there is no interval set at the queue, the job is not checkpointed

  c=<minutes of CPU time>
  Checkpoint at intervals of specified number of minutes of job CPU time. This value must be greater than zero. If the interval specified is less than that set on the job’s execution queue, the queue's interval is used.
  Format: Integer

  w
  Checkpoint at intervals, measured in walltime, set on job’s execution queue. If there is no interval set at the queue, the job is not checkpointed.

  w=<minutes of walltime>
  Checkpoint at intervals of the specified number of minutes of job walltime. This value must be greater than zero. If the interval specified is less than that set on the job’s execution queue, the queue's interval is used.
  Format: Integer

  n
  No checkpointing.

  s
  Checkpoint only when the server is shut down.

  u
  Unset. Defaults to behavior when interval argument is set to s.
Default: u
Format: String
-C <directive prefix>
Defines the prefix identifying a PBS directive. Default prefix is “#PBS”.
If the directive prefix argument is a null string, qsub does not scan the script file for directives. Overrides the PBS_DPREFIX environment variable and the default. The string "PBS_DPREFIX" cannot be used as a PBS directive. Length limit: 4096 characters.

-e <path>
Path to be used for the job’s standard error stream. Sets job’s Error_Path attribute to path. The path argument is of the form:

[<hostname>]:<path>
The path is interpreted as follows:

path
If path is relative, it is taken to be relative to the current working directory of the qsub command, where it is executing on the current host.
If path is absolute, it is taken to be an absolute path on the current host where the qsub command is executing.

hostname:path
If path is relative, it is taken to be relative to the user’s home directory on the host named hostname.
If path is absolute, it is an absolute path on the host named hostname.
If path does not include a filename, the default filename has the form <job ID>.ER
If the -e option is not specified, PBS copies the standard error to the current working directory where the qsub command was executed, and writes standard error to the default filename, which has this form:

<job name>.e<sequence number>
If you use a UNC path for output or error files, the hostname is optional. If you use a non-UNC path, the hostname is required.
This option is overridden by the -k option.

-f
Prevents qsub from spawning a background process. By default, qsub spawns a background process to manage communication with the PBS server. When this option is specified, the qsub process connects directly to the server and no background process is created.
NOTE: Use of this option degrades performance of qsub when calls to qsub are made in rapid succession.

-G [<path to GUI application or script>]
Starts a GUI session. When no application or script is provided, starts a GUI-enabled interactive shell. When an application or script is provided, starts the GUI application or script. Use full path to application or script unless the path is part of the user’s PATH environment variable on the execution host. When submission and execution hosts are different, this uses a remote viewer.
Session is terminated when remote viewer, GUI application, or interactive shell is terminated, or when job is deleted.
Can be used only with interactive jobs (the -I option).
Available only under Windows.

-h
Applies a User hold to the job. Sets the job’s Hold_Types attribute to “u”.

-I
Job is to be run interactively. Sets job’s interactive attribute to True. The job is queued and scheduled as any PBS batch job, but when executed, the standard input, output, and error streams of the job are connected to the terminal session in which qsub is running. If a job script is given, only its directives are processed. When the job begins execution, all input to the job is taken from the terminal session. See the PBS Professional User’s Guide for additional information on interactive jobs.
Interactive jobs are not rerunnable.
Job arrays cannot be interactive.
When used with -Wblock=true, no exit status is returned.

-j <join>
Specifies whether and how to join the job’s standard error and standard output streams. Sets job’s Join_Path attribute to join.
Default: n; not merged
The join argument can take the following values:

<table>
<thead>
<tr>
<th>Suboption</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>oe</td>
<td>Standard error and standard output are merged into standard output.</td>
</tr>
<tr>
<td>eo</td>
<td>Standard error and standard output are merged into standard error.</td>
</tr>
<tr>
<td>n</td>
<td>Standard error and standard output are not merged.</td>
</tr>
</tbody>
</table>

-J <range>
Makes this job an array job. Sets job’s array attribute to True. Use the range argument to specify the indices of the subjobs of the array. range is specified in the form X-Y[Z] where X is the first index, Y is the upper bound on the indices, and Z is the stepping factor. For example, 2-7:2 will produce indices of 2, 4, and 6. If Z is not specified, it is taken to be 1. Indices must be greater than or equal to zero.
Job arrays are always rerunnable.
-k <discard>
  Specifies whether and which of the standard output and standard error streams is left behind on the execution host, or written to their final destination. Sets the job’s Keep_Files attribute to discard. Overrides default path names for these streams. Overrides -o and -e options.

  Default: n; neither is retained, and files are not written directly to final destinations.

  In the case where output and/or error is retained on the execution host in a job-specific staging and execution directory created by PBS, these files are deleted when PBS deletes the directory.

  The discard argument can take the following values:

  **Table 2-34: Sub-options to discard Option**

<table>
<thead>
<tr>
<th>Suboption</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>The standard error stream is retained on the execution host, in the job’s staging and execution directory. The filename is &lt;job name&gt;.e&lt;sequence number&gt;</td>
</tr>
<tr>
<td>o</td>
<td>The standard output stream is retained on the execution host, in the job’s staging and execution directory. The filename is &lt;job name&gt;.o&lt;sequence number&gt;</td>
</tr>
<tr>
<td>eo, oe</td>
<td>Both standard output and standard error streams are retained on the execution host, in the job’s staging and execution directory.</td>
</tr>
<tr>
<td>d</td>
<td>Output and/or error are written directly to their final destination. Overrides action of leaving files on execution host.</td>
</tr>
<tr>
<td>n</td>
<td>Neither stream is retained.</td>
</tr>
</tbody>
</table>

-| <resource list>
  Allows the user to request resources and specify job placement. Sets job’s Resource_list attribute to resource list. Requesting a resource places a limit on its usage.

  For how to request resources and place jobs, see section 2.61.2.6, “Requesting Resources and Placing Jobs”, on page 210.
-m <mail events>
   Specifies the set of conditions under which mail about the job is sent. Sets job’s Mail_Points attribute to mail events. The mail events argument can be one of the following:
   - The single character “n”
   - Any combination of “a”, “b”, and “e”, with optional “j”
   The following table lists the sub-options to the -m option:

### Table 2-35: Sub-options to m Option

<table>
<thead>
<tr>
<th>Suboption</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>No mail is sent.</td>
</tr>
<tr>
<td>a</td>
<td>Mail is sent when the job is aborted by PBS.</td>
</tr>
<tr>
<td>b</td>
<td>Mail is sent when the job begins execution.</td>
</tr>
<tr>
<td>e</td>
<td>Mail is sent when the job terminates.</td>
</tr>
<tr>
<td>j</td>
<td>Mail is sent for subjobs. Must be combined with one or more of a, b, or e options</td>
</tr>
</tbody>
</table>

Format: String
Syntax: n | [j](one or more of a, b, e)
Example: -m ja
Default value: a

-M <user list>
List of users to whom mail about the job is sent. Sets job’s Mail_Users attribute to user list.
The user list argument has the form:
<username>[@<hostname>],<username>[@<hostname>],...
Default: Job owner

-N <name>
Sets job’s Job_Name attribute and name to name.
Format: Job Name; see "Job Name, Job Array Name" on page 365
Default: if a script is used to submit the job, the job’s name is the name of the script. If no script is used, the job’s name is “STDIN”.

-o <path>
Path to be used for the job’s standard output stream. Sets job’s Output_Path attribute to path. The path argument has the form:
[<hostname>]:<path>
The path is interpreted as follows:
- path
  - If path is relative, it is taken to be relative to the current working directory of the command, where it is executing on the current host.
  - If path is absolute, it is taken to be an absolute path on the current host where the command is executing.
  - hostname:path
    - If path is relative, it is taken to be relative to the user’s home directory on the host named hostname.
If `path` is absolute, it is an absolute path on the host named `hostname`.

If `path` does not include a filename, the default filename has the form `<job ID>.OU`

If the `-o` option is not specified, PBS copies the standard output to the current working directory where the `qsub` command was executed, and writes standard output to the default filename, which has this form: `<job name>..o<sequence number>`

If you use a UNC path, the hostname is optional. If you use a non-UNC path, the hostname is required.

This option is overridden by the `-k` option.

- **-p <priority>**
  Priority of the job. Sets job's `Priority` attribute to `priority`.
  Format: Host-dependent integer
  Range: [-1024, +1023] inclusive
  Default: Zero

- **-P <project>**
  Specifies a project for the job. Sets job's `project` attribute to `project`.
  Format: Project Name; see "Project Name" on page 367
  Default value: "_pbs_project_default".

- **-q <destination>**
  Where the job is sent upon submission.
  Specifies a queue, a server, or a queue at a server. The destination argument can have one of these formats:
  `<queue name>`
  Job is submitted to the specified queue at the default server.
  `@<server name>`
  Job is submitted to the default queue at the specified server.
  `<queue name>@<server name>`
  Job is submitted to the specified queue at the specified server.
  Default: Default queue at default server

- **-r <y|n>**
  Declares whether the job is rerunnable. Sets job's `Rerunnable` attribute to the argument value. Does not affect how the job is handled in the case where the job was unable to begin execution.
  Format: Single character, "y" or "n"

### Table 2-36: Sub-options to r Option

<table>
<thead>
<tr>
<th>Suboption</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>y</code></td>
<td>Job is rerunnable.</td>
</tr>
<tr>
<td><code>n</code></td>
<td>Job is not rerunnable.</td>
</tr>
</tbody>
</table>

Default: “y”

Interactive jobs are not rerunnable. Job arrays are always rerunnable. See "qrerun" on page 173.
-R <remove options>
Specifies whether standard output and/or standard error files are automatically removed (deleted) upon job completion.
Sets the job’s Remove_Files attribute to remove options. Overrides default path names for these streams.
Overrides -o and -e options.
This attribute cannot be altered once the job has begun execution.
Default: Unset; neither is removed
The remove options argument can take the following values:

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>The standard error stream is removed (deleted) upon job completion</td>
</tr>
<tr>
<td>o</td>
<td>The standard output stream is removed (deleted) upon job completion</td>
</tr>
<tr>
<td>eo, oe</td>
<td>Both standard output and standard error streams are removed (deleted) upon job completion</td>
</tr>
<tr>
<td>unset</td>
<td>Neither stream is removed.</td>
</tr>
</tbody>
</table>

-S <path list>
Specifies the interpreter or shell path for the job script. Sets job’s Shell_Path_List attribute to path list.
The path list argument is the full path to the interpreter or shell including the executable name.
Only one path may be specified without a hostname. Only one path may be specified per named host. The path selected is the one whose hostname is that of the server on which the job resides.
Format: <path>[@<hostname>][,<path>@<hostname> ...]
Default: User’s login shell on execution host
Example of using bash via a directive:

```
#PBS -S /bin/bash@mars,/usr/bin/bash@jupiter
```
Example of running a Python script from the command line on Linux:

```
qsub -S $PBS_EXEC/bin/pbs_python <script name>
```
Example of running a Python script from the command line on Windows:

```
qsub -S %PBS_EXEC%in\pbs_python.exe <script name>
```
-u <user list>
List of usernames. Job is run under a username from this list. Sets job’s User_List attribute to user list.
Only one username may be specified without a hostname. Only one username may be specified per named host.
The server on which the job resides will select first the username whose hostname is the same as the server name. Failing that, the next selection is the username with no specified hostname. The usernames on the server and execution hosts must be the same. The job owner must have authorization to run as the specified user.
Format of user list: <username>[@<hostname>][,<username>@<hostname> ...]
Default: Job owner (username on submission host)
-v <variable list>

Specifies environment variables and shell functions to be exported to the job. This is the list of environment variables which is added to those already automatically exported. These variables exist in the user’s login environment, from which qsub is run. The job’s Variable_List attribute is appended with the variables in variable list and their values. See section 2.61.7, “Environment Variables”, on page 223.

Format: comma-separated list of strings in the form:

<variable>
or
<variable>=<value>

If a <variable>=<value> pair contains any commas, the value must be enclosed in single or double quotes, and the <variable>=<value> pair must be enclosed in the kind of quotes not used to enclose the value. For example:

qsub -v “var1=A,B,C,D” job.sh
qsub -v a=10, “var2=A,B”, c=20, HOME=/home/zzz job.sh

Default: No environment variables are added to job’s variable list.

-V

All environment variables and shell functions in the user’s login environment where qsub is run are exported to the job. The job’s Variable_List attribute is appended with all of these environment variables and their values.

-W <additional attributes>

The -W option allows specification of some job attributes. Some job attributes must be specified using this option. Those attributes are listed below. Format:

-W <attribute name>=<value>[,<attribute name>=<value>...]

If white space occurs within the additional attributes argument, or the equal sign “=” occurs within a value string, it must be enclosed with single quotes or double quotes.

The following attributes can be set using the -W option only:

block=true

The qsub command waits for the job to terminate, then returns the job’s exit value. Sets job’s block attribute to True. When used with X11 forwarding or interactive jobs, no exit value is returned. See section 2.61.8, “Exit Status”, on page 224.
depend=<dependency list>
Defines dependencies between this and other jobs. Sets the job’s depend attribute to dependency list. The dependency list has the form:

\(<\text{type}}>:.\langle\text{arg list}\rangle[,\langle\text{type}>:.\langle\text{arg list}\rangle\ldots]\)

where except for the on type, the arg list is one or more PBS job IDs, and has the form:

\(<\text{job ID}>:.\langle\text{job ID}\rangle\ldots]\)

The type can be:

after: \(<\text{arg list}\rangle\)
This job may be scheduled for execution at any point after all jobs in arg list have started execution.

afterok: \(<\text{arg list}\rangle\)
This job may be scheduled for execution only after all jobs in arg list have terminated with no errors. See section 2.61.8.1, “Warning About Exit Status with csh”, on page 225.

afternotok: \(<\text{arg list}\rangle\)
This job may be scheduled for execution only after all jobs in arg list have terminated with errors. See section 2.61.8.1, “Warning About Exit Status with csh”, on page 225.

afterany: \(<\text{arg list}\rangle\)
This job may be scheduled for execution after all jobs in arg list have finished execution, with any exit status (with or without errors.) This job will not run if a job in the arg list was deleted without ever having been run.

before: \(<\text{arg list}\rangle\)
Jobs in arg list may begin execution once this job has begun execution.

beforeok: \(<\text{arg list}\rangle\)
Jobs in arg list may begin execution once this job terminates without errors. See section 2.61.8.1, “Warning About Exit Status with csh”, on page 225.

beforenotok: \(<\text{arg list}\rangle\)
If this job terminates execution with errors, jobs in arg list may begin. See section 2.61.8.1, “Warning About Exit Status with csh”, on page 225.

beforeany: \(<\text{arg list}\rangle\)
Jobs in arg list may begin execution once this job terminates execution, with or without errors.

on: \(<\text{count}\rangle\)
This job may be scheduled for execution after count dependencies on other jobs have been satisfied. This type is used in conjunction with one of the before types listed. count is an integer greater than 0. Job IDs in the arg list of before types must have been submitted with a type of on.

To use the before types, the user must have the authority to alter the jobs in arg list. Otherwise, the dependency is rejected and the new job aborted.

Error processing of the existence, state, or condition of the job on which the newly submitted job is performed after the job is queued. If an error is detected, the new job is deleted by the server. Mail is sent to the job submitter stating the error.

Dependency examples:

```bash
qsub -W depend=afterok:123.host1.domain.com /tmp/script
qsub -W depend=before:234.host1.com:235.host1.com /tmp/script
```

group_list=<group list>
List of group names. Job is run under a group name from this list. Sets job’s group_List attribute to group list.
Only one group name may be specified without a hostname. Only one group name may be specified per named host. The server on which the job resides will select first the group name whose hostname is the same as the server name. Failing that, the next selection is the group name with no specified hostname. The group names on the server and execution hosts must be the same. The job submitter’s primary group is automatically added to the list.

Under Windows, the primary group is the first group found for the user by PBS when it queries the accounts database.

Format of `group list`: `<group name>[@<hostname>],<group name>@<hostname> ...`

Default: Login group name of job owner

```
pwd
pwd=
pwd=
```

These forms prompt the user for a password. A space between `W` and `pwd` is optional. Spaces between the quotes are optional. Examples:

```
qsub ... -Wpwd <return>
qsub ... -W pwd='' <return>
qsub ... -W pwd=" " <return>
```

Available on Windows and supported Linux x86 and x86_64 platforms only.

```
release_nodes_on_stageout=<value>
```

When set to `True`, all of the job’s vnodes not on the primary execution host are released when stageout begins.

Cannot be used with vnodes managed by cpuset MoMs, (whose arch is linux_cpuset), or with vnodes tied to Cray X* series systems.

When cgroups is enabled and this is used with some but not all vnodes from one MoM, resources on those vnodes that are part of a cgroup are not released until the entire cgroup is released.

The job’s `stageout` attribute must be set for the `release_nodes_on_stageout` attribute to take effect.

Format: `Boolean`

Default: `False`

```
run_count=<value>
```

Sets the number of times the server thinks it has run the job. Sets the value of the job’s `run_count` attribute to `value`.

Format: Integer greater than or equal to zero

```
sandbox=<sandbox spec>
```

Determines which directory PBS uses for the job’s staging and execution. Sets job’s `sandbox` attribute to the value of `sandbox spec`.

Allowed values for `sandbox spec`:

- **PRIVATE**
  - PBS creates a job-specific directory for staging and execution.

- **HOME** or unset
  - PBS uses the user’s home directory for staging and execution.

Format: `String`
stagein=<path list>
stageout=<path list>

Specifies files or directories to be staged in before execution or staged out after execution is complete. Sets the job’s stagein and stageout attributes to the specified path lists. On completion of the job, all staged-in and staged-out files and directories are removed from the execution host(s). The path list has the form:

<file spec>[,<file spec>]

where <file spec> is

<execution path>@<hostname>:<storage path>

regardless of the direction of the copy. The name execution path is the name of the file or directory on the primary execution host. It can be relative to the staging and execution directory on the execution host, or it can be an absolute path.

The “@” character separates execution path from storage path.

The name storage path is the path on hostname. The name can be relative to the staging and execution directory on the execution host, or it can be an absolute path.

If path list has more than one file spec, i.e. it contains commas, it must be enclosed in double quotes.

If you use a UNC path, the hostname is optional. If you use a non-UNC path, the hostname is required.

umask=<mask value>

The umask with which the job is started. Sets job’s umask attribute to mask value. Controls umask of job’s standard output and standard error.

The following example allows group and world read of the job’s output and error:

-W umask=33

Format: one to four digits; typically two

Default: 077

-X

Allows user to receive X output from interactive job.

DISPLAY variable in submission environment must be set to desired display.

Can be used only with interactive jobs: must be used with one of the following:

-I

-W interactive=true (deprecated)

Cannot be used with -v DISPLAY.

When used with -W block=true, no exit status is returned.

Can be used with -V option.

Not available under Windows.

-Z

Job identifier is not written to standard output.

--version

The qsub command returns its PBS version information and exits. This option can only be used alone.

2.61.4 Operands

The qsub command accepts as operands one of the following:

(no operands)

Same as with a dash. Any PBS directives and user tasks are read from the command line.
<script>
Path to script. Can be absolute or relative to current directory where qsub is run.

- When you use a dash, any PBS directives and user tasks are read from the command line.

-- <executable> [<arguments to executable>]
A single executable (preceded by two dashes) and its arguments
The executable, and any arguments to the executable, are given on the qsub command line. The executable is preceded by two dashes, “--”.
If a script or executable is specified, it must be the last argument to qsub. The arguments to an executable must follow the name of the executable.
When you run qsub this way, it runs the executable directly. It does not start a shell, so no shell initialization scripts are run, and execution paths and other environment variables are not set. You should make sure that environment variables are set correctly.

2.61.5 Standard Output
Job ID for submitted job
If the job is successfully created
(No output)
If the -z option is set

2.61.6 Standard Error
The qsub command writes a diagnostic message to standard error for each error occurrence.

2.61.7 Environment Variables
The qsub command uses the following environment variables:

PBS_DEFAULT
Name of default server.

PBS_DPREFIX
Prefix string which identifies PBS directives.

Environment variables beginning with “PBS_O_” are created by qsub. PBS automatically exports the following environment variables to the job, and the job’s Variable_List attribute is set to this list:

PBS_ENVIRONMENT
Set to PBS_BATCH for a batch job. Set to PBS_INTERACTIVE for an interactive job. Created when qsub is run.

PBS_JOBDIR
Pathname of job’s staging and execution directory on the primary execution host.

PBS_JOBID
Job identifier given by PBS when the job is submitted. Created when qsub is run.

PBS_JOBNAME
Job name specified by submitter. Created when qsub is run.
PBS_NODEFILE
   Name of file containing the list of vnodes assigned to the job. Created when qsub is run.

PBS_O_HOME
   User’s home directory. Value of HOME taken from user’s submission environment.

PBS_O_HOST
   Name of submit host. Value taken from user’s submission environment.

PBS_O_LANG
   Value of LANG taken from user’s submission environment.

PBS_O_LOGNAME
   User’s login name. Value of LOGNAME taken from user’s submission environment.

PBS_O_MAIL
   Value of MAIL taken from user’s submission environment.

PBS_O_PATH
   User’s PATH. Value of PATH taken from user’s submission environment.

PBS_O_QUEUE
   Name of the queue to which the job was submitted. Value is taken from job submission, otherwise default queue.

PBS_O_SHELL
   Value taken from user’s submission environment.

PBS_O_SYSTEM
   Operating system, from uname -s, on submit host. Value taken from user’s submission environment.

PBS_O_TZ
   Timezone. Value taken from user’s submission environment.

PBS_O_WORKDIR
   Absolute path to directory where qsub is run. Value taken from user’s submission environment.

PBS_QUEUE
   Name of the queue from which the job is executed. Created when qsub is run.

PBS_TMPDIR
   Pathname of job’s scratch directory. Set when PBS assigns it.

## 2.61.8 Exit Status

For non-blocking jobs:
- **Zero**
  - Upon successful processing of input
- **Greater than zero**
  - Upon failure of qsub

For blocking jobs:
- **Exit value of job**
  - When job runs successfully
  - **3**
    - If the job is deleted without being run
2.61.8.1 Warning About Exit Status with csh

If a job is run in csh and a .logout file exists in the home directory in which the job executes, the exit status of the job is that of the .logout script, not the job script. This may impact any inter-job dependencies.

2.61.9 See Also

2.62 qterm

Terminates one or both PBS servers, and optionally terminates scheduler(s) and/or MoMs

2.62.1 Synopsis

qterm [-f | -F | -i] [-m] [-s] [-t <type>] [<server name> [<server name> ...]]
qterm --version

2.62.2 Description

The qterm command terminates a PBS batch server. Once the server is terminating, no new jobs are accepted by the server, and no jobs are allowed to begin execution. The impact on running jobs depends on the way the server is shut down.

The qterm command does not exit until the server has completed its shutdown procedure.

If the complex is configured for failover, and the primary server is shut down, the normal behavior for the secondary server is to become active. The qterm command provides options to manage the behavior of the secondary server; it can be shut down, forced to remain idle, or shut down in place of the primary server.

2.62.2.1 Required Privilege

In order to run the qterm command, you must have PBS Operator or Manager privilege.

2.62.3 Options to qterm

(no options)

The qterm command defaults to qterm -t quick.

-f
If the complex is configured for failover, shuts down both the primary and secondary servers.
Without the -f option, qterm shuts down the primary server and makes the secondary server active.
The -f option cannot be used with the -i or -F options.

-F
If the complex is configured for failover, shuts down only the secondary server, leaving the primary server active.
The -F option cannot be used with the -f or -i options.

-i
If the complex is configured for failover, leaves the secondary server idle when the primary server is shut down.
The -i option cannot be used with the -f or -F options.

-m
Shuts down the primary server and all MoMs (pbs_mom). This option does not cause jobs or subjobs to be killed. Jobs are left running subject to other options to the qterm command.

-s
Shuts down the primary server and the scheduler (pbs_sched).
-t <type>
immediate
Shuts down the primary server. Immediately stops all running jobs. Any running jobs that can be checkpointed are checkpointed, terminated, and requeued. Jobs that cannot be checkpointed are terminated and requeued if they are rerunnable, otherwise they are killed.

If any job cannot be terminated, for example the server cannot contact the MoM of a running job, the server continues to execute and the job is listed as running. The server can be terminated by a second qterm -t immediate command.

While terminating, the server is in the Terminating state.

delay
Shuts down the primary server. The server waits to terminate until all non-checkpointable, non-rerunnable jobs are finished executing. Any running jobs that can be checkpointed are checkpointed, terminated, and requeued. Jobs that cannot be checkpointed are terminated and requeued if they are rerunnable, otherwise they are allowed to continue to run.

While terminating, the server is in the Terminating-Delayed state.

quick
Shuts down the primary server. Running jobs and subjobs are left running.

This is the default behavior when no options are given to the qterm command.

While terminating, the server is in the Terminating state.

--version
The qterm command returns its PBS version information and exits. This option can only be used alone.

2.62.4 Operands

You optionally specify the list of servers to shut down using [<server name>[ <server name> ...]].

If you do not specify any servers, the qterm command shuts down the default server.

2.62.4.1 Standard Error

The qterm command writes a diagnostic message to standard error for each error occurrence.

2.62.4.2 Exit Status

Zero
Upon successful processing of all operands presented to the qterm command

Greater than zero
If the qterm command fails to process any operand

2.62.4.3 See Also

The PBS Professional Administrator's Guide, "pbs_server" on page 106, "pbs_sched" on page 104, "pbs_mom" on page 72
2.63 tracejob

Extracts and prints log messages for a PBS job

2.63.1 Synopsis

```
tracejob --version
```

2.63.2 Description

The tracejob command extracts log messages for a given job ID and prints them in chronological order.

The tracejob command extracts information from server, default scheduler, accounting, and MoM logs. Server logs contain information such as when a job was queued or modified. Scheduler logs contain clues as to why a job is not running. Accounting logs contain accounting records for when a job was queued, started, ended, or deleted. MoM logs contain information about what happened to a job while it was running.

To get MoM log messages for a job, tracejob must be run on the machine on which the job ran. If the job ran on multiple hosts, you must run tracejob on each of those hosts.

Some log messages appear many times. In order to make the output of tracejob more readable, messages that appear over a certain number of times (see option -c below) are restricted to only the most recent message.

2.63.3 Using tracejob on Job Arrays

If tracejob is run on a job array, the information returned is about the job array itself, and not its subjobs. Job arrays do not have associated MoM log messages. If tracejob is run on a subjob, the same types of log messages are available as for a job. Certain log messages that occur for a regular job will not occur for a subjob.

2.63.4 Required Privilege

All users have access to server, scheduler, and MoM information. Only Administrator or root can access accounting information.

2.63.5 Options to tracejob

- **-a**
  Do not report accounting information.

- **-c <count>**
  Set excessive message limit to count. If a message is logged at least count times, only the most recent message is printed.
  The default for count is 15.
-f <filter>
Do not include log events of type filter. The -f option can be used more than once on the command line. The following table shows each filter with its hex value and category:

### Table 2-38: tracejob Filters

<table>
<thead>
<tr>
<th>Filter</th>
<th>Hex Value</th>
<th>Message Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>0x0001</td>
<td>Internal errors</td>
</tr>
<tr>
<td>system</td>
<td>0x0002</td>
<td>System errors</td>
</tr>
<tr>
<td>admin</td>
<td>0x0004</td>
<td>Administrative events</td>
</tr>
<tr>
<td>job</td>
<td>0x0008</td>
<td>Job-related events</td>
</tr>
<tr>
<td>job_usage</td>
<td>0x0010</td>
<td>Job accounting info</td>
</tr>
<tr>
<td>security</td>
<td>0x0020</td>
<td>Security violations</td>
</tr>
<tr>
<td>sched</td>
<td>0x0040</td>
<td>Scheduler events</td>
</tr>
<tr>
<td>debug</td>
<td>0x0080</td>
<td>Common debug messages</td>
</tr>
<tr>
<td>debug2</td>
<td>0x0100</td>
<td>Uncommon debug messages</td>
</tr>
<tr>
<td>reserv</td>
<td>0x0200</td>
<td>Reservation debug messages</td>
</tr>
<tr>
<td>debug3</td>
<td>0x0400</td>
<td>Less common than debug2</td>
</tr>
<tr>
<td>debug4</td>
<td>0x0800</td>
<td>Less common than debug3</td>
</tr>
</tbody>
</table>

-li
Do not report scheduler information.

-m
Do not report MoM information.

-n <days>
Report information from up to days days in the past.
Default number of days: 1 = today

-p <path>
Use path as path to PBS_HOME on machine being queried.

-s
Do not report server information.

-w <cols>
Width of current terminal. If cols is not specified, tracejob queries OS to get terminal width. If OS doesn’t return anything, defaults to 80.

-v
Verbose. Report more of tracejob’s errors than default.

-z
Suppresses printing of duplicate messages.

--version
The tracejob command returns its PBS version information and exits. This option can only be used alone.
2.63.6 Operands

The `tracejob` command accepts one job ID operand.

For a job, this has the form:

`<sequence number>[.<server name>][@<server name>]`

For a job array, the form is:

`<sequence number>[][][.<server name>][@<server name>]`

For a subjob, the form is:

`<sequence number>[<index>][.<server name>][@<server name>]`

Note that some shells require that you enclose a job array identifier in double quotes.

2.63.7 Exit Status

Zero

Upon successful processing of all options

Greater than zero

If `tracejob` is unable to process any options

2.63.8 See Also

The PBS Professional Administrator’s Guide
2.64 win_postinstall.py

For Windows. Configures PBS services

2.64.1 Synopsis

\texttt{\textbackslash etc\textbackslash python win\_postinstall.py -u <PBS service account> -p <PBS service account password>} [-t server]
\texttt{\textbackslash etc\textbackslash python win\_postinstall.py -u <PBS service account> -p <PBS service account password> -t <non-server installation type> -s <server name>}

2.64.2 Description

The win\_postinstall.py command configures PBS services. It performs post-installation steps such as validating the PBS service account username and password, installing the Visual C++ redistributable binary, creating \texttt{PBS\_HOME} and the default queue, initializing the database, and registering and starting the PBS\_SERVER, PBS\_SCHED, PBS\_MOM, PBS\_COMM and PBS\_RSHD PBS services.

When you use this command during installation of PBS, the command automatically un-registers any old PBS services. Available on Windows only.

2.64.2.1 Required Privilege

You must have Administrator privilege to run this command.

2.64.3 Options to win_postinstall.py

\texttt{-p, --passwd <PBS service account password>}

Specifies password for PBS service account.

\texttt{-s, --server <server name>}

Specifies the hostname on which the PBS server will run; required when the installation type is one of "execution", "client", or "comm".

\texttt{-t, --type <installation type>}

Specifies type of installation. Type can be one of "server", "execution", "client", or "comm". Default is server installation.

\texttt{-u, --user <PBS service account>}

Specifies PBS service account. When you specify the PBS service account, whether or not you are on a domain machine, include only the username, not the domain. For example, if the full username on a domain machine is \texttt{<domain>\<username>}, pass only \texttt{<username>} as an argument.
2.65  xpbs

Deprecated. GUI front end to PBS commands

2.65.1  Synopsis

xpbs [-admin]

xpbs --version

2.65.2  Description

The xpbs command provides a user-friendly point-and-click interface to PBS commands. See the sections below for a tour and tutorials. You can find a Help button within every dialog box.

2.65.3  Options

- admin

A mode where additional buttons are made available for terminating PBS servers, starting/ stopping/ disabling/ enabling queues, and running/ rerunning jobs.

--version

The xpbs command returns its PBS version information and exits. This option can only be used alone.

2.65.4  Getting Started

Running xpbs initializes the X resource database from various sources in the following order:

1. The RESOURCE_MANAGER property on the root window (updated via xrdb) with settings usually defined in the .Xdefaults file

2. Preference settings defined by the system administrator in the global xpbsrc file

3. User’s ~/.xbpsrc file - this file defines various X resources like fonts, colors, list of PBS hosts to query, criteria for listing queues and jobs, and various view states. See section 2.65.13, “Setting Preferences”, on page 239 below for a list of resources that can be set.

2.65.5  Running xpbs

To run xpbs as a regular, non-privileged user, type:

    setenv DISPLAY <display_host>:0

    xpbs

To run xpbs with the additional purpose of terminating PBS servers, stopping and starting queues, or running/ rerunning jobs:

    xpbs -admin

Be sure to appropriately set ~/.rhosts file if you’re planning to submit jobs to some remote server, and expecting output files to be returned to the local host (where xpbs was run). Usually, adding the PBS hostname running the server to your .rhosts file locally, and adding the name of the local machine to the .rhosts file at remote host, should be sufficient.
Also, be sure that the PBS client commands are in the default PATH because `xpbs` will call these commands.

### 2.65.6 The `xpbs` Display

This section describes the main parts of the `xpbs` display. The main window is composed of 5 distinct areas (subwindows) arranged vertically (one above another) in the following order:

1. Menu
2. Hosts
3. Queues
4. Jobs
5. Info

#### 2.65.6.1 Menu

The Menu area is composed of a row of command buttons that signal some action with a click of the left mouse button. The buttons are:

- **Manual Update**
  - Update the information on hosts, queues, and jobs.

- **Auto Update**
  - Update the information on hosts, queues, and jobs. Updating is done automatically every specified number of minutes.

- **Track Job**
  - Periodically check for returned output files of jobs.

- **Preferences**
  - Set certain parameters such as the list of server host(s) to query.

- **Help**
  - Show some help information.

- **About**
  - Display name of author and where to send comments, bugs, or suggestions.

- **Close**
  - Exit `xpbs` and save the current setup information (if anything has changed) in the user’s `$HOME/.xpbsrc` file. Information saved includes the selected host(s), queue(s), and job(s), the job-listing criteria, the view states (i.e. minimized/maximized) of the Hosts, Queues, Jobs, and INFO regions, and anything in the Preferences section.

#### 2.65.6.2 Hosts

The Hosts area is composed of a leading horizontal HOSTS bar, a listbox, and a set of command buttons. The HOSTS bar contains a minimize/maximize button, identified by a dot or a rectangular image, for displaying or iconifying the Hosts region. The listbox displays information about favorite server host(s), and each entry is meant to be selected via a single left mouse button click, a Shift key + mouse button 1 click for contiguous selection, or a Control key + mouse button 1 click for non-contiguous selection. The command buttons represent actions on selected host(s). Commonly found buttons:

- **detail**
  - Obtain detailed information about selected server host(s). This functionality can also be achieved by double-clicking on an entry in the Hosts listbox.
Submit
Submit a job to any of the queues managed by the selected host(s).

terminate
Terminate PBS servers on selected host(s). Available only with the -admin option.
The server hosts can be specified the resource in the ~/xpbsrc file (or .Xdefaults):

*serverHosts: hostname1 hostname2 ...

Another way of specifying the host is to click on the Preferences button in the Menu region, and manipulate the server Hosts entry widget from the preferences dialog box.

2.65.6.3 Queues

The Queues area is composed of a leading horizontal QUEUES bar, a listbox, and a set of command buttons. The QUEUES bar lists the hosts that are consulted when listing queues; the bar also contains a minimize/maximize button for displaying or iconifying the Queues region. The listbox displays information about queues managed by the server host(s) selected from the Hosts listbox; each listbox entry is meant to be selected (highlighted) via a single left mouse button click, a Shift key + mouse button 1 click for contiguous selection, or a Control key + mouse button 1 click for non-contiguous selection. The command buttons represent actions for operating on selected queue(s). Commonly found buttons:

detail
Obtain detailed information about selected queue(s). This functionality can also be achieved by double-clicking on a Queues listbox entry.

disable
Disable the selected queue(s). Available only with the -admin option.

enable
Enable the selected queue(s). Available only with the -admin option.

start
Start the selected queue(s). Available only with the -admin option.

stop
Stop the selected queue(s). Available only with the -admin option.

2.65.6.4 Jobs

The Jobs area is composed of a leading horizontal JOBS bar, a listbox, and a set of command buttons. The JOBS bar lists the queues that are consulted when listing jobs; the bar also contains a minimize/maximize button for displaying or iconifying the Jobs region. The listbox displays information about jobs that are found in the queue(s) selected from the Queues listbox; each listbox entry is meant to be selected (highlighted) via a single left mouse button click, a Shift key + mouse button 1 click for contiguous selection, or a Control key + mouse button 1 click for non-contiguous selection. The region just above the Jobs listbox shows a collection of command buttons whose labels describe criteria used for filtering the Jobs listbox contents. The list of jobs can be selected according to the job owner (JobOwner), job state (JobState), name of the job (JobName), type of hold placed on the job (HoldTypes), the account name associated with the job (AccountName), checkpoint attribute (Checkpoint), time the job is eligible for queueing/execution (QueueTime), resources requested by the job (ResourceList), priority attached to the job (Priority), and whether or not the job is rerunnable (Rerunable). The selection criteria can be modified by clicking on any of the appropriate command buttons to bring up a selection box. The criteria command buttons are accompanied by a Select Jobs button, which when clicked, updates the contents of the Jobs listbox based on the new selection criteria. Please see qselect(1B) for more details on how the jobs are filtered.

Finally, to the right of the listbox, the Jobs region is accompanied by the following command buttons, for operating on selected job(s):
detail
   Obtain detailed information about selected job(s). This functionality can also be achieved by double-clicking on a Jobs listbox entry.

modify
   Modify attributes of the selected job(s).

delete
   Delete the selected job(s).

hold
   Place hold on selected job(s).

release
   Release held job(s).

signal
   Send signals to selected running job(s).

msg
   Write a message string into the output streams of the selected job(s).

move
   Move selected job(s) into specified destination queue.

order
   Exchange order of two selected jobs in a queue.

run
   Run selected job(s). Available only with the -admin option.

rerun
   Requeue selected running job(s). Available only with the -admin option.

2.65.6.5 Info
The Info Area shows the progress of the commands executed by xpbs. Scan this box for errors. The INFO bar also contains a minimize/maximize button for displaying or iconifying the Info region.

2.65.7 Widgets Used in xpbs
Some of the widgets used in xpbs and how they are manipulated are described in the following:

2.65.7.1 listbox
Can be multi-selectable (a number of entries can be selected/highlighted using a mouse click) or single-selectable (one entry can be highlighted at a time). For a multi-selectable listbox, the following operations are allowed:

- Single click with mouse button 1 to select/highlight an entry
- Shift key + mouse button 1 to contiguously select more than one entry
- Control key + mouse button 1 to non-contiguously select more than one entry. NOTE: For systems running Tk < 4.0, the newly selected item is reshuffled to appear next to already selected items
- Click the Select All/Deselect All button to select all entries or deselect all entries at once
- Double-clicking an entry usually activates some action that uses the selected entry as a parameter
2.65.7.2 scrollbar

Usually appears either vertically or horizontally and contains 5 distinct areas that are mouse-clicked to achieve different effects:

- **top arrow**
  Causes the view in the associated widget to shift up by one unit (i.e. the object appears to move down one unit in its window). If the button is held down the action auto-repeats.

- **top gap**
  Causes the view in the associated window to shift up by one less than the number of units in the window (i.e. the portion of the object that used to appear at the very top of the window will now appear at the very bottom). If the button is held down the action auto-repeats.

- **slider**
  Pressing button 1 in this area has no immediate effect except to cause the slider to appear sunken rather than raised. However, if the mouse is moved with the button down then the slider is dragged, adjusting the view as the mouse is moved.

- **bottom gap**
  Causes the view in the associated window to shift down by one less than the number of units in the window (i.e. the portion of the object that used to appear at the very bottom of the window will now appear at the very top). If the button is held down the action auto-repeats.

- **bottom arrow**
  Causes the view in the associated window to shift down by one unit (i.e. the object appears to move up one unit in its window). If the button is held down the action auto-repeats.

2.65.7.3 entry

This is brought into focus with a click of the left mouse button. To manipulate this widget, type in the text value. You can use arrow keys, use the mouse to select text for deletion or overwrite, and copy and paste using only mouse buttons. This widget is usually accompanied by a scrollbar for horizontally scanning a long text entry string.

2.65.7.4 matrix of entry boxes

Usually shown as several rows of entry widgets where a number of entries (called fields) can be found per row. The matrix is accompanied by up/down arrow buttons for paging through the rows of data, and each group of fields gets one scrollbar for horizontally scanning long entry strings. Moving from field to field can be done using the <Tab>, <Cntrl-f>, or <Cntrl-b> (move backwards) keys.

2.65.7.5 spinbox

Combination of an entry widget and a horizontal scrollbar. The entry widget will only accept values that fall within a defined list of valid values, and incrementing through the valid values is done by clicking on the up/down arrows.

2.65.7.6 button

Rectangular region appearing either raised or pressed that invokes an action when clicked with the left mouse button. When the button appears pressed, hitting the <RETURN> key automatically selects the button.
2.65.7.7 text

Editor-like widget. This widget is brought into focus with a click of the left mouse button. To manipulate this widget, type in the text. You can use arrow keys and the backspace/delete key, use the mouse to select of text for deletion or overwrite, and copy and paste using only mouse buttons. This widget is usually accompanied by a scrollbar for vertically scanning a long entry.

2.65.8 Submitting Jobs

Submitting a PBS job requires only manipulation of the widgets found in the Submit window. The submit dialog box is composed of 4 distinct regions:

1. Job Script
2. Options
3. OTHER Options
4. Command Buttons

The Job Script file region is at the upper left, the Options region containing various widgets for setting job attributes is scattered all over the dialog box, the OTHER Options is located just below the Job Script file region, and the Command Buttons region is at the bottom.

The job script region is composed of a header box, the text box, FILE entry box, and a couple of buttons labeled load and save. If you have a script file containing PBS options and executable lines, type the name of the file on the FILE entry box, and then click on the load button. The various widgets in the Submit window are loaded with values found in the script file. The script file text box is loaded only with executable lines (non-#PBS) found in the script. The job script header box has a Prefix entry box that can be modified to specify the PBS directive to look for when parsing a script file for PBS options. If you don’t have a script file, you can start typing the executable lines of the job in the file text box.

To submit a job, perform the following steps:

1. Select a host from the HOSTS listbox in the main xpbs display
2. Click on the Submit button located in the Menu bar
3. Specify the script file containing the job execution lines and job resource and attribute values, or type the execution lines into the FILE textbox
4. Start manipulating the various widgets in the Submit window. Particularly, pay close attention to the Destination listbox. This box lists all the queues found in the host that you selected. A special entry called “@host” refers to the default queue at that host. Select the destination queue for the job. More options can be found by clicking the OTHER Options buttons.
5. At the bottom of the Submit window, click confirm submit. You can also click on interactive to run the job interactively. Running a job interactively opens an xterm window to your display host containing the session.

The script FILE entry box is accompanied by a save button that you click to save the current widget values to the specified file in a form that can later be read by xpbs or by the qsub command.
2.65.9 Modifying Attributes of Jobs

Modifying a PBS job requires only manipulation of the widgets found in the Modify window. To modify a job or jobs, do the following steps:
1. Select one or more jobs from the JOBS listbox in the main `xpbs` display
2. Click on the modify button located to the right of the listbox
3. The Modify window is structured similarly to the Submit window. Manipulate the widgets to specify replacement or additional values of job attributes
4. Click on the confirm modify button located at the bottom of the dialog box

2.65.10 Deleting Jobs

Deleting a PBS job requires only manipulation of the widgets found in the Delete window. To delete a job or jobs, do the following steps:
1. Select one or more jobs from the JOBS listbox in the main `xpbs` display
2. Click on the delete button located to the right of the listbox
3. Manipulate the spinbox widget to set the kill delay signal interval
4. Click on the delete button located at the bottom of the dialog box

2.65.11 Tracking Returned Output Files

If you want to be informed of returned output files of current jobs, and be able to quickly see the contents of those files, enable the “track job” feature, as follows:
1. Submit all the jobs that you want monitored
2. Click on the Track Job button located in the Menu bar to bring up the Track Job dialog box
3. Specify the list of usernames whose jobs are to be monitored for returned output files, in the matrix located at the upper left of the dialog box
4. Manipulate the minutes spinbox, located just below the user names matrix, to specify the interval value when output files are to be periodically checked
5. Specify the location of job output files (whether local or remote) by clicking on one of the radio buttons located at the upper right of the dialog box. Returned locally means the output files are returned to the host where `xpbs` was run. If the output files are returned to some remote host, `xpbs` will execute an RSH `<remote_host> test -f <output_files>` to test the existence of the files. RSH is whatever you set the remote shell command to in the corresponding entry box.
   Be sure the files are accessible from the host where `xpbs` was run (i.e. `.rhosts` appropriately set).
6. Click the start/reset tracking button located at the bottom of the dialog box. This does the following:
   • Cancel any previous tracking
   • Build a new list of jobs to be monitored for returned output files based on currently queued jobs
   • Start periodic tracking
7. Click on the close window button
When an output file for a job being monitored is found, the Track Job button (the one that originally invoked the Track Job dialog box) turns a different color, and the Jobs Found Completed listbox, located in the Track Job dialog box, is loaded with the corresponding job ID(s). Double-click on a job ID to see the contents of the output file and the error file. Click stop tracking if you want to cancel tracking.

2.65.12 Leaving xpbs

Click on the Close button located in the Menu bar to leave xpbs. If anything has changed, it brings up a dialog box asking for a confirmation with regard to saving state information such as the view states (minimize/maximize) of the HOSTS, QUEUES, JOBS, and INFO subwindows, and various criteria for listing queues and jobs. The information is saved in ~/.xpbsrc file.

2.65.13 Setting Preferences

You can set the following resources in the X resources file, ~/.xpbsrc:

*serverHosts
  List of server hosts (space-separated) to query by xpbs keyword. PBS_DEFAULT_SERVER can be used as a placeholder for the value obtained from *defServerFile.

*defServerFile
  File containing the name of the default server host. The contents of this is substituted for the PBS_DEFAULT_SERVER keyword in *serverHosts value.

*timeoutSecs
  Specifies the number of seconds before timing out while waiting for a connection to a PBS host.

*xtermCmd
  The xterm command to run when driving an interactive PBS session.

*labelFont
  Font applied to text appearing in labels.

*fixLabelFont
  Font applied to text that label fixed-width widgets such as listbox labels. This must be a fixed-width font.

*textFont
  Font applied to a text widget. Keep this as fixed-width font.

*backgroundColor
  Color applied to background of frames, buttons, entries, and scrollbar handles.

*foregroundColor
  Color applied to text in any context (under selection, insertion, etc.).

*activeColor
  Color applied to the background of a selection, a selected command button, or a selected scrollbar handle.

*disabledColor
  Color applied to a disabled widget.

*signalColor
  Color applied to buttons that signal something to the user about a change of state. For example, the color of the Track Job button when returned output files are detected.

*shadingColor
  Color shading applied to some of the frames to emphasize focus and/or decoration.
*selectorColor
   Color applied to the selector box of a radio button or checkbox.

*selectHosts
   List of hosts (space-separated) to automatically select/highlight in the HOSTS listbox.

*selectQueues
   List of queues (space-separated) to automatically select/highlight in the QUEUES listbox.

*selectJobs
   List of jobs (space-separated) to automatically select/highlight in the JOBS listbox.

*selectOwners
   List of owners checked when limiting the jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Owners: <list_of_owners>”. See -u option in qselect(1B) for format of <list_of_owners>.

*selectStates
   List of job states to look for (do not separate with spaces) when limiting the jobs appearing in the Jobs listbox in the main xpbs window. Specify value as “Job_States: <states_string>”. See -s option in qselect(1B) for format of <states_string>.

*selectRes
   List of resource amounts (space-separated) to consult when limiting the jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Resources: <res_string>”. See -l option in qselect(1B) for format of <res_string>.

*selectExecTime
   The Execution_Time attribute to consult when limiting the list of jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Queue_Time: <exec_time>”. See -a option in qselect(1B) for format of <exec_time>.

*selectAcctName
   Name of the account to check when limiting the jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Account_Name: <account_name>”. See -A option in qselect(1B) for format of <account_name>.

*selectCheckpoint
   The Checkpoint attribute relationship (including the logical operator) to consult when limiting the list of jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Checkpoint: <checkpoint_arg>”. See -c option in qselect(1B) for format of <checkpoint_arg>.

*selectHold
   The Hold_Types attribute string to look for in a job when limiting the jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Hold_Types: <hold_string>”. See -h option in qselect(1B) for format of <hold_string>.

*selectPriority
   The priority relationship (including the logical operator) to consult when limiting the list of jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Priority: <priority_value>”. See -p option in qselect(1B) for format of <priority_value>.

*selectRerun
   The Rerunable attribute value to consult when limiting the list of jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Rerunnable: <rerun_val>”. See -r option in qselect(1B) for format of <rerun_val>.

*selectJobName
   Name of the job to check when limiting the jobs appearing on the Jobs listbox in the main xpbs window. Specify value as “Job_Name: <jobname>”. See -N option in qselect(1B) for format of <jobname>.
2.65.14  \texttt{xpbs} and PBS Commands

\texttt{xpbs} calls PBS commands as follows:

\begin{center}
\begin{tabular}{|l|l|}
\hline
\textbf{Command Button} & \textbf{PBS Command} \\
\hline
detail (Hosts) & qstat -B -f <selected server host(s)> \\
terminate & qterm <selected server host(s)> \\
detail (Queues) & qstat -Q -f <selected queue(s)> \\
stop & qstop <selected queue(s)> \\
start & qstart <selected queue(s)> \\
enable & qenable <selected queue(s)> \\
disable & qdisable <selected queue(s)> \\
detail (Jobs) & qstat -f <selected job(s)> \\
modify & qalter <selected job(s)> \\
delete & qdel <selected job(s)> \\
hold & qhold <selected job(s)> \\
release & qrls <selected job(s)> \\
run & qrun <selected job(s)> \\
rerun & qrerun <selected job(s)> \\
signal & qsig <selected job(s)> \\
msg & qmsg <selected job(s)> \\
moving & qmove <selected job(s)> \\
order & qorder <selected job(s)> \\
\hline
\end{tabular}
\end{center}
2.65.15 Exit Status

Zero
   Upon successful processing
Greater than zero
   If the xpbs command fails

2.65.16 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide
2.66  xpbsmon

Deprecation. PBS GUI for displaying and monitoring execution hosts

2.66.1  Synopsis

xpbsmon
xpbsmon --version

2.66.2  Description

The xpbsmon command provides a way to graphically display information about the vnodes that run jobs. A vnode or execution host can be either running or not running a pbs_mom daemon. In the latter case, it could be just a vnode name that appears in a nodes file that is managed by a pbs_server running on a different host. The xpbsmon utility also lets you monitor values of certain system resources by querying the pbs_mom. You can see what job is running on what vnode, who owns the job, how many vnodes assigned to a job, status of each vnode (vnodes are color-coded, and the colors are user-modifiable), how many vnodes are available, free, down, reserved, offline, of unknown status, in use running multiple jobs, or running only 1 job. See the sections below for a tour and tutorials of xpbsmon. You can find a Help button within every dialog box.

2.66.3  Getting Started

Running xpbsmon initializes the X resource database from various sources in the following order:

1. The RESOURCE_MANAGER property on the root window (updated via xrdb) with settings usually defined in the .Xdefaults file
2. Preference settings defined by the system administrator in the global xpbsmonrc file
3. User’s ~/.xpemonsrc file - this file defines various X resources such as fonts, colors, a list of colors to use to represent the status of the vnodes, a list of PBS sites to query, a list of server hosts on each site, a list of vnodes/execution hosts on each server host, a list of system resource queries to send to the nodes’ pbs_mom, and view states. See section 2.66.10, “Setting Preferences”, on page 248 below for a list of resources that can be set.

2.66.4  Running xpbsmon

You can run xpbsmon as either a regular user or superuser. If you run it with less privilege, you may not be able to see all the information for a vnode. If you run it as a regular user, you should still be able to see what jobs are running on what vnodes, possibly state, as this information is obtained by xpbsmon talking directly to the specified server. If you want other system resource values, it may require special privilege since xpbsmon will have to talk directly to the pbs_mom of a vnode. In addition, the host where xpbsmon is running must also have been given explicit access permission by the MoM (unless the GUI is running on the same host where MoM is running). This is done done by updating the $clienthost and/or the $restricted parameter in the MoM’s configuration file.

To run xpbsmon, type:

```
setenv DISPLAY <display_host>:0
xpbsmon
```

If you are running the GUI and are interested only in job data, be sure to set all the vnodes’ type to NOMOM in the Pref dialog box.
2.66.5 Options

--version
The xpbsmon command returns its PBS version information and exits. This option can only be used alone.

2.66.6 The xpbsmon Display

This section describes the main parts of the xpbsmon display. The main window is composed of 3 distinct areas (sub-windows) arranged vertically (one above another) in the following order:

1. Menu
2. Site Information
3. Info

2.66.6.1 Menu

The Menu area is composed of a row of command buttons that trigger some action with a click of the left mouse button. The buttons are:

Site..
Displays a popup menu containing the list of PBS sites that have been added using the Sites Preferences window. Drag your mouse and release it on the site name whose servers/vnodes information you would like to see.

Pref..
Brings up dialog boxes for specifying the list of sites, servers on each site, vnodes that are known to a server, and system resource queries to be sent to a vnode’s pbs_mom daemon.

Auto Update..
Brings up a window for specifying whether or not to do auto updates of vnodes information, along with the interval in number of minutes between updates.

Help
Contains help information.

About
 Tells who the author is and where to send comments, bugs, or suggestions.

Close
Exits xpbsmon and saves the current setup information, if anything has changed, in the user’s $HOME/.xpbsmonrc file. Information saved includes the specified list of sites, servers on each site, vnodes known to each server, and system resource queries to send to vnode’s pbs_mom.

Minimize Button
Shows the iconified view of Site Information where vnodes are represented as tiny boxes, where each box is colored according to status. In order to get more information about a vnode, you need to double-click on the colored box.

Maximize button
Shows the full view of Site Information where vnodes are represented in bigger boxes, still colored according to status, with other information displayed.
2.66.6.2 Site Information

Only one site at a time can be displayed. This area (shown as one huge box referred to as the site box) can be further subdivided into three areas: the site name label at the top, server boxes in the middle, and the color status bar at the bottom. The site name label shows the name of the site as specified in the Pref. window. The middle of the site box shows a row of big boxes housing smaller boxes.

The big box is an abstraction of a server host (called a server box), showing its server display label at the top of the box, a grid of smaller boxes representing the vnodes that the server knows about (where jobs are run), and summary status for the vnodes under the server. Status information shows counters for the number of vnodes used, available, reserved, offline, or of unknown status, and the number of CPUs assigned. For a cleaner display, some counters with a value of zero are not displayed. The server boxes are placed in a grid, with a new row for when *siteBoxMaxNumServerBoxesPerRow or *siteBoxMaxWidth limit has been reached.

The smaller boxes represent the vnodes/execution hosts where jobs are run (referred to as node boxes). Each node box shows the name at the top, and a sub-box (a smaller square) colored according to the status of the vnode that it represents, and if the view type is FULL, it displays vnode information according to the system resource queries specified on the Pref. window. Clicking on the sub-box shows a much bigger box (called the MIRROR view) with bigger fonts containing vnode information. The ICON view shows a tiny box with a colored area. The node boxes are arranged in a grid, with a new row for when the *serverBoxMaxNumNodeBoxesPerRow or *serverBoxMaxWidth limit has been reached. ICON view of the node boxes is constrained by the *nodeBoxIconMaxHeight and *nodeBoxIconMaxWidth pixel values; FULL view of the node boxes is bounded by *nodeBoxFullMaxWidth and *nodeBoxFullMaxHeight; the size of the mirror view of the node boxes is *nodeBoxMirrorMaxWidth and *nodeBoxMirrorMaxHeight.

Horizontal and vertical scrollbars for the site box, server box, and node box are displayed as needed.

Finally, the color bar information shows a color chart explaining the how colors indicate vnode status. The color-to-status mapping can be modified by setting the X resources: *nodeColorNOINFO, *nodeColorFREE, *nodeColorINUSE-shared, *nodeColorINUSE-exclusive, *nodeColorDOWN, *nodeColorRSVD, *nodeColorOFFL, and *nodeColorBUSY.

2.66.6.3 Info

The Info Area shows the progress of some of the background actions performed by xpbsmon. Scan this box for errors.

2.66.7 Widgets Used in xpbsmon

We describe some of the widgets used in xpbsmon and how they are manipulated here:

2.66.7.1 listbox

Listboxes found in this GUI are single-selectable only (one entry can be highlighted/selected at a time via a mouse click).

2.66.7.2 scrollbar

Usually appears either vertically or horizontally and contains five distinct mouse-clickable areas:

- **top arrow**
  
  Causes the view in the associated widget to shift up by one unit (i.e. the object appears to move down one unit in its window). If the button is held down the action auto-repeats.

- **top gap**
  
  Causes the view in the associated window to shift up by one less than the number of units in the window (i.e. the portion of the object that used to appear at the very top of the window will now appear at the very bottom). If the button is held down the action auto-repeats.
### slider

Pressing button 1 in this area has no immediate effect except to cause the slider to appear sunken rather than raised. However, if the mouse is moved with the button down then the slider is dragged, adjusting the view as the mouse is moved.

### bottom gap

Causes the view in the associated window to shift down by one less than the number of units in the window (i.e. the portion of the object that used to appear at the very bottom of the window will now appear at the very top). If the button is held down the action auto-repeats.

### bottom arrow

Causes the view in the associated window to shift down by one unit (i.e. the object appears to move up one unit in its window). If the button is held down the action auto-repeats.

#### 2.66.7.3 entry

Bring this into focus with a click of the left mouse button. To manipulate this widget, type in the text value. You can use arrow keys, use the mouse to select text for deletion or overwrite, and copy and paste using only mouse buttons. This widget is usually accompanied by a scrollbar for horizontally scanning a long text entry string.

#### 2.66.7.4 box

This is made up of one or more listboxes displayed adjacent to each other giving the effect of a “matrix”. Each row from the listboxes makes up an element of the box. In order to add items to the box, manipulate the accompanying entry widgets, one for each listbox, and then click the add button. Removing items from the box is done by selecting an element, and then clicking delete.

#### 2.66.7.5 spinbox

Combination of an entry widget and a horizontal scrollbar. The entry widget will accept only those values that fall within a defined list of valid values, and stepping through the valid values is done by clicking on the up/down arrows.

#### 2.66.7.6 button

Rectangular region appearing either raised or pressed that invokes an action when clicked with the left mouse button. When the button appears pressed, hitting the <RETURN> key will automatically select the button.
2.66.8 Updating Preferences

2.66.8.1 Time Sharing

Suppose you have a time-sharing environment where the front end is called *bower* and you have four vnodes: bower1, bower2, bower3, and bower4. *bower* is the host that runs the server; jobs are submitted to host bower where it enqueues it for future execution. A *pbs_mom* daemon is running on each of the execution hosts. If the server bower also maintains a nodes list containing information such as state for the four vnodes, this is reported. To set up *xpbsmon*, do the following:

1. Click the *Pref.* button on the Menu section.
2. On the *Sites Preference* dialog, enter any arbitrary site name, for example “Local”. Then click the *add* button.
3. On the *Server_Host* entry box, enter “bower”, and on the *DisplayLabel* entry box, put an arbitrary label (as it would appear on the header of the server box) such as “Bower”, and then click *add*.
4. Click the *nodes.* button that is accompanying the *Servers* box. This brings up the *Server Preference* dialog.
5. Now add the entries “bower1”, “bower2”, “bower3”, “bower4” specifying type *MoM* for each on the *Nodes* box.
6. If you need to monitor certain system resource parameters for each of the vnodes, specify query expressions containing resource queries to be sent to the individual PBS MoMs. For example, if you want to obtain memory usage, select a vnode from the Nodes list, click on the *query.* button that accompanies the Nodes list, and this brings up the *Query Table* dialog. Specify the following input:

   *Query.Expr*: (availmem/totmem) * 100
   *Display_Info*: Memory Usage:
   *Display_Type*: SCALE

   The above says to display the result of the “Query.Expr” in a scale widget calibrated over 100. The queries “availmem” and “totmem” are sent to the PBS MoM, and the expression is evaluated upon receiving all results from the MoM. If you want to display the result of another query, say “loadave”, directly, specify the following:

   *Query.Expr*: loadave
   *Display_Info*: Load Average:
   *Display_Type*: TEXT

   For a list of queries that can be sent to a *pbs_mom*, click on the *Help* button on the Query table window.

2.66.8.2 Jobs-Exclusive Environment

Suppose you have a “space non-sharing” environment where the server maintains a list of vnodes that it runs jobs on exclusively (one job at a time outstanding per vnode). Let’s call this server b1. Update Preferences information as follows:

1. Click the *Pref.* button on the Menu section.
2. In the *Sites Preference* dialog, enter a site name, for example “B System”. Click the *add* button.
3. In the *Server_Host* entry box, enter “b1”, in the *DisplayLabel* entry box type “B1” (or whatever label that you would like to appear on the header of the server box), and then click *add*.

2.66.8.3 Hybrid Time Sharing/Space Sharing Environment

A cluster of heterogeneous machines, time-sharing or jobs-exclusive, could easily be represented in *xpbsmon* by combining steps in section 2.66.8.1, “Time Sharing”, on page 247 and section 2.66.8.2, “Jobs-Exclusive Environment”, on page 247.
2.66.9 Leaving \texttt{xpbsmon}

Click on the \textit{Close} button located in the Menu bar to leave \texttt{xpbsmon}. If anything has changed, it brings up a dialog box asking for a confirmation about saving preferences information such as a list of sites, their view types, the list of servers on each site, the list of vnodes known to each server, and the list of queries to be sent to the \texttt{pbs\_mom} of each vnode. The information is saved in \texttt{~/.xpbsmonrc} file.

2.66.10 Setting Preferences

The resources that can be set in the X resources file, \texttt{~/.xpbsmonrc}, are described in the following:

2.66.10.1 Node Box Properties

Resource names beginning with “*small” or “*node” apply to the properties of the node boxes. A node box is made of an outer frame where the node label sits on top, the canvas (smaller box) is on the middle, and possibly some horizontal/vertical scrollbars.

- \texttt{nodeColorNOINFO}
  
  Color of node box when information for the vnode it represents could not be obtained.

- \texttt{*nodeColorFREE}
  
  Color of canvas when vnode it represents is up.

- \texttt{*nodeColorINUSEshared}
  
  Color when vnode it represents has more than one job running on it, or when vnode has been marked by the server that manages it as “job-sharing”.

- \texttt{*nodeColorINUSEexclusive}
  
  List of colors to assign to a node box when host it represents is running only one job, or when vnode has been marked by the server that manages it as “time-sharing”. \texttt{xpbsmon} uses this list to assign one distinct color per job unless all the colors have been exhausted, in which case colors start being assigned more than once, in a round-robin fashion.

- \texttt{*nodeColorDOWN}
  
  Color when vnode it represents is down.

- \texttt{*nodeColorRSVD}
  
  Color when vnode it represents is reserved.

- \texttt{*nodeColorOFFFL}
  
  Color when vnode it represents is offline.

- \texttt{*nodeColorBUSY}
  
  Color when vnode it represents is busy (high load average).

- \texttt{*smallForeground}
  
  Color of text inside the canvas.

- \texttt{*smallBackground}
  
  Color of the frame.

- \texttt{*smallBorderWidth}
  
  Distance (in pixels) from other node boxes.

- \texttt{*smallRelief}
  
  How node box will visually appear (style).
PBS Commands

Chapter 2

*smallScrollBorderWidth
  Significant only in FULL mode; the distance of the horizontal/vertical scrollbars from the canvas and lower edge of the frame.

*smallScrollBackground
  Background color of the scrollbars

*smallScrollRelief
  How scrollbars visually appear (style).

*smallCanvasBackground
  Color of the canvas (later overridden depending on status of the vnode it represents)

*smallCanvasBorderWidth
  Distance of the canvas from the frame and possibly the scrollbars.

*smallCanvasRelief
  How the canvas is visually represented (style).

*smallLabelBorderWidth
  Distance of the node label from the canvas and the topmost edge of the frame.

*smallLabelBackground
  Background of the area of the node label that is not filled.

*smallLabelRelief
  How the label appear visually (style).

*smallLabelForeground
  Color of node label text.

*smallLabelFont
  Font for the node label text.

*smallLabelFontWidth
  Font width (in pixels) of *smallLabelFont

*smallLabelFontHeight
  Font height (in pixels) of *smallLabelFont

*smallTextFont
  Font for the text that appear inside a canvas.

*smallTextFontWidth
  Font width (in pixels) of *smallTextFont.

*smallTextFontHeight
  Font height (in pixels) of *smallTextFont.

*nodeColorTrough
  Color of trough part (the /100 portion) of a canvas scale item.

*nodeColorSlider
  Color of slider part (value portion) of a canvas scale item.

*nodeColorExtendedTrough
  Color of extended trough (over 100 portion when value exceeds max) of a canvas scale item.

*nodeScaleFactor
  How much bigger you want the scale item on the canvas to appear. (1 means to keep size as is)
*nodeBoxFullMaxWidth
*nodeBoxFullMaxHeight
    Maximum width and height (in pixels) of a node box in FULL mode.

*nodeBoxIconMaxWidth
*nodeBoxIconMaxHeight
    Maximum width and height (in pixels) of a node box in ICON mode.

*nodeBoxMirrorMaxWidth
*nodeBoxMirrorMaxHeight
    Maximum width and height (in pixels) of a node box displayed on a separate window (after it has been clicked with the mouse to obtain a bigger view)

*nodeBoxMirrorScaleFactor
    How much bigger you want the scale item on the canvas to appear while the node box is displayed on a separate window (1 means to keep size as is)

2.66.10.2 Server Box Properties

Resource names beginning with “*medium” apply to the properties of the server boxes. A server box is made of an outer frame where the server display label sits on top, a canvas filled with node boxes is on the middle, possibly some horizontal/vertical scrollbars, and a status label at the bottom.

*mediumLabelForeground
    Color of text applied to the server display label and status label.

*mediumLabelBackground
    Background color of the unfilled portions of the server display label and status label.

*mediumLabelBorderWidth
    Distance of the server display label and status label from other parts of the server box.

*mediumLabelRelief
    How the server display label and status label appear visually (style).

*mediumLabelFont
    Font for the text of the server display label and status label.

*mediumLabelFontWidth
    Font width (in pixels) of *mediumLabelFont.

*mediumLabelFontHeight
    Font height (in pixels) of *mediumLabelFont.

*mediumCanvasBorderWidth
    Distance of the server box’s canvas from the label widgets.

*mediumCanvasBackground
    Background color of the canvas.

*mediumCanvasRelief
    How the canvas appear visually (style).

*mediumScrollBorderWidth
    Distance of the scrollbars from the other parts of the server box.

*mediumScrollBackground
    Background color of the scrollbars
PBS Commands

**2.66.10.3 Miscellaneous Properties**

Resource names beginning with "**big**" apply to the properties of a site box, as well as to widgets found outside of the server box and node box. This includes the dialog boxes that appear when the menu buttons of the main window are manipulated. The site box is the one that appears on the main region of `xpbsmon`.

- **bigBackground**: Background color of the outer layer of the main window.
- **bigForeground**: Color applied to regular text that appear outside of the node box and server box.
- **bigBorderWidth**: Distance of the site box from the menu area and the color information area.
- **bigRelief**: How the site box is visually represented (style).
- **bigActiveColor**: Color applied to the background of a selection, a selected command button, or a selected scroll bar handle.
- **bigShadingColor**: Color shading applied to some of the frames to emphasize focus as well as decoration.
- **bigSelectorColor**: Color applied to the selector box of a radiobutton or checkbutton.
- **bigDisabledColor**: Color applied to a disabled widget.
- **bigLabelBackground**: Color applied to the unfilled portions of label widgets.
- **bigLabelBorderWidth**: Distance from other widgets of a label widget.
- **bigLabelRelief**: How label widgets appear visually (style).
- **bigLabelFont**: Font for labels.
*bigLabelFontWidth
  Font width (in pixels) of *bigLabelFont.

*bigLabelFontHeight
  Font height (in pixels) of *bigLabelFont.

*bigLabelForeground
  Color applied to text that function as labels.

*bigCanvasBackgroundColor
  Color of the main region.

*bigCanvasRelief
  How the main region looks like visually (style)

*bigCanvasBorderWidth:
  Distance of the main region from the menu and info regions.

*bigScrollBorderWidth
  If the main region has a scrollbar, this is its distance from other widgets appearing on the region.

*bigScrollBackgroundColor
  Background color of the scrollbar appearing outside a server box and node box.

*bigScrollRelief
  How the scrollbar that appears outside a server box and node box looks like visually (style)

*bigTextFontWidth
  Font width (in pixels) of *bigTextFont

*bigTextFontHeight
  Font height (in pixels) of *bigTextFont

*siteBoxMaxWidth
  Maximum width (in pixels) of the site box.

*siteBoxMaxHeight
  Maximum height (in pixels) of the site box.

*siteBoxMaxNumServerBoxesPerRow
  Maximum number of server boxes to appear in a row inside the site box.

*autoUpdate
  If set to True, information about vnodes is gathered periodically.

*autoUpdateMins
  Number of minutes between polling for data regarding vnodes when *autoUpdate is set.

*siteInView
  Name of the site that should be in view.

*rcSiteInfoDelimeterChar
  Separator character for each input within a curly-bracketed line of input of *siteInfo.

*siteInfo
  
  {<site1name><sep><site1-display-type><sep><server-name><sep><server-display-label><sep><nodename><sep><nodetype><sep><node-query-expr>}
  ...
  {<site2name><sep><site2-display-type> <sep><server-name><sep>
<server-display-label><sep><nodename><sep><nodetype><sep>

Information about a site where <site-display-type> can be either {FULL,ICON}, <nodetype> can be {MOM, NOMOM}, and <node-query-expr> has the format:

{/ <expr>} {expr-label} <output-format>

where <output-format> could be {TEXT, SCALE}. It’s probably better to use the Pref dialog boxes in order to specify a value for this.

Example:

*rcSiteInfoDelimiterChar ;
*sitesInfo: {NAS;ICON;newton;Newton; newton3; NOMOM;} {Langley; FULL; db;DB; db.nas.nasa.gov; MOM; {{ ( availmem / totmem ) * 100} {Memory Usage:} SCALE} {{ ( loadave / ncpus ) * 100} {Cpu Usage:} SCALE} {ncpus {Number of Cpus:} TEXT} {physmem {Physical Memory:} TEXT} {idletime {Idle Time (s):} TEXT} {loadave {Load Avg:} TEXT} {NAS;ICON;newton;Newton;newton4; NOMOM;} {NAS;ICON;newton;Newton; newton1;NOMOM;} {NAS; ICON;newton;Newton; newton2;NOMOM;} {NAS;ICON;b0101;DB;aspasia.nas.nasa.gov; MOM;{{ ( availmem / totmem ) * 100} {Memory Usage:} SCALE} {{ ( loadave / ncpus ) * 100} {Cpu Usage:} SCALE} {ncpus {Number of Cpus:} TEXT} {physmem {Physical Memory:} TEXT} {idletime {Idle Time (s):} TEXT} {loadave {Load Avg:} TEXT} {NAS;ICON;newton;Newton;newton7;NOMOM;}

2.66.11 Exit Status

Zero

Upon successful processing

Greater than zero

If the xpbsmon command fails

2.66.11.1 Harmless Error Message

If xpbsmon is querying a host running a server with an incompatible version, you may see the following messages:

Internal error: pbsstatnode: End of File (15031)

The above message can be safely ignored.

2.66.12 See Also

The PBS Professional User's Guide, the PBS Professional Administrator's Guide
This chapter describes the configuration files used by MoM and lists the MoM configuration parameters that are found in the Version 1 MoM configuration file, PBS_HOME/mom_priv/config.

### 3.1 Syntax of MoM Configuration File

The Version 1 MoM configuration file contains parameter settings for the MoM on the local host.

Version 1 configuration files list local resources and initialization values for MoM. Local resources are either static, listed by name and value, or externally-provided, listed by name and command path. Local static resources are for use only by the scheduler for MoM’s partition. They do not appear in a `pbsnodes -a` query. See the `-c` option to the `pbs_mom` command. Do not change the syntax of the Version 1 configuration file.

Each configuration item is listed on a single line, with its parts separated by white space. Comments begin with a hash-mark (`#`).

#### 3.1.1 Externally-provided Resources

Externally-provided resources, for example dynamic resources such as scratch space, use a shell escape to run a command. These resources are described with a name and value, where the first character of the value is an exclamation mark (`!`). The remainder of the value is the path and command to execute.

Parameters in the command beginning with a percent sign (`%`) can be replaced when the command is executed. For example, this line in a configuration file describes a resource named "escape":

```plaintext
escape   !echo %xxx %yyy
```

If a query for the "escape" resource is sent with no parameter replacements, the command executed is "echo %xxx %yyy". If one parameter replacement is sent, "escape[xxx=hi there]", the command executed is "echo hi there %yyy". If two parameter replacements are sent, "escape[xxx=hi][yyy=there]", the command executed is "echo hi there". If a parameter replacement is sent with no matching token in the command line, "escape[zzz=snafu]", an error is reported.

#### 3.1.2 Windows Notes

If the argument to a MoM option is a pathname containing a space, enclose it in double quotes as in the following:

```plaintext
hostn !"\Program Files\PBS\exec\bin\hostn* host
```

When you edit any PBS configuration file, make sure that you put a newline at the end of the file. The Notepad application does not automatically add a newline at the end of a file; you must explicitly add the newline.
3.2 Contents of MoM Configuration File

3.2.1 Replacing Actions

\$\textit{action} <\textit{default action}> <\textit{timeout}> <\textit{new action}>

Replaces the \textit{default action} for an event with the site-specified \textit{new action}. \textit{timeout} is the time allowed for \textit{new action} to run. \textit{new action} is the site-supplied script that replaces \textit{default action}. This is the complete list of values for \textit{default action}:

<table>
<thead>
<tr>
<th>\textit{default action}</th>
<th>\text{Result}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{checkpoint}</td>
<td>Run \textit{new action} in place of the periodic job checkpoint, after which the job continues to run.</td>
</tr>
<tr>
<td>\textit{checkpoint_abort}</td>
<td>Run \textit{new action} to checkpoint the job, after which the job must be terminated by the script.</td>
</tr>
<tr>
<td>\textit{multinodebusy &lt;timeout&gt; requeue}</td>
<td>Used with cycle harvesting and multi-vnode jobs. Changes default behavior when a vnode becomes busy. Instead of allowing the job to run, the job is requeued. Timeout is ignored. The only \textit{new action} is \textit{requeue}.</td>
</tr>
<tr>
<td>\textit{restart}</td>
<td>Runs \textit{new action} in place of restart.</td>
</tr>
<tr>
<td>\textit{terminate}</td>
<td>Runs \textit{new action} in place of SIGTERM or SIGKILL when MoM terminates a job.</td>
</tr>
</tbody>
</table>

3.2.2 Initialization Values

Initialization value directives have names beginning with a dollar sign ("\$"). We list them here:

\$\textit{alps\_client <path>}

Cray only. Path to the Cray apbasil command. Must be full path to command.

Format: \textit{path to command}

Default: None

\$\textit{alps\_release\_timeout <timeout>}

Cray only. Specifies the amount of time that PBS tries to release an ALPS reservation before giving up. After this amount of time has passed, PBS stops trying to release the ALPS reservation, the job exits, and the job’s resources are released. PBS sends a HUP to the MoM so that she rereads the ALPS inventory to get the current available ALPS resources.

We recommend that the value for this parameter be twice the value for \textit{suspectbegin}.

Format: \textit{Seconds, specified as positive integer}

Default: 600 (10 minutes)
**$checkpoint_path <path>**

MoM passes this parameter to the checkpoint and restart scripts. This path can be absolute or relative to PBS_HOME/mom_priv. Overrides default. Overridden by path specified in the pbs_mom -C option and by PBS_CHECKPOINT_PATH environment variable. See "Specifying Checkpoint Path" on page 421 in the PBS Professional Administrator’s Guide.

**$clienthost <hostname>**

hostname is added to the list of hosts which are allowed to connect to MoM as long as they are using a privileged port. For example, this allows the hosts “fred” and “wilma” to connect to MoM:

```
$clienthost fred
$clienthost wilma
```

The following hostnames are added to $clienthost automatically: the server, the localhost, and if configured, the secondary server. The server sends each MoM a list of the hosts in the nodes file, and these are added internally to $clienthost. None of these hostnames need to be listed in the configuration file.

Two hostnames are always allowed to connect to pbs_mom, “localhost” and the name returned to MoM by the system call `gethostname()` . These hostnames do not need to be added to the MoM configuration file.

The hosts listed as “clienthosts” make up a “sisterhood” of machines. Any one of the sisterhood will accept connections from within the sisterhood. The sisterhood must all use the same port number.

**cpuset_create_flags <flags>**

Lists the flags for when MoM does a `cpusetCreate(3)` for each job. flags is an or-ed list of flags. The flags are:

- HPE SGI machine with supported versions of HPE MPI:
  ```
  CPUSET_CPU_EXCLUSIVE | 0
  ```
  Default: 0

**cpuset_destroy_delay <delay>**

MoM waits up to delay seconds before destroying a cpuset of a just-completed job, but not longer than necessary. This gives the operating system more time to clean up leftover processes after they have been killed.

Example:

```
cpuset_destroy_delay 10
```

Format: Integer

Default for HPE SGI systems: 0

**$cpuset_error_action**

When using a cpuset-enabled MoM, specifies the action taken when a cpuset creation error occurs. Can take one of the following values:

- continue
  - The error is logged and the job is killed and requeued.

- offline
  - The vnodes on this host for this job are marked offline, and the job is requeued.

Format: String

Allowable values: continue, offline

Default: offline
$cputmult <factor>
This sets a factor used to adjust CPU time used by each job. This allows adjustment of time charged and limits enforced where jobs run on a system with different CPU performance. If MoM’s system is faster than the reference system, set factor to a decimal value greater than 1.0. For example:
$cputmult 1.5
If MoM’s system is slower, set factor to a value between 1.0 and 0.0. For example:
$cputmult 0.75
$\text{dce\_refresh\_delta} <\text{delta}>
Defines the number of seconds between successive refreshings of a job’s DCE login context. For example:
$\text{dce\_refresh\_delta} 18000
$enforce <\text{limit}>
MoM will enforce the given limit. Some limits have associated values. Syntax:
$enforce <\text{variable name}> <\text{value}>
$enforce mem
MoM will enforce each job’s memory limit.
$enforce cpuaverage
MoM will enforce ncpus when the average CPU usage over a job’s lifetime usage is greater than the job’s limit.
$enforce average\_trialperiod <\text{seconds}>
Modifies cpuaverage. Minimum number of seconds of job walltime before enforcement begins.
Format: Integer
Default: 120
$enforce average\_percent\_over <\text{percentage}>
Modifies cpuaverage. Gives percentage by which a job may exceed its ncpus limit.
Format: Integer
Default: 50
$enforce average\_cpufactor <factor>
Modifies cpuaverage. The ncpus limit is multiplied by factor to produce actual limit.
Format: Float
Default: 1.025
$enforce cpuburst
MoM will enforce the ncpus limit when CPU burst usage exceeds the job’s limit.
$enforce delta\_percent\_over <\text{percentage}>
Modifies cpuburst. Gives percentage over limit to be allowed.
Format: Integer
Default: 50
$enforce delta\_cpufactor <factor>
Modifies cpuburst. The ncpus limit is multiplied by factor to produce actual limit.
Format: Float
Default: 1.5
$enforce delta\_weightup <factor>
Modifies cpuburst. Weighting factor for smoothing burst usage when average is increasing.
Format: Float
Default: 0.4
$\text{enforce delta_weightdown <factor>}$

Modifies $\text{cpuburst}$. Weighting factor for smoothing burst usage when average is decreasing.

Format: Float
Default: 0.4

$\text{ideal_load <load>}$

Defines the load below which the vnode is not considered to be busy. Used with the $\text{max_load}$ directive.

Example:

$\text{ideal_load 1.8}$
Format: Float
No default

$\text{jobdir_root <stage directory root>}$

Directory under which PBS creates job-specific staging and execution directories. PBS creates a job’s staging and execution directory when the job’s sandbox attribute is set to $\text{PRIVATE}$. If $\text{jobdir_root}$ is unset, it defaults to the job owner’s home directory. In this case the user’s home directory must exist. If $\text{stage directory root}$ does not exist when MoM starts up, MoM will abort. If $\text{stage directory root}$ does not exist when MoM tries to run a job, MoM will kill the job. Path must be owned by root, and permissions must be 1777. On Windows, this directory should have Full Control Permission for the local Administrators group.

Example:

$\text{jobdir_root /scratch/foo}$

$\text{kbd_idle <idle wait> <min use> <poll interval>}$

Declares that the vnode will be used for batch jobs during periods when the keyboard and mouse are not in use.

$\text{idle wait}$

Time, in seconds, that the workstation keyboard and mouse must be idle before being considered available for batch jobs.

Must be set to non-zero value for cycle harvesting to be enabled.

Format: Integer
No default

$\text{min use}$

Time, in seconds, during which the workstation keyboard or mouse must continue to be in use before the workstation is determined to be unavailable for batch jobs.

Format: Integer
Default: 10

$\text{poll interval}$

Interval, in seconds, at which MoM checks for keyboard and mouse activity.

Format: Integer
Default: 1

Example:

$\text{kbd_idle 1800 10 5}$
$\text{logevent} <\text{mask}>$

Sets the mask that determines which event types are logged by \( \text{pbs\_mom} \). To include all debug events, use \( 0xffffffff \).

Log events:

<table>
<thead>
<tr>
<th>Name</th>
<th>Hex Value</th>
<th>Message Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>0001</td>
<td>Internal errors</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>0002</td>
<td>System errors</td>
</tr>
<tr>
<td>ADMIN</td>
<td>0004</td>
<td>Administrative events</td>
</tr>
<tr>
<td>JOB</td>
<td>0008</td>
<td>Job-related events</td>
</tr>
<tr>
<td>JOB_USAGE</td>
<td>0010</td>
<td>Job accounting info</td>
</tr>
<tr>
<td>SECURITY</td>
<td>0020</td>
<td>Security violations</td>
</tr>
<tr>
<td>SCHED</td>
<td>0040</td>
<td>Scheduler events</td>
</tr>
<tr>
<td>DEBUG</td>
<td>0080</td>
<td>Common debug messages</td>
</tr>
<tr>
<td>DEBUG2</td>
<td>0100</td>
<td>Uncommon debug messages</td>
</tr>
<tr>
<td>RESV</td>
<td>0200</td>
<td>Reservation-related info</td>
</tr>
<tr>
<td>DEBUG3</td>
<td>0400</td>
<td>Rare debug messages</td>
</tr>
<tr>
<td>DEBUG4</td>
<td>0800</td>
<td>Limit-related messages</td>
</tr>
</tbody>
</table>

$\text{max\_check\_poll} <\text{seconds}>$


Minimum value: 1 second

Default: 120 seconds

Format: Integer

$\text{max\_load} <\text{load}> [\text{suspend}]$

Defines the load above which the vnode is considered to be busy. Used with the \$\text{ideal\_load} directive. No new jobs are started on a busy vnode.

The optional suspend directive tells PBS to suspend jobs running on the vnode if the load average exceeds the $\text{max\_load}$ number, regardless of the source of the load (PBS and/or logged-in users). Without this directive, PBS will not suspend jobs due to load.

We recommend setting load to a value that is slightly higher than the number of CPUs, for example .25 + ncpus.

Example:

$\text{max\_load} 3.5$

Format: Float

Default: number of CPUs on machine
memreserved <megabytes>
   
   **Deprecated.** The amount of per-vnode memory reserved for system overhead. This much memory is deducted from the value of `resources_available.mem` for each vnode managed by this MoM.

   For example,
   ```
   memreserved 16
   ```

   Default: **0MB**

$min_check_poll <seconds>
   
   Minimum time between polling cycles, in seconds. Must be greater than zero and less than `$max_check_poll`. See "Configuring MoM Polling Cycle" on page 45 in the PBS Professional Administrator’s Guide. Minimum recommended value: **10 seconds**.

   Minimum value: **1 second**

   Default: **10 seconds**

   Format: **Integer**

pbs_accounting_workload_mgmt <value>
   
   Controls whether CSA accounting is enabled. Name does not start with dollar sign. If set to "1", "on", or "true", CSA accounting is enabled. If set to "0", "off", or "false", accounting is disabled. Cray only. Requires CLE 5.2.

   Default: **"true"**; enabled

$prologalarm <timeout>
   
   Defines the maximum number of seconds the prologue and epilogue may run before timing out.

   Example:
   ```
   $prologalarm 30
   ```

   Format: **Integer**

   Default: **30 seconds**

$reject_root_scripts <True|False>
   
   When set to **True**, MoM won’t acquire any new hook scripts, and MoM won’t run job scripts that would execute as root or Admin. However, MoM will run previously-acquired hooks that run as root.

   Format: **Boolean**

   Default: **False**

$restart_background <True|False>
   
   Controls how MoM runs a restart script after checkpointing a job. When this option is set to **True**, MoM forks a child which runs the restart script. The child returns when all restarts for all the local tasks of the job are done. MoM does not block on the restart. When this option is set to **False**, MoM runs the restart script and waits for the result.

   Format: **Boolean**

   Default: **False**
$restart_transmogrify <True | False>
Controls how MoM runs a restart script after checkpointing a job.
When this option is set to True, MoM runs the restart script, replacing the session ID of the original task’s top process with the session ID of the script.
When this option is set to False, MoM runs the restart script and waits for the result. The restart script must restore the original session ID for all the processes of each task so that MoM can continue to track the job.
When this option is set to False and the restart uses an external command, the configuration parameter restart_background is ignored and treated as if it were set to True, preventing MoM from blocking on the restart.
Format: Boolean
Default: False

$restrict_user <True | False>
Controls whether users not submitting jobs have access to this machine. If value is True, restrictions are applied.
See $restrict_user_exceptions and $restrict_user_maxsysid.
Not supported on Windows.
Format: Boolean
Default: False

$restrict_user_exceptions <user list>
Comma-separated list of users who are exempt from access restrictions applied by $restrict_user. Leading spaces within each entry are allowed. Maximum of 10 names.

$restrict_user_maxsysid <value>
Any user with a numeric user ID less than or equal to value is exempt from restrictions applied by $restrict_user.
If $restrict_user is True and no value exists for $restrict_user_maxsysid, PBS looks in /etc/login.defs, if it exists, for the value. Otherwise the default is used.
Format: Integer
Default: 999

$restricted <hostname>
The hostname is added to the list of hosts which are allowed to connect to MoM without being required to use a privileged port. Queries from the hosts in the restricted list are only allowed access to information internal to this host, such as load average, memory available, etc. They may not run shell commands.
Hostnames can be wildcarded. For example, to allow queries from any host from the domain “xyz.com”:
$restricted *.xyz.com

$suspendsig <suspend signal> [resume signal]
Alternate signal suspend signal is used to suspend jobs instead of SIGSTOP. Optional resume signal is used to resume jobs instead of SIGCONT.
MoM Parameters

$\text{tmpdir} \ <\text{directory}>$

Location where each job’s scratch directory will be created.

PBS creates a temporary directory for use by the job, not by PBS. PBS creates the directory before the job is run and removes the directory and its contents when the job is finished. It is scratch space for use by the job. Permission must be 1777 on Linux, writable by Everyone on Windows.

Example:

```
$\text{tmpdir} /\text{memfs}
```

Default on Linux: /var/tmp

Default on Windows: value of the TMP environment variable

$\text{usecp} \ <\text{hostname}:source prefix}> \ <\text{destination prefix}>

MoM uses /bin/cp to deliver output files when the destination is a network mounted file system, or when the source and destination are both on the local host, or when the source prefix can be replaced with the destination prefix on hostname. Both source prefix and destination prefix are absolute pathnames of directories, not files.

Overrides PBS_RCP and PBS_SCP.

Use trailing slashes on both the source and destination. For example:

```
$\text{usecp} \text{HostA:/users/work/myproj/} \ /\text{sharedwork/proj_results/}
```

$\text{vnodedef_additive}$

Specifies whether MoM considers a vnode that appeared previously either in the inventory or in a vnode definition file, but that does not appear now, to be in her list of vnodes.

When $\text{vnodedef_additive}$ is True, MoM treats missing vnodes as if they are still present, and continues to report them as if they are present. This means that the server does not mark missing vnodes as stale.

When $\text{vnodedef_additive}$ is False, MoM does not list missing vnodes, the server’s information is brought up to date with the inventory and vnode definition files, and the server marks missing vnodes as stale.

PBS automatically sets the value of the $\text{vnodedef_additive}$ MoM configuration option to False on any MoM on a login node.

Visible in configuration file on Cray only.

Format: Boolean

Default for MoM on Cray login node: False

$\text{wallmult} \ <\text{factor}>$

Each job’s walltime usage is multiplied by factor. For example:

```
$\text{wallmult} 1.5
```

3.2.3 Static MoM Resources

Static resources local to the vnode are described one resource to a line, with a name and value separated by white space. For example, tape drives of different types could be specified by:

```
tape3480 4
tape3420 2
tapedat 1
tape8mm 1
```
This chapter lists scheduler configuration parameters. These parameters are found in each scheduler’s configuration file, PBS_HOME/sched_priv/sched_config.

**4.1 Format of Scheduler Configuration File**

**4.1.1 Parameters with Separate Primetime and Non-primetime Specification**

If a scheduler parameter can be specified separately for primetime and non-primetime, the format for the parameter is the following:

\[ \text{name: value [prime | non_prime | all | none]} \]

- The name field cannot contain any whitespace.
- The value field may contain whitespace if the string is double-quoted. value can be: True | False | <number> | <string>. “True” and “False” are not case-sensitive.
- The third field allows you to specify that the setting is to apply during primetime, non-primetime, all the time, or none of the time. A blank third field is equivalent to “all” which means that it applies to both primetime and non-primetime.


**4.1.2 Parameters without Separate Primetime and Non-primetime Specification**

If a scheduler parameter cannot be specified separately for primetime and non-primetime, the format for the parameter is the same as the above, except that there is no third field.

**4.1.3 Format Details**

- Each entry must be a single, unbroken line.
- Entries must be quoted if they contain whitespace.
- Any line starting with a “#” is a comment, and is ignored.

**4.1.4 Editing Configuration Files Under Windows**

When you edit any PBS configuration file, make sure that you put a newline at the end of the file. The Notepad application does not automatically add a newline at the end of a file; you must explicitly add the newline.
4.2 Configuration Parameters

backfill

Deprecated. Use the backfill_depth queue/server attribute instead. Toggle that controls whether PBS uses backfilling. If this is set to True, this scheduler attempts to schedule smaller jobs around higher-priority jobs when using strict_ordering, as long as running the smaller jobs won’t change the start time of the jobs they were scheduled around. This scheduler chooses jobs in the standard order, so other high-priority jobs will be considered first in the set to fit around the highest-priority job.

When this parameter is True and help_starving_jobs is True, this scheduler backfills around starving jobs.

Can be used with strict_ordering and help_starving_jobs

Format: Boolean
Default: True all

backfill_prime

This scheduler will not run jobs which would overlap the boundary between primetime and non-primetime. This assures that jobs restricted to running in either primetime or non-primetime can start as soon as the time boundary happens.

See also prime_spill, prime_exempt_anytime_queues.

Format: Boolean
Default: False all

by_queue

If set to True, all jobs that can be run from the highest-priority queue are run, then any jobs that can be run from the next queue are run, and so on. Queues are ordered highest-priority first. If by_queue is set to False, all jobs are treated as if they are in one large queue. The by_queue parameter is overridden by the round_robin parameter when round_robin is set to True.

See "Examining Jobs Queue by Queue" on page 106 in the PBS Professional Administrator’s Guide.

Format: Boolean
Default: True all

cpus_per_ssinode

Deprecated. Such configuration now occurs automatically.

dedicated_prefix

Queue names with this prefix are treated as dedicated queues, meaning jobs in that queue are considered for execution only when the system is in dedicated time as specified in the configuration file PBS_HOME/sched_priv/dedicated_time.

See "Dedicated Time" on page 124 in the PBS Professional Administrator’s Guide.

Format: String
Default: ded

fair_share

Enables the fairshare algorithm, and turns on usage collecting. Jobs will be selected based on a function of their recent usage and priority (shares). Not a prime option.

See "Using Fairshare" on page 136 in the PBS Professional Administrator’s Guide.

Format: Boolean
Default: False all
fairshare_decay_factor
Decay multiplier for fairshare usage reduction. Each decay period, the usage is multiplied by this value. Valid values: between 0 and 1, not inclusive. Not a prime option.
Format: Float
Default: 0.5

fairshare_decay_time
Time between fairshare usage decay operations. Not a prime option.
Format: Duration
Default: 24:00:00

fairshare_entity
Specifies the entity for which fairshare usage data will be collected. Can be one of “euser”, “egroup”, “Account_Name”, “queue”, or “egroup:euser”. Not a prime option.
Format: String
Default: euser

fairshare_enforce_no_shares
If this option is enabled, jobs whose entity has zero shares will never run. Requires fair_share parameter to be enabled. Not a prime option.
Format: Boolean
Default: False

fairshare_usage_res
Specifies the mathematical formula to use in fairshare calculations. Is composed of PBS resources as well as mathematical operators that are standard Python operators and/or those in the Python math module. When using a PBS resource, if resources_used.<resource name> exists, that value is used. Otherwise, the value is taken from Resource_List.<resource name>. Not a prime option.
See "Tracking Resource Usage" on page 139 in the PBS Professional Administrator’s Guide.
Format: String
Default: cput

half_life
Deprecated (as of 13.0).
The half-life for fairshare usage; after the amount of time specified, the fairshare usage is halved. Requires that fair_share parameter be enabled. Not a prime option.
See "Using Fairshare" on page 136 in the PBS Professional Administrator’s Guide.
Format: Duration
Default: 24:00:00

help_starving_jobs
Setting this option enables starving job support. Once jobs have waited for the amount of time given by max_starve they are considered starving. If a job is considered starving, no lower-priority jobs will run until the starving job can be run, unless backfilling is also used. To use this option, the max_starve configuration parameter needs to be set as well. See also max_starve, and the server's backfill_depth and eligible_time_enable attributes.
At each scheduler iteration, PBS calculates estimated.start_time and estimated.exec_vnode for starving jobs being backfilled around.
Format: Boolean
Default: True all
job_sort_key

Specifies how jobs should be sorted. job_sort_key can be used to sort using either (a) resources or (b) special case sorting routines. Multiple job_sort_key entries can be used, one to a line, in which case the first entry will be the primary sort key, the second will be used to sort equivalent items from the first sort, etc. This attribute is overridden by the job_sort_formula attribute. If both are set, job_sort_key is ignored and an error message is printed.

Syntax:

job_sort_key: "<resource name> HIGH|LOW"
job_sort_key: "fairshare_perc HIGH|LOW"
job_sort_key: "job_priority HIGH|LOW"

Options: One of the following is required.

HIGH
Specifies descending sort.

LOW
Specifies ascending sort.

There are three special case sorting routines, which can be used instead of resource name:

Table 4-1: Special Sorting in job_sort_key

<table>
<thead>
<tr>
<th>Special Sort</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fairshare_perc HIGH</td>
<td>Sort based on how much fairshare percentage the entity deserves, based on the values in the resource_group file. If user A has more priority than user B, all of user A's jobs will always be run first. Past history is not used. For calculation, see &quot;Computing Target Usage for Each Vertex (fairshare_perc)&quot; on page 141 in the PBS Professional Administrator’s Guide. This should only be used if entity share (strict priority) sorting is needed. See &quot;Sorting Jobs by Entity Shares (Was Strict Priority)&quot; on page 129 in the PBS Professional Administrator’s Guide. Incompatible with fair_share scheduling parameter being True.</td>
</tr>
<tr>
<td>job_priority HIGH</td>
<td>LOW</td>
</tr>
<tr>
<td>sort_priority HIGH</td>
<td>LOW</td>
</tr>
</tbody>
</table>

The following example illustrates how to sort jobs so that those with high CPU count come first:

job_sort_key: "ncpus HIGH" all

The following example shows how to sort jobs so that those with lower memory come first:

job_sort_key: "mem LOW" prime

Format: Quoted string
Default: Not in force

key

Deprecated. Use job_sort_key.
load_balancing
When set to True, this scheduler takes into account the load average on vnodes as well as the resources listed in the resources line in sched_config. Load balancing can result in overloaded CPUs.
Format: Boolean
Default: False

load_balancing_rr
Deprecated. To duplicate this setting, enable load_balancing and set smp_cluster_dist to round_robin.

log_filter
Defines which event types to keep out of this scheduler’s logfile. The value should be set to the bitwise OR of the event classes which should be filtered. A value of 0 specifies maximum logging.
See "Specifying Scheduler Log Events" on page 537 in the PBS Professional Administrator’s Guide.
Format: Integer
Default: 3328

max_starve
The amount of time before a job is considered starving. This variable is used only if help_starving_jobs is set.
Upper limit: None
Format: Duration
Default: 24:00:00

mem_per_ssinode
Deprecated. Such configuration now occurs automatically.

mom_resources
This option is used to query the MoMs to set the value of resources_available.<resource name> where resource name is a site-defined resource. Each MoM is queried with the resource name and the return value is used to replace resources_available.<resource name> on that vnode. On a multi-vnoded machine with a natural vnode, all vnodes share anything set in mom_resources.
Format: String
Default: Unset

node_sort_key
Defines sorting on resource or priority values on vnodes. Resource must be numerical, for example, long or float. Up to 20 node_sort_key entries can be used, in which case the first entry will be the primary sort key, the second will be used to sort equivalent items from the first sort, etc.
Syntax:
node_sort_key: <resource name> | sort_priority <HIGH | LOW>
node_sort_key: <resource name> <HIGH | LOW> <total | assigned | unused>
where
total
Use the resources_available value. This is the default setting when sorting on a resource.
assigned
Use the resources_assigned value.
unused
Use the value given by resources_available - resources_assigned.
**sort_priority**

Sort vnodes by the value of the vnode priority attribute.

When sorting on a resource, the default third field is “total”.

See "Sorting Vnodes on a Key" on page 221 in the PBS Professional Administrator’s Guide.

Format: *String*

Default: `node_sort_key: sort_priority HIGH all`

**nonprimetime_prefix**

Queue names which start with this prefix are treated as non-primetime queues. Jobs in these queues run only during non-primetime. Primetime and non-primetime are defined in the holidays file.

See "Using Primetime and Holidays" on page 190 in the PBS Professional Administrator’s Guide.

Format: *String*

Default: `np_`

**peer_queue**

Defines the mapping of a pulling queue to a furnishing queue for peer scheduling. Maximum number is 50 peer queues per scheduler.

See "Peer Scheduling" on page 163 in the PBS Professional Administrator’s Guide.

Format: *String*

Default: Unset

**preemptive_sched**

Enables job preemption.


Format: *String*

Default: `True all`

**preempt_checkpoint**

*Deprecated.* Add “C” to preempt_order parameter.

**preempt_fairshare**

*Deprecated.* Add “fairshare” to preempt_prio parameter.

**preempt_order**

Defines the order of preemption methods which this scheduler uses on jobs. This order can change depending on the percentage of time remaining on the job. The ordering can be any combination of *S*, *C*, and *R*:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>S</em></td>
<td>suspend</td>
</tr>
<tr>
<td><em>C</em></td>
<td>checkpoint</td>
</tr>
<tr>
<td><em>R</em></td>
<td>requeue</td>
</tr>
</tbody>
</table>

Usage: an ordering (SCR) optionally followed by a percentage of time remaining and another ordering. Must be a quoted list(""").
For example, PBS should first attempt to use suspension to preempt a job, and if that is unsuccessful, requeue the job:

`preempt_order: “SR”`

For example, if the job has between 100% and 81% of requested time remaining, first try to suspend the job, then try checkpoint, then requeue. If the job has between 80% and 51% of requested time remaining, attempt suspend, then checkpoint. Between 50% and 0% time remaining, just attempt to suspend the job:

`preempt_order: “SCR 80 SC 50 S”`

**Format:** *Quoted list*

**Default:** SCR

**preempt_prio**

Specifies the ordering of priority for different preemption levels. Two or more job types may be combined at the same priority level with a plus sign (“+”) between them, using no whitespace. Comma-separated preemption levels are evaluated left to right, with higher priority to the left. The table below lists the six preemption levels. Any level not specified in the `preempt_prio` list is ignored.

**Table 4-3: Preemption Levels**

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>express_queue</td>
<td>Jobs in express queues preempt other jobs. See <code>preempt_queue_prio</code>. Does not require by_queue to be True.</td>
</tr>
<tr>
<td>starving_jobs</td>
<td>When a job becomes starving it can preempt other jobs.</td>
</tr>
<tr>
<td>fairshare</td>
<td>When the entity owning a job exceeds its fairshare limit.</td>
</tr>
<tr>
<td>queue_softlimits</td>
<td>Jobs which are over their queue soft limits</td>
</tr>
<tr>
<td>server_softlimits</td>
<td>Jobs which are over their server soft limits</td>
</tr>
<tr>
<td>normal_jobs</td>
<td>The preemption level into which a job falls if it does not fit into any other specified level.</td>
</tr>
</tbody>
</table>

For example, starving jobs have the highest priority, then normal jobs, and jobs whose entities are over their fairshare limit are third highest:

`preempt_prio: “starving_jobs, normal_jobs, fairshare”`

For example, starving jobs whose entities are also over their fairshare limit are lower priority than normal jobs:

`preempt_prio: “normal_jobs, starving_jobs+fairshare”`

**Format:** *Quoted list*

**Default:** express_queue, normal_jobs

**preempt_queue_prio**

Specifies the minimum queue priority required for a queue to be classified as an express queue. Express queues do not require by_queue to be True.

**Format:** *Integer*

**Default:** 150

**preempt_requeue**

**Deprecated.** Add an “R” to `preempt_order` parameter.
**preempt_sort**

Specifies whether jobs most eligible for preemption are sorted according to their start times.

If set to "*min_time_since_start*", first job preempted will be that with most recent start time.

If not set, meaning that this parameter is commented out, preempted job will be that with longest running time.

Must be commented out in order to be unset.

Allowable values: "*min_time_since_start*", or no *preempt_sort* setting.

See "Sorting Within Preemption Level" on page 186 in the PBS Professional Administrator’s Guide.

Format: *String*

Default: *min_time_since_start*

**preempt_starving**

*Deprecated*. Add "starving_jobs" to *preempt prio* parameter.

**preempt_suspend**

*Deprecated*. Add an “S” to *preempt order* parameter.

**primetime_prefix**

Queue names starting with this prefix are treated as primetime queues. Jobs in these queues run only during primetime. Primetime and non-primetime are defined in the *holidays* file.

See "Using Primetime and Holidays" on page 190 in the PBS Professional Administrator’s Guide.

Format: *String*

Default: *p_*

**prime_exempt_anytime_queues**

Determines whether *anytime* queues are controlled by *backfill_prime*.

If set to *True*, jobs in an *anytime* queue are not prevented from running across a primetime/non-primetime or non-primetime/primetime boundary.

If set to *False*, the jobs in an *anytime* queue may not cross this boundary, except for the amount specified by their *prime_spill* setting.

See also *backfill_prime*, *prime_spill*.

Format: *Boolean*

Default: *False*

**prime_spill**

Specifies the amount of time a job can spill over from non-primetime into primetime or from primetime into non-primetime. This option can be separately specified for primetime and non-primetime. This option is only meaningful if *backfill_prime* is *True*.

See also *backfill_prime*, *prime_exempt_anytime_queues*.

For example, non-primetime jobs can spill into primetime by 1 hour:

```
prime_spill: 1:00:00 prime
```

For example, jobs in either prime/non-prime can spill into the other by 1 hour:

```
prime_spill: 1:00:00 all
```

Format: *Duration*

Default: *00:00:00*

**provision_policy**

Specifies how vnodes are selected for provisioning. Can be set by Manager only; readable by all. Can be set to one of the following:
avoid_provision
PBS first tries to satisfy the job's request from free vnodes that already have the requested AOE instantiated. PBS uses `node_sort_key` to sort these vnodes.

If PBS cannot satisfy the job's request using vnodes that already have the requested AOE instantiated, PBS uses the server's `node_sort_key` to select the free vnodes that must be provisioned in order to run the job, choosing from any free vnodes, regardless of which AOE is instantiated on them.

Of the selected vnodes, PBS provisions any that do not have the requested AOE instantiated on them.

ggressive_provision
PBS selects vnodes to be provisioned without considering which AOE is currently instantiated.

PBS uses the server's `node_sort_key` to select the vnodes on which to run the job, choosing from any free vnodes, regardless of which AOE is instantiated on them. Of the selected vnodes, PBS provisions any that do not have the requested AOE instantiated on them.

Format: `String`
Default: `aggressive_provision`

resources
Specifies those resources which are not to be over-allocated, or if Boolean are to be honored, when scheduling jobs. Vnode-level Boolean resources are automatically honored and do not need to be listed here. Limits are set by setting `resources_available.<resource name>` on vnodes, queues, and the server. A scheduler considers numeric (integer or float) items as consumable resources and ensures that no more are assigned than are available (e.g. `ncpus` or `mem`). Any string resources are compared using string comparisons. If "host" is not added to the `resources` line, when the user submits a job requesting a specific vnode in the following syntax:

```
qsub -l select=host=vnodeName
```
the job will run on any host.

Format: `String`
Default: `ncpus, mem, arch, host, vnode, aoe`

resource_unset_infinite
Resources in this list are treated as infinite if they are unset. Cannot be set differently for primetime and non-primetime.

Example:
```
resource_unset_infinite: "vmem, foo_licenses"
```

Format: `Comma-delimited list of resources`
Default: Empty list

round_robin
If set to `True`, this scheduler considers one job from the first queue, then one job from the second queue, and so on in a circular fashion. The queues are ordered with the highest-priority queue first. Each scheduling cycle starts with the same highest-priority queue, which will therefore get preferential treatment.

If there are groups of queues with the same priority, and this parameter is set to `True`, this scheduler round-robins through each group of queues before moving to the next group.

If `round_robin` is set to `False`, this scheduler considers jobs according to the setting of the `by_queue` parameter.

When `True`, overrides the `by_queue` parameter.

Format: `Boolean`
Default: `False all`
server_dyn_res

Directs this scheduler to replace the server’s `resources_available` values with new values returned by a site-specific external program.


Format: `String`
Default: Unset

smp_cluster_dist

**Deprecated** (12.2). Specifies how single-host jobs should be distributed to all hosts of the complex.

Options:
- `pack`
  - Keep putting jobs onto one host until it is full and then move on to the next.
- `round_robin`
  - Put one job on each vnode in turn before cycling back to the first one.
- `lowest_load`
  - Put the job on the lowest-loaded host.


Format: `String`
Default: `pack all`

sort_by

**Deprecated.** Use `job_sort_key`.

sort_queues

**Deprecated** (12.2). If set to `True` queues are sorted so that the highest-priority queues are considered first. Queues are sorted by each queue’s priority attribute. The queues are sorted in a descending fashion, that is, a queue with priority 6 comes before a queue with priority 3.

When set to `False`, queues are not sorted.

This is a prime option, which means it can be selectively applied to primetime or non-primetime.

The sorted order of queues is not taken into consideration unless `by_queue` is set to `True`.

See "Sorting Queues into Priority Order" on page 218 in the PBS Professional Administrator’s Guide.

Format: `Boolean`
Default: `True ALL`

strict_fifo

**Deprecated.** Use `strict_ordering`.

strict_ordering

Specifies that jobs must be run in the order determined by whatever sorting parameters are being used. This means that a job cannot be skipped due to resources required not being available. If a job due to run next cannot run, no job will run, unless backfilling is used, in which case jobs can be backfilled around the job that is due to run next.

See "FIFO with Strict Ordering" on page 147 in the PBS Professional Administrator’s Guide.

Example line in `PBS_HOME/sched_priv/sched_config`:

```
  strict_ordering: True ALL
```

Format: `Boolean`
Default: `False all`
sync_time

- **Deprecated.** The amount of time between writing the fairshare usage data to disk. Requires `fair_share` to be enabled.

  Format: *Duration*
  Default: 1:00:00

unknown_shares

The number of shares for the `unknown` group. These shares determine the portion of a resource to be allotted to that group via fairshare. Requires `fair_share` to be enabled.

See "Using Fairshare" on page 136 in the PBS Professional Administrator’s Guide.

Format: *Integer*

Default: The unknown group gets 0 shares
This chapter lists all of the built-in PBS resources. For information on setting, viewing, and using resources, see "Using PBS Resources" on page 225 in the PBS Professional Administrator’s Guide.

### 5.1 Resource Data Types

Data types for resources are described in section 7.1, “List of Formats”, on page 363.

### 5.2 Viewing Resource Information

See "Viewing Resource Information" on page 310 in the PBS Professional Administrator’s Guide.
5.3 Attributes where Resources Are Tracked

Resources are tracked in the following attributes:

Table 5-1: Attributes Where Resources Are Tracked

<table>
<thead>
<tr>
<th>Resource Being Tracked</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount of each resource available for use at the object (server, queue, vnode)</td>
<td>resources_available.&lt;resource name&gt;</td>
</tr>
<tr>
<td></td>
<td>resources_available.&lt;resource name&gt;</td>
</tr>
<tr>
<td>Amount of each resource allocated to jobs running and exiting at the object</td>
<td>resources_assigned.&lt;resource name&gt;</td>
</tr>
<tr>
<td>(server, queue, vnode)</td>
<td>resources_assigned.&lt;resource name&gt;</td>
</tr>
<tr>
<td>Amount of each resource used by the job</td>
<td>resources_used.&lt;resource name&gt;</td>
</tr>
<tr>
<td>Amount of each job-wide resource that is assigned to any job that does not</td>
<td>resources_default.&lt;resource name&gt;</td>
</tr>
<tr>
<td>explicitly request the resource</td>
<td>resources_default.&lt;resource name&gt;</td>
</tr>
<tr>
<td>Amount of each host-level resource that is assigned to each chunk of any job</td>
<td>default_chunk.&lt;resource name&gt;</td>
</tr>
<tr>
<td>where that does not explicitly request the resource</td>
<td>default_chunk.&lt;resource name&gt;</td>
</tr>
<tr>
<td>List of resources requested by the object (job or reservation)</td>
<td>Resource_List.&lt;resource name&gt;</td>
</tr>
<tr>
<td>List of chunks for the job. Each chunk shows the name of the vnode from</td>
<td>exec_vnode</td>
</tr>
<tr>
<td>which it is taken along with the host-level, consumable resources allocated</td>
<td>resv_nodes</td>
</tr>
<tr>
<td>from that vnode.</td>
<td>resv_nodes</td>
</tr>
</tbody>
</table>

5.4 Resource Table Format

In the following tables, the columns contain the following information:
List of Built-in Resources

Name
The name of the resource

Description
A description of the resource’s function

Format
The resource’s format

Scope
Some resources are either:
• Job-wide and can be requested only outside of a select statement
• Host-level and can be requested only inside of a select statement

Consumable
A resource is consumable if use of this resource by a job reduces the amount available to other jobs

Val/Opt
If the resource can take only specific values or options, each is listed here

Value/Option Description
If the resource can take only specific values or options, the behavior of each value or option is described here

Default Value
The resource’s default value, if any

Python Type
The resource’s Python type

Platform
Platform where available
## 5.5 Resources Built Into PBS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>accelerator</td>
<td>Indicates whether this vnode is associated with an accelerator. Used for requesting accelerators. On Cray, this resource exists only when there is at least one associated accelerator.</td>
<td>Boolean</td>
<td>Host-level</td>
<td>No</td>
<td>True</td>
<td>On Cray, this is set to True when there is at least one associated accelerator whose state is UP.</td>
<td>False</td>
<td>bool</td>
<td></td>
</tr>
<tr>
<td>accelerator_memory</td>
<td>Indicates amount of memory for accelerator(s) associated with this vnode. On Cray, PBS sets this resource only on vnodes with at least one accelerator with state = UP. For Cray, PBS sets this resource on the 0th NUMA node (the vnode with PBScray-seg=0), and the resource is shared by other vnodes on the compute node. For example, on vnodeA_2_0: resources_available.accelerator_memory=4196mb. On vnodeA_2_1: resources_available.accelerator_memory=@vnodeA_2_0. A scheduler rounds all resources of type size up to the nearest kb.</td>
<td>Size</td>
<td>Host-level</td>
<td>Yes</td>
<td></td>
<td>No default</td>
<td></td>
<td>pbs.size</td>
<td></td>
</tr>
<tr>
<td>accelerator_model</td>
<td>Indicates model of the accelerator(s) associated with this vnode. On Cray, PBS sets this resource only on vnodes with at least one accelerator with state = UP.</td>
<td>String</td>
<td>Host-level</td>
<td>No</td>
<td></td>
<td>No default</td>
<td></td>
<td>str</td>
<td></td>
</tr>
</tbody>
</table>
## List of Built-in Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>aoe</td>
<td>List of AOE (Application Operating Environments) that can be instantiated on this vnode. Case-sensitive. An AOE is the environment that results from provisioning a vnode. Each job can request at most one AOE. Cannot be set on server's host.</td>
<td>string array</td>
<td>Host-level</td>
<td>No</td>
<td>Allowable values are site-dependent.</td>
<td>No default</td>
<td>str</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arch</td>
<td>System architecture. One architecture can be defined for a vnode. One architecture can be requested per vnode. Allowable values and effect on job placement are site-dependent. The <code>resources_available.arch</code> resource is the value reported by MoM unless explicitly set by the administrator.</td>
<td>String</td>
<td>Host-level</td>
<td>No</td>
<td>linux</td>
<td>Linux</td>
<td>No default</td>
<td>str</td>
<td>Linux</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>linux_cpuset</td>
<td>Linux with cpusets</td>
<td></td>
<td>Linux with cpusets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XT</td>
<td>CLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cput</td>
<td>Amount of CPU time used by the job for all processes on all vnodes. Establishes a job-wide resource limit.</td>
<td>Duration</td>
<td>Job-wide</td>
<td>No</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.duration</td>
<td></td>
</tr>
<tr>
<td>energy</td>
<td>The energy used by a job. Set by PBS.</td>
<td>Float. Units: kWh</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eoe</td>
<td>Stands for “Energy Operational Environment”. When set on a vnode in <code>resources_available.eoe</code>, contains the list of available power profiles. When set for a job in <code>Resource_List.eoe</code>, can contain at most one power profile. (A job can request only one power profile.) Automatically added to <code>resources: line in sched_config</code>.</td>
<td>string array</td>
<td>No</td>
<td></td>
<td>For <code>resources_available.eoe: unset</code></td>
<td>For <code>Resource_List.eoe: no default</code></td>
<td>str</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# List of Built-in Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val/Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>exec_vnode</td>
<td>The vnodes that PBS estimates this job will use. This is not the job’s exec_vnode attribute. This appears only in job’s estimated attribute. Cannot be requested for a job; used for reporting only. Read-only.</td>
<td>String</td>
<td>Job-wide</td>
<td>No</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td></td>
</tr>
<tr>
<td>file</td>
<td>Size of any single file that may be created by the job. A scheduler rounds all resources of type size up to the nearest kb.</td>
<td>Size</td>
<td>Job-wide</td>
<td>Yes</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.size</td>
<td></td>
</tr>
<tr>
<td>hbmem</td>
<td>High-bandwidth memory. Available only on some architectures such as Xeon Phi KNL.</td>
<td>Size</td>
<td>Host-level</td>
<td>Yes</td>
<td></td>
<td>Values must be greater than or equal to zero.</td>
<td>No default</td>
<td>pbs.size</td>
<td>Xeon Phi KNL</td>
</tr>
<tr>
<td>host</td>
<td>Name of execution host. Cannot be changed. Site-dependent.</td>
<td>String</td>
<td>Host-level</td>
<td>No</td>
<td></td>
<td></td>
<td>Automatically set to the short form of the hostname in the Mom attribute. On Cray compute node, set to <code>&lt;mpp_host&gt;_&lt;nid&gt;</code></td>
<td>str</td>
<td></td>
</tr>
<tr>
<td>max_walltime</td>
<td>Maximum walltime allowed for a shrink-to-fit job. Job’s actual wall-time is between max_walltime and min_walltime. PBS sets walltime for a shrink-to-fit job. If max_walltime is specified, min_walltime must also be specified. Cannot be used for resources_min or resources_max. Cannot be set on job arrays or reservations.</td>
<td>Duration</td>
<td>Job-wide</td>
<td>No</td>
<td></td>
<td>Must be greater than or equal to min_walltime.</td>
<td>5 years</td>
<td>pbs.duration</td>
<td></td>
</tr>
</tbody>
</table>
### List of Built-in Resources

#### Chapter 5

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem</td>
<td>Amount of physical memory i.e. workingset allocated to the job, either job-wide or host-level. A scheduler rounds all resources of type size up to the nearest kb.</td>
<td>Size</td>
<td>Either job-wide or host-level. Can be requested only inside of a select statement.</td>
<td>Yes</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.size</td>
<td></td>
</tr>
<tr>
<td>min_walltime</td>
<td>Minimum walltime allowed for a shrink-to-fit job. When min_walltime is specified, job is a shrink-to-fit job. If this attribute is set, PBS sets the job’s walltime. Job’s actual walltime is between max_walltime and min_walltime. Cannot be used for resources_min or resources_max. Cannot be set on job arrays or reservations.</td>
<td>Duration</td>
<td>Job-wide</td>
<td>No</td>
<td>Must be less than or equal to max_walltime.</td>
<td>No default</td>
<td>pbs.duration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mpiprocs</td>
<td>Number of MPI processes for this chunk. Cannot use sum from chunks as job-wide limit. The number of lines in PBS_NODEFILE is the sum of the values of mpiprocs for all chunks requested by the job. For each chunk with mpiprocs=P, the host name for that chunk is written to the PBS_NODEFILE P times.</td>
<td>Integer</td>
<td>Host-level</td>
<td></td>
<td></td>
<td></td>
<td>If ncpus &gt;0: 1</td>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>
### List of Built-in Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val/Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>naccelerators</td>
<td>Number of accelerators on the host. PBS sets this resource to the number of accelerators with state = <em>UP</em>. On Cray, PBS sets this resource only on vnodes whose hosts have at least one accelerator with state = <em>UP</em>. For Cray, PBS sets this resource on the 0th NUMA node (the vnode with PBScrayseg=0), and the resource is shared by other vnodes on the compute node. For example, on vnodeA_2_0: resources_available.naccelerators=1 On vnodeA_2_1: resources_available.naccelerators=@vnodeA_2_0</td>
<td>Integer</td>
<td>Host-level</td>
<td>Yes</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>nchunk</td>
<td>Number of chunks requested between plus symbols in a select statement. For example, if the select statement is <code>-lselect 4:ncpus=2+12:ncpus=8</code>, the value of nchunk for the first part is 4, and for the second part it is 12. The nchunk resource cannot be named in a select statement; it can only be specified by placing a number before the colon, as in the above example. When the number is omitted, nchunk is 1. This resource can be used to specify the default number of chunks at the server or queue. Example: <code>set queue myqueue default_chunk.nchunk=2</code> This resource cannot be used in server and queue resources_min and resources_max.</td>
<td>Integer</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td>1</td>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>
## List of Built-in Resources

### Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncpus</td>
<td>Number of processors.</td>
<td>Integer</td>
<td>Host-level</td>
<td>Yes</td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nice</td>
<td>Nice value with which the job is to be run. Host-dependent.</td>
<td>Integer</td>
<td>Job-wide</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nodect</td>
<td><strong>Deprecated.</strong> Number of chunks in resource request from selection directive, or number of hosts requested from node specification. Read-only.</td>
<td>Integer</td>
<td>Job-wide</td>
<td></td>
<td></td>
<td>1</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nodes</td>
<td><strong>Deprecated.</strong> Number of hosts requested.</td>
<td>Integer</td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ompthreads</td>
<td>Number of OpenMP threads for this chunk. Cannot use sum from chunks as job-wide limit. For the MPI process with rank 0, the environment variables NCPUS and OMP_NUM_THREADS are set to the value of ompthreads. For other MPI processes, behavior is dependent on MPI implementation.</td>
<td>Integer</td>
<td>Host-level</td>
<td>No</td>
<td></td>
<td>Value of ncpus</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PBScrayhost</td>
<td>Used to differentiate a Cray system, containing ALPS, login nodes running PBS MoMs, and compute nodes, from a separate Cray system with a separate ALPS.</td>
<td>String</td>
<td></td>
<td>No</td>
<td></td>
<td>Value of mpp_host for this system</td>
<td>str</td>
<td>Cray only</td>
<td></td>
</tr>
</tbody>
</table>
### List of Built-in Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBScraylabel_&lt;label name&gt;</td>
<td>Tracks labels applied to compute nodes. For each label on a compute node, PBS creates a custom resource whose name is a concatenation of <code>PBScraylabel_</code> and the name of the label. Name format: <code>PBScraylabel_&lt;label name&gt;</code> For example, if the label name is <code>Blue</code>, the name of this resource is <code>PBScraylabel_Blue</code>.</td>
<td>Boolean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PBS sets the value of the resource to <code>True</code> on all vnodes representing the compute node.</td>
<td>bool</td>
<td>Cray only</td>
</tr>
<tr>
<td>PBScraynid</td>
<td>Tracks the node ID of the associated compute node. All vnodes representing a particular compute node share a value for <code>PBScraynid</code>.</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The value of <code>PBScraynid</code> is set to the value of <code>node_id</code> for this compute node.</td>
<td>str</td>
<td>Cray only</td>
</tr>
<tr>
<td>PBScrayorder</td>
<td>Tracks the order in which compute nodes are listed in the Cray inventory. All vnodes associated with a particular compute node share a value for <code>PBScrayorder</code>. Do not use this resource in a resource request.</td>
<td>Integer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Vnodes for the first compute node listed are assigned a value of 1 for <code>PBScrayorder</code>. The vnodes for each subsequent compute node listed are assigned a value one greater than the previous value.</td>
<td>int</td>
<td>Cray only</td>
</tr>
<tr>
<td>PBScrayseg</td>
<td>Not used.</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>Cray only</td>
</tr>
<tr>
<td>pcput</td>
<td>Amount of CPU time allocated to any single process in the job. Establishes a per-process resource limit.</td>
<td>Duration</td>
<td>Job-wide</td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.duration</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Format</td>
<td>Scope</td>
<td>Consumable</td>
<td>Val / Opt</td>
<td>Value/Option Description</td>
<td>Default Value</td>
<td>Python Type</td>
<td>Platform</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>----------</td>
<td>------------</td>
<td>-----------</td>
<td>--------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>pmem</td>
<td>Amount of physical memory (workingset) for use by any single process of the job. Establishes a per-process resource limit. A scheduler rounds all resources of type size up to the nearest kb.</td>
<td>Size</td>
<td>Job-wide</td>
<td>No</td>
<td>No default</td>
<td>No default</td>
<td>pbs.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>preempt_targets</td>
<td>List of resources and/or queues. Jobs requesting those resources or in those queues are preemption targets.</td>
<td>string array</td>
<td>Job-wide</td>
<td>No</td>
<td>No default</td>
<td>No default</td>
<td>str</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pvmem</td>
<td>Amount of virtual memory for use by any single process in the job. Establishes a per-process resource limit. A scheduler rounds all resources of type size up to the nearest kb.</td>
<td>Size</td>
<td>Job-wide</td>
<td>No</td>
<td>No default</td>
<td>No default</td>
<td>pbs.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>site</td>
<td>Arbitrary string resource.</td>
<td>String</td>
<td>Job-wide</td>
<td>No</td>
<td>No default</td>
<td>No default</td>
<td>str</td>
<td></td>
<td></td>
</tr>
<tr>
<td>software</td>
<td>Site-specific software specification.</td>
<td>String</td>
<td>Job-wide</td>
<td>No</td>
<td>No default</td>
<td>No default</td>
<td>pbs.software</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## List of Built-in Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>soft_walltime</strong></td>
<td>Soft limit on walltime. Similar to <code>walltime</code>, but cannot be requested by unprivileged users, and job is not killed if it exceeds its <code>soft_walltime</code>. A job’s <code>soft_walltime</code> cannot exceed its <code>walltime</code>. Can be set by Manager only.</td>
<td><em>Duration</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>start_time</strong></td>
<td>The estimated start time for this job. Cannot be requested for a job; used for reporting only. Appears only in job’s <code>estimated</code> attribute. Read-only.</td>
<td><em>Integer</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td><strong>vmem</strong></td>
<td>Amount of virtual memory for use by all concurrent processes in the job. Establishes a per-chunk resource limit. A scheduler rounds all resources of type <code>size</code> up to the nearest kb.</td>
<td><em>Size</em></td>
<td>Host-level</td>
<td>Yes</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.size</td>
<td></td>
</tr>
<tr>
<td><strong>vnode</strong></td>
<td>Name of virtual node (vnode) on which to execute. Site-dependent. See “Vnode Attributes” on page 333 of the PBS Professional Reference Guide.</td>
<td><em>String</em></td>
<td>Host-level</td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td></td>
</tr>
<tr>
<td><strong>vntype</strong></td>
<td>The type of the vnode. Automatically set by PBS to one of two specific values for Cray vnodes. Has no meaning for non-Cray vnodes. On CLE, automatically added to <code>resources: line in sched_conf</code>. Must be manually added to `resources: line when your server/scheduler runs on non-CLE host.</td>
<td><em>string array</em></td>
<td>Host-level</td>
<td>No</td>
<td></td>
<td><code>cray_compute</code></td>
<td>Automatically set for Cray; none for non-Cray</td>
<td>str</td>
<td>all</td>
</tr>
</tbody>
</table>

--

**RG-288**

PBS Professional 18.2.3 Reference Guide
### List of Built-in Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Scope</th>
<th>Consumable</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>walltime</code></td>
<td>Amount of wall-clock time. Establishes a job-wide resource limit. Actual elapsed time may differ from <code>walltime</code> during Daylight Savings transitions.</td>
<td><code>Duration</code></td>
<td>Job-wide</td>
<td>No</td>
<td></td>
<td></td>
<td>5 years</td>
<td>pbs.duration</td>
<td></td>
</tr>
</tbody>
</table>
List of Built-in Resources
This chapter lists all of the supported PBS attributes. Attributes are listed by the PBS object they modify. For example, all supported attributes of jobs are listed in section 6.11, “Job Attributes”, on page 341. Attributes are case-sensitive.

6.1 Attribute Behavior

- When you set the value of most attributes, the change takes place immediately. You do not need to restart any daemons in order to make the change.
- When an attribute is unset, it behaves as if it is at its default value.

6.2 How To Set Attributes

Most attributes are set using the `qmgr` command. However, some vnode attributes must be set using the `pbs_mom -s insert` command, to create a Version 2 configuration file. For information about these requirements, see "Choosing Configuration Method" on page 42 in the PBS Professional Administrator’s Guide. The following are the instructions for setting all other attributes.

To set the value of a non-string_array attribute, use the `qmgr` command, either from the command line or within `qmgr`:

```bash
qmgr -c "set <object> <attribute> = <value>"
Qmgr: set <object> <attribute> = <value>
```

To set or change the value of a string_array attribute, use the `qmgr` command, either from the command line or within `qmgr`:

```bash
qmgr -c "set <object> <attribute> = <value>"
qmgr -c 'set <object> <attribute> = "<value,value>"'
qmgr -c 'set <object> <attribute> += <value>,'
qmgr -c 'set <object> <attribute> -= <value>,'
Qmgr: set <object> <attribute> = '<value,value>'
Qmgr: set <object> <attribute> += <value>
Qmgr: set <object> <attribute> -= <value>
```

PBS Professional 18.2.3 Reference Guide  RG-291
To unset the value of an attribute:

```
qmgr -c "unset <object> <attribute>"
Qmgr: unset <object> <attribute>
```

where `<object>` is one of `server, queue, hook, node, or sched`

For example, to set `resources_max.walltime` at the server to be 24 hours:

```
Qmgr: set server resources_max.walltime = 24:00:00
```

See “qmgr” on page 146.

### 6.3 Viewing Attribute Values

If you want to view attribute values, the following commands are helpful:

- `qstat`; see section 2.59, “qstat”, on page 192
- `qmgr`; see section 2.49, “qmgr”, on page 146
- `pbs_rstat`; see section 2.34, “pbs_rstat”, on page 95

- To see server attributes, use one of the following:
  ```
  qstat -B -f
  Qmgr: list server
  ```

- To see queue attributes, use one of the following:
  ```
  qstat -Q -f <queue name>
  Qmgr: list queue <queue name>
  ```

- To see job attributes:
  ```
  qstat -f <job ID>
  ```

- To see hook attributes:
  ```
  Qmgr: list hook <hook name>
  ```

- To see scheduler attributes:
  ```
  Qmgr: list sched
  ```

- To see vnode attributes:
  ```
  Qmgr: list node <node name>
  ```

- To see reservation attributes:
  ```
  pbs_rstat -F
  ```
6.4 Attribute Table Format

In the following tables, the columns contain the following information:

- **Name**: The name of the attribute
- **Description**: A description of the attribute’s function
- **Format**: The attribute’s format
- **Val/Opt**: If the attribute can take only specific values or options, each is listed here
- **Value/Option Description**: If the attribute can take only specific values or options, the behavior of each value or option is described here
- **Default Value, Def Val**: The attribute’s default value, if any
- **Python Type**: The attribute’s Python type
- **User, Oper, Mgr**: Indicates the actions allowed for unprivileged users, Operators, and Managers

The following table shows the operations allowed and their symbols:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>Entity can read attribute</td>
</tr>
<tr>
<td>w</td>
<td>Entity can directly set or alter attribute</td>
</tr>
<tr>
<td>s</td>
<td>Entity can set but not alter attribute</td>
</tr>
</tbody>
</table>
### Attributes

#### 6.5 Caveats

- The Python types listed as Python dictionaries support a restricted set of operations. They can reference values by index. Other features, such as `has_key()`, are not available.
- Do not use `qmgr` to set attributes for reservation queues.

#### Table 6-1: User, Operator, Manager Actions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;a&quot;</td>
<td>Entity can alter but not set attribute</td>
</tr>
<tr>
<td>&quot;i&quot;</td>
<td>Entity can indirectly set attribute</td>
</tr>
<tr>
<td>&quot;-&quot;</td>
<td>Entity cannot set or alter attribute, whether directly or indirectly</td>
</tr>
</tbody>
</table>
### 6.6 Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl_host_enable</td>
<td>Specifies whether the server obeys the host access control list in the acl_hosts server attribute.</td>
<td>Boolean</td>
<td></td>
<td>When this attribute is True, the server limits host access according to the access control list.</td>
<td>False; all hosts allowed access</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>acl_host_moms_enable</td>
<td>Specifies whether all MoMs are automatically allowed to contact the server with the same privilege as hosts listed in the acl_hosts server attribute.</td>
<td>Boolean</td>
<td>True</td>
<td>All MoMs are automatically allowed to contact the server with the same privilege as hosts listed in the acl_hosts server attribute.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>acl_host_moms_enable</td>
<td>Specifies whether all MoMs are automatically allowed to contact the server with the same privilege as hosts listed in the acl_hosts server attribute.</td>
<td>Boolean</td>
<td>False</td>
<td>MoMs are not automatically allowed to contact the server with the same privilege as hosts listed in the acl_hosts server attribute.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>acl_hosts</td>
<td>List of hosts from which services can be requested of this server. Requests from the server host are always honored whether or not that host is in the list. This list contains the fully qualified domain names of the hosts. List is evaluated left-to-right; first match in list is used.</td>
<td>String Syntax: &quot;[+</td>
<td>-]&lt;hostname&gt;.&lt;domain&gt;[, ...]&quot;</td>
<td></td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>acl_resv_group_enable</td>
<td>Specifies whether the server obeys the group reservation access control list in the acl_resv_groups server attribute.</td>
<td>Boolean</td>
<td></td>
<td>When this attribute is True, the server limits group access according to the access control list.</td>
<td>False; all groups allowed access</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>acl_resv_groups</td>
<td>List of groups allowed or denied permission to create reservations in this PBS complex. The groups in the list are groups on the server host, not submission hosts. List is evaluated left-to-right; first match in list is used.</td>
<td>String Syntax: &quot;[+</td>
<td>-]&lt;group name&gt;[, ...]&quot;</td>
<td></td>
<td></td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>acl_resv_host_enable</td>
<td>Specifies whether the server obeys the host reservation access control list in the acl_resv_hosts server attribute.</td>
<td>Boolean</td>
<td></td>
<td>When this attribute is True, the server limits host access according to the access control list.</td>
<td>False; access allowed from all hosts</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
</tbody>
</table>
## Attributes

### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mr</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl_resv_hosts</td>
<td>List of hosts from which reservations can be created in this PBS complex.</td>
<td>String</td>
<td>Val / Opt</td>
<td>When this attribute is True, the server limits user reservation creation according to the access control list.</td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td></td>
<td>This list is made up of the fully-qualified domain names of the hosts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>List is evaluated left-to-right; first match in list is used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>acl_resv_user_enable</td>
<td>Specifies whether the server limits which users are allowed to create reservations, according to the access control list in the acl_resv_users server attribute.</td>
<td>Boolean</td>
<td></td>
<td></td>
<td>False; all users are allowed to create reservations</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td>acl_resv_users</td>
<td>List of users allowed or denied permission to create reservations in this PBS complex. List is evaluated left-to-right; first match in list is used.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>acl_roots</td>
<td>List of users with root privilege who can submit and run jobs in this PBS complex. For any job whose owner is root or Administrator, the job owner must be listed in this access control list, or the job is rejected. List is evaluated left-to-right; first match in list is used.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default; no root jobs allowed</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>Can be set or altered by root only, and only at the server host.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>acl_user_enable</td>
<td>Specifies whether the server limits which users are allowed to run commands at the server, according to the control list in the acl_users server attribute.</td>
<td>Boolean</td>
<td></td>
<td></td>
<td>False; all users have access</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>acl_users</td>
<td>List of users allowed or denied permission to run commands at this server.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>List is evaluated left-to-right; first match in list is used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>backfill_depth</td>
<td>Specifies backfilling behavior. Sets the number of jobs that are to be backfilled around. Overridden by backfill_depth queue attribute. Recommendation: set this to less than 100.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>Unset; When unset, backfill depth is 1</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User Op</th>
<th>Other Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>Informational text. Can be set by a scheduler or other privileged client.</td>
<td><code>String of any form</code></td>
<td></td>
<td>No default</td>
<td>str</td>
<td></td>
<td>r</td>
<td>r, r, w</td>
</tr>
<tr>
<td>default_chunk</td>
<td>The list of resources which will be inserted into each chunk of a job's select specification if the corresponding resource is not specified by the user. This provides a means for a site to be sure a given resource is properly accounted for even if not specified by the user.</td>
<td><code>String</code>. Syntax: <code>default_chunk. &lt;resource name&gt; = &lt;value&gt;, default_chunk. &lt;resource name&gt; = &lt;value&gt;, ...</code></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource Syntax: <code>default_chunk[&quot;&lt;resource name&gt;&quot;] = &lt;value&gt;</code> where <code>resource name</code> is any built-in or custom resource</td>
<td>r</td>
<td>r, r, w</td>
<td></td>
</tr>
<tr>
<td>default_node</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>default_qdel_arguments</td>
<td>Argument to qdel command. Automatically added to all qdel commands. See <code>qdel(1B)</code>. Overrides standard defaults. Overridden by arguments given on the command line.</td>
<td><code>String</code>. Syntax: <code>&quot;-Wsuppr ess_mail=&lt;N&gt;&quot;</code></td>
<td></td>
<td>No default</td>
<td>pbs.args</td>
<td></td>
<td>r</td>
<td>r, r, w</td>
</tr>
<tr>
<td>default_qsub_arguments</td>
<td>Arguments that are automatically added to the qsub command. Any valid arguments to qsub command, such as job attributes. Setting a job attribute via <code>default_qsub_arguments</code> sets that attribute for each job which does not explicitly override it. See <code>qsub(1B)</code>. Settable by the administrator via the qmgr command. Overrides standard defaults. Overridden by arguments given on the command line and in script directives.</td>
<td><code>String</code>. Syntax: <code>&quot; &lt;option&gt; &lt;value&gt; &quot;, e.g. &quot;-r y -N MyJob&quot;</code></td>
<td></td>
<td>No default</td>
<td>pbs.args</td>
<td></td>
<td>r</td>
<td>r, r, w</td>
</tr>
<tr>
<td>default_queue</td>
<td>The name of the default target queue. Used for requests that do not specify a queue name. Must be set to an existing queue.</td>
<td><code>Queue name</code></td>
<td></td>
<td></td>
<td>workq</td>
<td>pbs.queue</td>
<td>r</td>
<td>r, r, w</td>
</tr>
</tbody>
</table>

---

**PBS Professional 18.2.3 Reference Guide**

**RG-297**
# Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User Op Or Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>eligible_time_enable</td>
<td>Controls starving behavior. Toggles between using the value of the job’s eligible_time attribute and the value of now() - etime to evaluate whether job is starving.</td>
<td>Boolean</td>
<td>True</td>
<td>The value of the job’s eligible_time attribute is used for its starving time.</td>
<td>False</td>
<td>bool</td>
<td>r r r r</td>
</tr>
<tr>
<td>False</td>
<td>The value of now() - etime is used for the job’s starving time.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>est_start_time_freq</td>
<td>Obsolete. No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flatuid</td>
<td>Used for authorization allowing users to submit and alter jobs. Specifies whether user names are treated as being the same across the PBS server and all submission hosts in the PBS complex. Can be used to allow users without accounts at the server host to submit jobs. If UserA has an account at the server host, PBS requires that UserA@&lt;server host&gt; is the same as UserA@&lt;execution host&gt;.</td>
<td>Boolean</td>
<td>True</td>
<td>PBS assumes that UserA@&lt;submithost&gt; is same user as UserA@&lt;server name&gt;. Jobs that run under the name of the job owner do not need authorization. A job submitted under a different username, by using the u option to the qsub command, requires authorization. Entries in .rhosts or hosts.equiv are not checked, so even if UserA@host1 has an entry for UserB@host2, UserB@host2 cannot operate on UserA@host1’s jobs. User without account on server can submit jobs.</td>
<td>False; authorization is required</td>
<td>bool</td>
<td>r r r r</td>
</tr>
<tr>
<td>False</td>
<td>PBS does not assume that UserA@&lt;submithost&gt; is the same user as UserA@&lt;server host&gt;. Jobs that run under the name of the job owner need authorization. Users must have accounts on the server host to submit jobs.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLicenses</td>
<td>The number of floating licenses currently available for allocation to unlicensed CPUs. One license is required for each virtual CPU.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r r r r</td>
</tr>
</tbody>
</table>
# Attributes

## Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Operator</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>job_history_duration</td>
<td>The length of time PBS will keep each job's history.</td>
<td>Duration</td>
<td></td>
<td>Two weeks</td>
<td>pbs.duration</td>
<td>r</td>
<td>r, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>job_history_enable</td>
<td>Enables job history management. Setting this attribute to True enables job history management.</td>
<td>Boolean</td>
<td></td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>job_requeue_timeout</td>
<td>The amount of time that can be taken while requeueing a job. Minimum allowed value: 1 second. Maximum allowed value: 3 hours.</td>
<td>Duration</td>
<td></td>
<td>45 seconds</td>
<td>pbs.duration</td>
<td>r</td>
<td>r, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>job_sort_formula</td>
<td>Formula for computing job priorities. Described in the PBS Professional Administrator's Guide. If the attribute job_sort_formula is set, all schedulers use the formula in it to compute job priorities. When this scheduler sorts jobs according to the formula, it computes a priority for each job, where that priority is the value produced by the formula. Jobs with a higher value get higher priority.</td>
<td>String</td>
<td></td>
<td>Unset</td>
<td>pbs.job_sort_formula</td>
<td>r</td>
<td>r, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jobscript_max_size</td>
<td>Limit on the size of any job script. Units default to bytes</td>
<td>size</td>
<td></td>
<td>100MB</td>
<td>pbs.size</td>
<td>r</td>
<td>r, w</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

#### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>license_count</code></td>
<td>The <code>license_count</code> attribute contains the following elements with their values: <code>Avail_Global</code>, <code>Avail_Local</code>, <code>Used</code>, <code>High_Use</code>, <code>Avail_Sockets</code>, <code>Avail_Nodes</code>, <code>Unused_Sockets</code>, <code>Unused_Nodes</code>. When this attribute is displayed, it shows only the elements that are relevant to the type of licensing being used. For example, if you are using node licenses, it shows only <code>Avail_Nodes</code> and <code>Unused_Nodes</code>.</td>
<td><code>String</code></td>
<td>Avail_Global</td>
<td>The number of PBS CPU licenses still kept by the Altair license server (checked in.)</td>
<td><code>Avail_Global:0</code></td>
<td><code>string</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Avail_Local</td>
<td>The number of PBS CPU licenses still kept by PBS (checked out.)</td>
<td><code>Avail_Local:0</code></td>
<td><code>int</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Used</td>
<td>The number of PBS CPU licenses currently in use.</td>
<td><code>Used:0</code></td>
<td><code>int</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>High_Use</td>
<td>The highest number of PBS CPU licenses ever checked out and used by the current instance of the PBS server.</td>
<td><code>High_Use:0</code></td>
<td><code>int</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Avail_Sockets</td>
<td>The total number of socket licenses in the socket license file.</td>
<td><code>Avail_Sockets:0</code></td>
<td><code>int</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Unused_Sockets</td>
<td>The number of unused socket licenses.</td>
<td><code>Unused_Sockets:0</code></td>
<td><code>int</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Avail_Nodes</td>
<td>The total number of node licenses in the node license file.</td>
<td><code>Avail_Nodes:0</code></td>
<td><code>int</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Unused_Nodes</td>
<td>The number of unused node licenses.</td>
<td><code>Unused_Nodes:0</code></td>
<td><code>int</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
</tr>
<tr>
<td><code>log_events</code></td>
<td>The types of events the server logs.</td>
<td><code>Integer</code></td>
<td></td>
<td>511</td>
<td>int</td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>w</code></td>
</tr>
<tr>
<td><code>mail_from</code></td>
<td>The username from which server-generated mail is sent to users. Mail is sent to this address upon failover. On Windows, requires fully qualified mail address.</td>
<td><code>String</code></td>
<td></td>
<td>adm</td>
<td>str</td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>r</code></td>
<td><code>w</code></td>
</tr>
</tbody>
</table>
## Attributes

### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>managers</td>
<td>List of PBS Managers.</td>
<td>String</td>
<td></td>
<td><code>&lt;user-name&gt;@&lt;host-name&gt;.&lt;subdomain&gt;.&lt;domain&gt;[,&lt;user name&gt;@&lt;host-name&gt;.&lt;subdomain&gt;.&lt;domain&gt;...]</code> The host, subdomain, or domain name may be wildcarded with an asterisk (*).</td>
<td>Root on the server host</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>max_array_size</td>
<td>The maximum number of subjobs allowed in any array job.</td>
<td>Integer</td>
<td></td>
<td>&gt;0</td>
<td>10000</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>max_concurrent_provision</td>
<td>The max_concurrent_provision attribute is the number of vnodes allowed to be in the process of being provisioned. Cannot be set to zero. When unset, default value is used.</td>
<td>Integer</td>
<td></td>
<td>&gt;0</td>
<td>5</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>max_group_res</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum amount of the specified resource that any single group may consume in this PBS complex.</td>
<td>String</td>
<td></td>
<td>Any PBS resource, e.g. “ncpus”, “mem”, “pmem”</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
</tbody>
</table>

### Additional Information

- **managers**
  - Description: List of PBS Managers.
  - Format: String
  - Value/Option Description: `<user-name>@<host-name>.<subdomain>.<domain>[,<user name>@<host-name>.<subdomain>.<domain>...]` The host, subdomain, or domain name may be wildcarded with an asterisk (*).
  - Default Value: Root on the server host
  - Python Type: pbs.acl
  - User: Read, Write
  - Oper: Read, Write
  - Mgr: Read, Write

- **max_array_size**
  - Description: The maximum number of subjobs allowed in any array job.
  - Format: Integer
  - Value/Option Description: >0
  - Default Value: 10000
  - Python Type: int
  - User: Read, Write
  - Oper: Read, Write
  - Mgr: Read, Write

- **max_concurrent_provision**
  - Description: The max_concurrent_provision attribute is the number of vnodes allowed to be in the process of being provisioned. Cannot be set to zero. When unset, default value is used.
  - Format: Integer
  - Value/Option Description: >0
  - Default Value: 5
  - Python Type: int
  - User: Read, Write
  - Oper: Read, Write
  - Mgr: Read, Write

- **max_group_res**
  - Description: Old limit attribute. Incompatible with new limit attributes. The maximum amount of the specified resource that any single group may consume in this PBS complex.
  - Format: String
  - Value/Option Description: Any PBS resource, e.g. “ncpus”, “mem”, “pmem”
  - Default Value: No default
  - Python Type: pbs.pbs_resource
  - User: Read, Write
  - Oper: Read, Write
  - Mgr: Read, Write
### Chapter 6

## Attributes

### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mov</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_group_res_soft</td>
<td>Old limit attribute. Incompatible with new limit attributes. The soft limit for the specified resource that any single group may consume in this complex. If a group is consuming more than this amount of the specified resource, their jobs are eligible to be preempted by jobs from groups who are not over their soft limit.</td>
<td>String. Syntax: max_group_res_soft.&lt;resource name&gt;=&lt;value&gt;</td>
<td>Any PBS resource, e.g. “ncpus”, “mem”, “pmem”, etc.</td>
<td>None</td>
<td>pbs.pbs_resource</td>
<td>Syntax: max_group_res_soft[&quot;&lt;resource name&gt;&quot;]=&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r, r, r, w, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_group_run</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs owned by the users in one group allowed to be running within this complex at one time.</td>
<td>Integer</td>
<td>No default</td>
<td>int</td>
<td></td>
<td>r, r, r, w, w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_group_run_soft</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs owned by the users in one group allowed to be running in this complex at one time. If a group has more than this number of jobs running, their jobs are eligible to be preempted by jobs from groups who are not over their soft limit.</td>
<td>Integer</td>
<td>No default</td>
<td>int</td>
<td></td>
<td>r, r, r, w, w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_queued</td>
<td>Limit attribute. The maximum number of jobs allowed to be queued or running in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td>Limit specification. See Chapter 7, &quot;Formats&quot;, on page 363.</td>
<td>No default</td>
<td>str</td>
<td></td>
<td>r, r, r, w, w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_queued_res</td>
<td>Limit attribute. The maximum amount of the specified resource allowed to be allocated to jobs queued or running in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td>Limit specification. See Chapter 7, &quot;Formats&quot;, on page 363. Syntax: max_queued_res.&lt;resource name&gt; = &lt;.limit&gt;</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>Syntax: max_queued_res[&quot;&lt;resource name&gt;&quot;]=&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r, r, r, w, w</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

#### max_run
- **Description**: Limit attribute. The maximum number of jobs allowed to be running in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.
- **Format**: Limit specification. See Chapter 7, "Formats", on page 363.
- **Default Value**: No default
- **Python Type**: str
- **User Op**: r, r, r, w

#### max_run_res
- **Description**: Limit attribute. The maximum amount of the specified resource allowed to be allocated to jobs running in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.
- **Format**: Limit specification. See Chapter 7, "Formats", on page 363.
- **Syntax**: `max_run_res.<resource name> = <limit>`
- **Default Value**: No default
- **Python Type**: `pbs.pbs_resource`
- **User Op**: r, r, r, w

#### max_run_res_soft
- **Description**: Limit attribute. Soft limit on the amount of the specified resource allowed to be allocated to jobs running in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.
- **Format**: Limit specification. See Chapter 7, "Formats", on page 363.
- **Syntax**: `max_run_res_soft.<resource name> = <limit>`
- **Default Value**: No default
- **Python Type**: `pbs.pbs_resource`
- **User Op**: r, r, r, w

#### max_run_soft
- **Description**: Limit attribute. Soft limit on the number of jobs allowed to be running in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.
- **Format**: Limit specification. See Chapter 7, "Formats", on page 363.
- **Default Value**: No default
- **Python Type**: str
- **User Op**: r, r, r, w

#### max_running
- **Description**: Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs in this complex allowed to be running at any given time.
- **Format**: Integer
- **Default Value**: No default
- **Python Type**: int
- **User Op**: r, r, r, w
### Attributes

**Server Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Operator</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_user_res</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum amount of the specified resource that any single user may consume within this complex.</td>
<td>String. Syntax: <code>max_user_res.&lt;resource name&gt;=&lt;value&gt;</code></td>
<td>Any PBS resource, e.g. “ncpus”, “mem”, “pmem”, etc.</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>Syntax: <code>max_user_res[&quot;&lt;resource name&gt;&quot;]=&lt;value&gt;</code> where <code>resource name</code> is any built-in or custom resource</td>
<td>r, r, r, r, w, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_user_res_soft</td>
<td>Old limit attribute. Incompatible with new limit attributes. The soft limit on the amount of the specified resource that any single user may consume within this complex. If a user is consuming more than this amount of the specified resource, their jobs are eligible to be preempted by jobs from users who are not over their soft limit.</td>
<td>String. Syntax: <code>max_user_res_soft.&lt;resource name&gt;=&lt;value&gt;</code></td>
<td>Any valid PBS resource, e.g. “ncpus”, “mem”, “pmem”, etc.</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>Syntax: <code>max_user_res_soft[&quot;&lt;resource name&gt;&quot;]=&lt;value&gt;</code> where <code>resource name</code> is any built-in or custom resource</td>
<td>r, w, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_user_run</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs owned by a single user allowed to be running within this complex at one time.</td>
<td>Integer</td>
<td>No default</td>
<td>int</td>
<td>r, r, r, w, w</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_user_run_soft</td>
<td>Old limit attribute. Incompatible with new limit attributes. The soft limit on the number of jobs owned by a single user that are allowed to be running within this complex at one time. If a user has more than this number of jobs running, their jobs are eligible to be preempted by jobs from users who are not over their soft limit.</td>
<td>Integer</td>
<td>No default</td>
<td>int</td>
<td>r, r, r, w, w</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### node_fail_requeue

Controls whether running jobs are automatically requeued or are deleted when the primary execution host fails. Number of seconds to wait after losing contact with Mother Superior before requeueing or deleting jobs. Reverts to default value when server is restarted.

<table>
<thead>
<tr>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer.</td>
<td>&lt;0</td>
<td>Behaves as if set to 1.</td>
<td>310</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>Units:</td>
<td>0</td>
<td>Jobs are not requeued; they are left in the Running state until the execution host is recovered.</td>
<td></td>
<td></td>
<td>r</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td>Seconds.</td>
<td>&gt;0</td>
<td>When the host has been down for the specified number of seconds, jobs are requeued if they are marked as rerunnable, or are deleted.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unset</td>
<td></td>
<td>Behaves as if set to default value of 310.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### node_group_enable

Specifies whether placement sets (which includes node grouping) are enabled. See node_group_key server attribute.

<table>
<thead>
<tr>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td></td>
<td>When set to True, placement sets are enabled.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
</tbody>
</table>

### node_group_key

Specifies the resources to use for placement sets (node grouping). Overridden by queue’s node_group_key attribute. See node_group_enable server attribute.

<table>
<thead>
<tr>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>String_array</td>
<td></td>
<td>When specifying multiple resources, separate them with commas and enclose the value in double quotes.</td>
<td>Unset</td>
<td>pbs.node_group_key</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
</tbody>
</table>

### operators

List of PBS Operators.

<table>
<thead>
<tr>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>String.</td>
<td></td>
<td>Syntax:</td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td>&lt;user-name&gt;@&lt;host-name&gt;.&lt;subdomain &gt;. &lt;domain name&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...f.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### pbs_license_file_location

**Deprecated.** Do not use.
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pbs_license_info</strong></td>
<td>Location of license information. Can be port number and hostname of license server, or local path-name to the actual license file associated with a license server.</td>
<td>String</td>
<td>Port number and hostname form: (&lt;port1&gt;@&lt;host1&gt;:&lt;port2&gt;@&lt;host2&gt;:...&lt;portN&gt;@&lt;hostN&gt;:&lt;path to license file&gt;) where host1, host2, ... hostN can be IP addresses, and the license file can be listed first or last.Delimiter between items is colon ((:)) for Linux and semicolon ((;)) for Windows.</td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td><strong>pbs_license_linger_time</strong></td>
<td>The number of seconds to keep an unused CPU license, when the number of licenses is above the value given by pbs_license_min.</td>
<td>Integer</td>
<td>31536000 seconds (1 year).</td>
<td>PBS</td>
<td>pbs.duration</td>
<td>r</td>
<td>r</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td><strong>pbs_license_max</strong></td>
<td>Maximum number of licenses to be checked out at any time, i.e maximum number of CPU licenses to keep in the PBS local license pool. Sets a cap on the number of CPUs that can be licensed at one time.</td>
<td>Integer</td>
<td>Maximum value for an integer</td>
<td>1</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td><strong>pbs_license_min</strong></td>
<td>Minimum number of CPUs to permanently keep licensed, i.e. the minimum number of CPU licenses to keep in the PBS local license pool. This is the minimum number of licenses to keep checked out. If set to zero or unset, PBS automatically sets the value to 1.</td>
<td>Integer</td>
<td>1</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>w</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>pbs_version</strong></td>
<td>The version of PBS for this server.</td>
<td>String</td>
<td>No default</td>
<td>PBS</td>
<td>pbs.version</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td><strong>power_provisioning</strong></td>
<td>Enables use of power profiles via PBS.</td>
<td>Boolean</td>
<td>True</td>
<td>Power provisioning is enabled.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>False</td>
<td>Power provisioning is disabled.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>python_restart_max_hooks</strong></td>
<td>The maximum number of hooks to be serviced before the Python interpreter is restarted. If this number is exceeded, and the time limit set in <strong>python_restart_min_interval</strong> has elapsed, the Python interpreter is restarted.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>100</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r,  w</td>
</tr>
<tr>
<td><strong>python_restart_max_objects</strong></td>
<td>The maximum number of objects to be created before the Python interpreter is restarted. If this number is exceeded, and the time limit set in <strong>python_restart_min_interval</strong> has elapsed, the Python interpreter is restarted.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>1000</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r,  w</td>
</tr>
<tr>
<td><strong>python_restart_min_interval</strong></td>
<td>The minimum time interval before the Python interpreter is restarted. If this interval has elapsed, and either the maximum number of hooks to be serviced (set in <strong>python_restart_max_hooks</strong>) has been exceeded or the maximum number of objects to be created (set in <strong>python_restart_max_objects</strong>) has been exceeded, the Python interpreter is restarted.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>30</td>
<td>pbs.duration</td>
<td>r</td>
<td>r</td>
<td>r,  w</td>
</tr>
<tr>
<td><strong>query_other_jobs</strong></td>
<td>Controls whether unprivileged users are allowed to select or query the status of jobs owned by other users.</td>
<td>Boolean</td>
<td></td>
<td>When this attribute is <strong>True</strong>, unprivileged users can query or select other users’ jobs.</td>
<td>On installation: <strong>True</strong> After being unset: <strong>False</strong></td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r,  w</td>
</tr>
<tr>
<td><strong>queued_jobs_threshold</strong></td>
<td>Limit attribute. The maximum number of jobs allowed to be queued in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td>Limit specification. See Chapter 7, &quot;Formats&quot;, on page 363.</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r,  w</td>
</tr>
</tbody>
</table>
## Attributes

### Chapter 6

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User Oper.</th>
</tr>
</thead>
<tbody>
<tr>
<td>queued_jobs_threshold_res</td>
<td>Limit attribute. The maximum amount of the specified resource allowed to be allocated to jobs queued in the complex. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r r r w</td>
</tr>
<tr>
<td>require_cred</td>
<td>Specifies the Kerberos credential authentication method required. All jobs submitted without the specified credential will be rejected. See require_cred_enable. Depends on optional Kerberos and DCE support. Not supported under Windows.</td>
<td>String</td>
<td></td>
<td></td>
<td>Unset</td>
<td>str</td>
<td>r r r w</td>
</tr>
<tr>
<td>require_cred_enable</td>
<td>Specifies whether the server is to use the Kerberos credential authentication method given in the require_cred server attribute. Depends on optional Kerberos and DCE support. Not supported under Windows.</td>
<td>Boolean</td>
<td></td>
<td></td>
<td>False; no Kerberos credential authentication used</td>
<td>bool</td>
<td>r r r w</td>
</tr>
<tr>
<td>reserve_retry_cutoff</td>
<td>The time period before the reservation start time during which PBS does not attempt to reconfirm a degraded reservation. When this value is changed, all degraded reservations use the new value. Must be greater than zero.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>7200 (2 hours)</td>
<td>int</td>
<td>- - r w</td>
</tr>
<tr>
<td>reserve_retry_init</td>
<td>The amount of time after a reservation becomes degraded that PBS waits before attempting to reconfirm the reservation. When this value is changed, only reservations that become degraded after the change use the new value. Must be greater than zero.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>7200 (2 hours)</td>
<td>int</td>
<td>- - r w</td>
</tr>
</tbody>
</table>

### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User Oper.</th>
</tr>
</thead>
<tbody>
<tr>
<td>queued_jobs_threshold_res</td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r r r w</td>
</tr>
<tr>
<td>require_cred</td>
<td></td>
<td></td>
<td></td>
<td>Unset</td>
<td>str</td>
<td>r r r w</td>
</tr>
<tr>
<td>require_cred_enable</td>
<td></td>
<td></td>
<td></td>
<td>False</td>
<td>bool</td>
<td>r r r w</td>
</tr>
<tr>
<td>reserve_retry_cutoff</td>
<td></td>
<td></td>
<td></td>
<td>7200</td>
<td>int</td>
<td>- - r w</td>
</tr>
<tr>
<td>reserve_retry_init</td>
<td></td>
<td></td>
<td></td>
<td>7200</td>
<td>int</td>
<td>- - r w</td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>resources_assigned</td>
<td>The total of each type of resource allocated to jobs running and exiting in this complex, plus the total of each type of resource allocated to any reservation. Reservation resources are added when the reservation starts.</td>
<td>String. Syntax: resources_assigned.&lt;resource name&gt;=&lt;value&gt;[], resources_assigned.&lt;resource name&gt;=&lt;value&gt;,...</td>
<td>r</td>
<td>No default</td>
<td>pbs.pbs_resources</td>
<td>Syntax: resources_assigned[&quot;&lt;resource name&gt; &quot;]=&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>resources_available</td>
<td>The list of available resources and their values defined on the server. Each resource is listed on a separate line.</td>
<td>String. Syntax: resources_available.&lt;resource name&gt;=&lt;value&gt;</td>
<td>r</td>
<td>No default</td>
<td>pbs.pbs_resources</td>
<td>Syntax: resources_available[&quot;&lt;resource name&gt; &quot;]=&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>w w</td>
</tr>
<tr>
<td>resources_cost</td>
<td>No longer used.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>resources_default</td>
<td>The list of default job-wide resource values that are set as limits for jobs in this complex when a) the job does not specify a limit, and b) there is no queue default. The value for a string array, e.g. resources_default.&lt;string array resource&gt;, can contain only one string. For host-level resources, see the default_chunk.&lt;resource name&gt; server attribute.</td>
<td>String. Syntax: resources_default.&lt;resource name&gt;=&lt;value&gt;[, ...]</td>
<td>r</td>
<td>No limit</td>
<td>pbs.pbs_resources</td>
<td>Syntax: resources_default[&quot;&lt;resource name&gt; &quot;]=&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>w w</td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Op</th>
<th>Mod</th>
</tr>
</thead>
<tbody>
<tr>
<td>resources_max</td>
<td>The maximum amount of each resource that can be requested by any single job in this complex, if there is not a <code>resources_max</code> value defined for the queue at which the job is targeted. This attribute functions as a gating value for jobs entering the PBS complex.</td>
<td>String</td>
<td>r, w, r</td>
<td>Syntax: <code>resources_max.&lt;resource name&gt;=&lt;value&gt;[, ...]</code></td>
<td>No limit</td>
<td>pbs.pbs_resource</td>
<td>r, w</td>
<td>r, w</td>
<td></td>
</tr>
<tr>
<td>restrict_res_to_release_on_suspend</td>
<td>Comma-separated list of consumable resources to be released when jobs are suspended. If unset, all consumable resources are released on suspension.</td>
<td>String_array</td>
<td>r, w, r</td>
<td>Syntax: <code>comma-separated list</code></td>
<td>unset</td>
<td>Python list</td>
<td>r, w</td>
<td>r, w</td>
<td>r, w</td>
</tr>
<tr>
<td>resv_enable</td>
<td>Specifies whether or not advance and standing reservations can be created in this complex.</td>
<td>Boolean</td>
<td>r, w, r</td>
<td>When set to <code>True</code>, new reservations can be created. When changed from <code>True</code> to <code>False</code>, new reservations cannot be created, but existing reservations are honored.</td>
<td><code>True</code></td>
<td>bool</td>
<td>r, w</td>
<td>r, w</td>
<td>r, w</td>
</tr>
<tr>
<td>resv_post_processing_time</td>
<td>The amount of time allowed for reservations to clean up after running jobs. Reservation duration and end time are extended by this amount of time. Jobs are not allowed to run during the cleanup period.</td>
<td>Duration</td>
<td>r, w, r</td>
<td>Unset; behaves as if zero</td>
<td>int</td>
<td>int</td>
<td>r, w</td>
<td>r, w</td>
<td>r, w</td>
</tr>
<tr>
<td>rpp_highwater</td>
<td>The maximum number of messages.</td>
<td>Integer</td>
<td>r, w, r</td>
<td>Greater than or equal to one</td>
<td>1024</td>
<td>int</td>
<td>r, w</td>
<td>r, w</td>
<td>r, w</td>
</tr>
<tr>
<td>rpp_max_pkt_check</td>
<td>Maximum number of TPP messages processed by the main server thread per iteration.</td>
<td>Integer</td>
<td>r, w, r</td>
<td>1024</td>
<td>int</td>
<td>int</td>
<td>r, w</td>
<td>r, w</td>
<td>r, w</td>
</tr>
<tr>
<td>rpp_retry</td>
<td>In a fault-tolerant setup (multiple <code>pbs_comm</code>), when the first <code>pbs_comm</code> fails partway through a message, this is number of times TPP tries to use the first <code>pbs_comm</code>.</td>
<td>Integer</td>
<td>r, w, r</td>
<td>Greater than or equal to zero</td>
<td>10</td>
<td>int</td>
<td>r, w</td>
<td>r, w</td>
<td>r, w</td>
</tr>
</tbody>
</table>
## Attributes

### Chapter 6

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mod</th>
</tr>
</thead>
</table>
| scheduler_iteration | The time between scheduling iterations.  
|                 |                                                                                   | Integer              |           |                          | 10 minutes (600 seconds)               | pbs.duration | r    | r,w  |     |
|                 |                                                                                   | Units: Seconds.      |           |                          |                                        |             | r    | r,w  |     |
| scheduling      | Enables scheduling of jobs. Specified by value of -a option to pbs_server command. If -a is not specified, value is taken from previous invocation of pbs_server. | Boolean              |           |                          | False if never set via pbs_server command. | bool         | r    | r,w  |     |
|                 |                                                                                   |                      |           |                          |                                        |             | r    | r,w  |     |
| server_host     | The name of the host on which the active server is running.  
|                 | If the secondary server takes over, this attribute is set to the name of the secondary server's host. When the primary server takes control again, this attribute shows the name of the primary server's host. | String. Syntax:  
|                 | <host-name>.<domain name>  
|                 | If the server is listening to a non-standard port, the port number is appended, with a colon, to the host-name: <host-name>.<domain name>:<port number> |                      |           |                          | No default                           | str          | r    | r,r  |     |

**Attributes Table**
## Attributes

### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User Operat</th>
<th>Mgr Operat</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_state</td>
<td>The current state of the server:</td>
<td><strong>String</strong></td>
<td><strong>Active</strong></td>
<td>The server is running. The scheduler is not in a scheduling cycle.</td>
<td>No default</td>
<td>Server state constant <code>pbs.SV_STATE_ACTIVE</code></td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Hot_Start</strong></td>
<td>The server will run first any jobs that were running when it was shut down.</td>
<td></td>
<td>Server state constant <code>pbs.SV_STATE_HOT</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Idle</strong></td>
<td>The server is running. Scheduling has been turned off.</td>
<td></td>
<td>Server state constant <code>pbs.SV_STATE_IDLE</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Scheduling</strong></td>
<td>The server is running. The scheduler is in a scheduling cycle.</td>
<td></td>
<td>Server state constant <code>pbs.SV_STATE_ACTIVE</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Terminating</strong></td>
<td>The server is terminating. No additional jobs will be run.</td>
<td></td>
<td>Server state constant <code>pbs.SV_STATE_ACTIVE</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Terminating_Delayed</strong></td>
<td>Server is terminating in delayed mode. No new jobs will be run. Server will shut down after all running jobs are finished.</td>
<td></td>
<td>Server state constant <code>pbs.SV_STATE_SHUTDEL</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>single_signon_password_enable</td>
<td>Only used on systems requiring passwords, such as Windows. Incompatible with other systems. Specifies whether or not users must give a password for each job. Can be enabled only when no jobs exist, or when all jobs have a bad password hold (“p” hold). Can be disabled only when no jobs exist.</td>
<td><strong>Boolean.</strong></td>
<td><strong>True</strong></td>
<td>Users submitting jobs must specify a password only once; PBS remembers it for future job execution.</td>
<td>Linux: <code>False</code> Windows: <code>True</code></td>
<td>bool</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>state_count</td>
<td>List of the number of jobs in each state in the complex. Suspended jobs are counted as running.</td>
<td><strong>String. Syntax:</strong> <code>transiting=&lt;value&gt;, queued=&lt;value&gt;, ...</code></td>
<td></td>
<td></td>
<td>No default</td>
<td><code>pbs.state_count</code></td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
## Attributes

### Server Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>system_cost</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>total_jobs</td>
<td>The total number of jobs in the complex. If the job_history_enable attribute is set to True, this includes jobs that are finished, deleted, and moved.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
# 6.7 Scheduler Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>For certain scheduler errors, PBS sets the scheduler’s comment attribute to specific error messages. You can use the comment attribute to notify another administrator of something, but PBS does overwrite the value of comment under certain circumstances.</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>do_not_span_psets</td>
<td>Specifies whether or not this scheduler requires the job to fit within one existing placement set.</td>
<td>Boolean</td>
<td>True</td>
<td>The job must fit in one existing placement set. All existing placement sets are checked. If the job fits in an occupied placement set, the job waits for the placement set to be available. If the job can’t fit within a single placement set, it won’t run.</td>
<td>False</td>
<td></td>
<td>r, r, r, w</td>
<td>r, w</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>False</td>
<td></td>
<td>This scheduler first attempts to place the job in a single placement set. All existing placement sets are checked. If the job fits in an occupied placement set, the job waits for the placement set to be available. If there is no existing placement set, occupied or empty, into which the job could fit, the job runs regardless of placement sets, running on whichever vnodes can satisfy the job’s resource request.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>job_sort_formula_threshold</td>
<td>Lower bound for calculated priority for job. If job priority is at or below this value, the job is not eligible to run in the current scheduler cycle.</td>
<td>Float</td>
<td></td>
<td></td>
<td>No default</td>
<td>float</td>
<td>r</td>
<td>r, r, w</td>
<td></td>
</tr>
<tr>
<td>only_explicit_psets</td>
<td>Specifies whether placement sets are created for unset resources.</td>
<td>Boolean</td>
<td>True</td>
<td>Placement sets are not created from vnodes whose value for a resource is unset.</td>
<td>False</td>
<td></td>
<td>r, r, r, w</td>
<td>r, w</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>False</td>
<td></td>
<td>Placement sets are created from vnodes whose value for a resource is unset.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>opt_backfill_fuzzy</td>
<td>Sets the trade-off between scheduling cycle speed and granularity of estimated start time calculation.</td>
<td>String</td>
<td>off</td>
<td>Finest granularity, no speedup</td>
<td>unset, behaves like low</td>
<td></td>
<td>r</td>
<td>r, r, w</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>low</td>
<td></td>
<td>Fairly fine granularity, some speedup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>medium</td>
<td></td>
<td>Medium granularity, medium speedup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>high</td>
<td></td>
<td>Coarse granularity, greatest speedup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Scheduler Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>partition</td>
<td>Comma-separated list of named partitions for which this scheduler is to run jobs. Cannot be set on default scheduler.</td>
<td>String_array</td>
<td></td>
<td>&quot;None&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pbs_version</td>
<td>The version of PBS for this scheduler.</td>
<td>String</td>
<td></td>
<td>No default</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scheduler_iteration</td>
<td>Time in seconds between scheduling iterations. If you set the server’s scheduler_iteration attribute, that value is assigned to the default scheduler’s scheduler_iteration attribute, and vice versa.</td>
<td>Integer</td>
<td></td>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scheduling</td>
<td>Enables scheduling of jobs. If you set the server’s scheduling attribute, that value is assigned to the default scheduler’s scheduling attribute, and vice versa.</td>
<td>Boolean</td>
<td></td>
<td>For default scheduler: True</td>
<td>For multi-scheds: False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sched_cycle_length</td>
<td>This scheduler’s maximum cycle length. Overwritten by the -a alarm option to pbs_sched command.</td>
<td>Duration</td>
<td></td>
<td>20:00 (20 minutes)</td>
<td>None</td>
<td>r r r r</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sched_host</td>
<td>The hostname of the machine on which this scheduler runs. Cannot be set on default scheduler; value for default scheduler is server hostname. Must be set by administrator.</td>
<td>String</td>
<td></td>
<td>Server's host</td>
<td>None</td>
<td>- r r r</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sched_log</td>
<td>Directory where this scheduler writes its logs. Permissions should be 755. Must be owned by root. Cannot be shared with another scheduler. Use default value; do not set.</td>
<td>String</td>
<td></td>
<td>$PBS_HOME/sched_logs_&lt;scheduler_name&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sched_port</td>
<td>Port on which this scheduler listens. Cannot be set on default scheduler. Must be set by administrator.</td>
<td>String</td>
<td></td>
<td>No default</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sched_preempt_enforce_resumption</td>
<td>Controls whether this scheduler treats preempted jobs as top jobs. When True, these are top jobs.</td>
<td>Boolean</td>
<td></td>
<td>False</td>
<td></td>
<td>r r r r</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

#### Scheduler Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>sched_priv</td>
<td>Directory where this scheduler keeps fairshare usage, resource_group, holidays, and sched_config files. Must be owned by root. Use default value; do not set.</td>
<td>String</td>
<td></td>
<td>$PBS_HOME/sched_priv_&lt;scheduler name&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>State of this scheduler.</td>
<td>String</td>
<td></td>
<td>down</td>
<td>Scheduler is not running</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>throughput_mode</td>
<td>Allows scheduler to run faster; it doesn’t have to wait for each job to be accepted, and doesn’t wait for execjob_begin hooks to finish. Also allows jobs that were changed via qalter, server_dyn_res scripts, or peering to run in the same scheduling cycle where they were changed.</td>
<td>Boolean</td>
<td></td>
<td>True</td>
<td>Scheduler runs asynchronously and faster. Only available when PBS complex is in TPP mode.</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 6.8 Reservation Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>User</th>
<th>Op</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account_Name</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
| Authorized_Groups    | List of groups who can or cannot submit jobs to this reservation. Group names are interpreted relative to the server, not the submission host. List is evaluated left-to-right; first match in list is used. This list is used to set the reservation queue’s acl_groups attribute. See the G option to the pbs_rsub command. | String. Syntax: 

```
[+|-]<group name> [, 

[+|-]<group name> ...]            
```

where ‘-’ means “deny” and ‘+’ means “allow”. | No default. (Jobs can be submitted by all groups) | pbs.acl | u | r, w, r |
| Authorized_Hosts     | The list of hosts from which jobs can and cannot be submitted to this reservation. List is evaluated left-to-right; first match in list is used. This list is used to set the reservation queue’s acl_hosts attribute. See the H option to the pbs_rsub command. | String. Syntax: 

```
[+|-]<hostname> [, [+|-]<hostname> ...]           
```

where ‘-’ means “deny” and ‘+’ means “allow”. Hostnames may be wildcarded using an asterisk, according to the following rules: 

- A hostname can contain at most one asterisk.
- The asterisk must be the leftmost label.

Examples: 

```
*.test.example.com  
*.example.com  
*.com
``` | No default. (Jobs can be submitted from all hosts) | pbs.acl | u | r, w, r |
### Attributes

#### Reservation Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>User Op</th>
<th>Owner Op</th>
<th>Oper Op</th>
<th>Mgr Op</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authorized_Users</strong></td>
<td>The list of users who can or cannot submit jobs to this reservation. List is evaluated left-to-right; first match in list is used. This list is used to set the reservation queue’s acl_users attribute. See the U option to the pbs_rsub command.</td>
<td><strong>String. Syntax:</strong> `[+</td>
<td>-]&lt;username&gt;[:&lt;hostname&gt;].&lt;domain&gt; [+,</td>
<td></td>
<td>Reservation owner only</td>
<td>pbs.acl</td>
<td></td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td><strong>ctime</strong></td>
<td>Timestamp; time at which the reservation was created.</td>
<td><strong>Timestamp.</strong></td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td><strong>group_list</strong></td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>hostname</strong></td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Reservation Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>interactive</td>
<td>Number of seconds that the pbs_rsub command will block while waiting for confirmation or denial of the reservation. See the -I block_time option to the pbs_rsub command.</td>
<td>Integer</td>
<td>Less than zero</td>
<td>The reservation is automatically deleted if it cannot be confirmed in the time specified.</td>
<td>Zero</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Zero or greater than zero</td>
<td>The reservation is not automatically deleted if it cannot be confirmed in the time specified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mail_Points</td>
<td>Sets the list of events for which mail is sent by the server. Mail is sent to the list of users specified in the Mail_Users attribute. See the mail_points option to the pbs_rsub command.</td>
<td>String</td>
<td>consisting of 1) one or more of the letters “a”, “b”, “c”, “e”, or 2) the string “n”. Cannot use “n” with any other letter</td>
<td>“ace”</td>
<td>pbs.mail_points</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>a</td>
<td>Notify when reservation is terminated</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b</td>
<td>Notify when reservation period begins</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>c</td>
<td>Notify when reservation is confirmed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>e</td>
<td>Notify when reservation period ends</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>n</td>
<td>Do not send mail. Cannot be used with other letters.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mail_Users</td>
<td>The set of users to whom mail is sent for the reservation events specified in the Mail_Points attribute. See the mail_list option to the pbs_rsub command.</td>
<td>String</td>
<td>Syntax: <code>&lt;username&gt;@&lt;hostname&gt;[,&lt;username&gt;@&lt;hostname&gt;, ...]</code></td>
<td>Reservation owner only</td>
<td></td>
<td>pbs.user_list</td>
</tr>
<tr>
<td>mtime</td>
<td>Timestamp: the time that the reservation was last modified.</td>
<td>Timestamp</td>
<td>Printed by qstat in human-readable Date format. Output in hooks as seconds since epoch.</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>User Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>queue</td>
<td>Name of the reservation queue. Jobs that are to use resources belonging to this reservation are submitted to this queue.</td>
<td><em>String.</em> Format for an advance reservation: (R&lt;unique\ integer&gt;) Format for a standing reservation: (S&lt;unique\ integer&gt;)</td>
<td></td>
<td></td>
<td>pbs.queue</td>
<td></td>
<td>r</td>
</tr>
<tr>
<td>reserve_count</td>
<td>The count of occurrences in a standing reservation.</td>
<td><em>Integer</em></td>
<td></td>
<td></td>
<td>int</td>
<td></td>
<td>r, w, r</td>
</tr>
<tr>
<td>reserve_duration</td>
<td>Reservation duration in seconds. For a standing reservation, this is the duration for one occurrence.</td>
<td><em>Integer</em></td>
<td></td>
<td></td>
<td>pbs.duration</td>
<td></td>
<td>r, w, r</td>
</tr>
<tr>
<td>reserve_end</td>
<td>The date and time when an advance reservation or the soonest occurrence of a standing reservation ends.</td>
<td><em>Timestamp.</em> Printed by <code>qstat</code> in human-readable <em>Date</em> format. Output in hooks as seconds since epoch.</td>
<td></td>
<td></td>
<td>int</td>
<td></td>
<td>r, w, r</td>
</tr>
<tr>
<td>reserve_ID</td>
<td>The reservation identifier.</td>
<td><em>String.</em> For an advance reservation: string of the form (R&lt;unique\ integer&gt;.&lt;server\ name&gt;) For a standing reservation: string of the form (S&lt;unique\ integer&gt;.&lt;server\ name&gt;)</td>
<td></td>
<td></td>
<td>str</td>
<td></td>
<td>r</td>
</tr>
<tr>
<td>reserve_index</td>
<td>The index of the soonest occurrence of a standing reservation.</td>
<td><em>Integer</em></td>
<td></td>
<td></td>
<td>int</td>
<td></td>
<td>r</td>
</tr>
<tr>
<td>Reserve_Name</td>
<td>The name assigned to the reservation during creation, if specified. See the N option to the <code>pbs_rsub</code> command.</td>
<td><em>String.</em> Syntax: up to 236 characters. First character is alphabetic</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r, w, r</td>
</tr>
</tbody>
</table>

---

**User Op:**
- `r` - Read
- `w` - Write
- `u` - Update
- `m` - Manage

**Val / Opt:**
- `V` - Value
- `O` - Option
## Attributes

### Reservation Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserve_Owner</td>
<td>The login name on the submission host of the user who created the reservation.</td>
<td>String. Syntax: <code>&lt;username&gt;@&lt;hostname&gt;</code></td>
<td>Val / Opt</td>
<td></td>
<td>str</td>
<td></td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>reserve_retry</td>
<td>If this reservation becomes degraded, this is the next time that PBS will attempt to reconfirm this reservation.</td>
<td>Timestamp. Printed by <code>qstat</code> in human-readable Date format. Output in hooks as seconds since epoch.</td>
<td>No default</td>
<td></td>
<td>int</td>
<td></td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>reserve_rule</td>
<td>The rule that describes the recurrence pattern of a standing reservation. See the <code>r</code> option to the <code>pbs_rsub</code> command.</td>
<td>String. Syntax: either of two forms: &quot;FREQ= &lt;freq_spec&gt;;&lt;COUNT= &lt;count_spec&gt;; &lt;interval_spec&gt;&quot; or &quot;FREQ= &lt;freq_spec&gt;;&lt;UNTIL= &lt;until_spec&gt;; &lt;interval_spec&gt;&quot;</td>
<td>freq_spec</td>
<td>Frequency with which the standing reservation repeats. Valid values are: WEEKLY</td>
<td>DAILY</td>
<td>HOURLY</td>
<td>No default</td>
<td>str</td>
<td></td>
</tr>
<tr>
<td>count_spec</td>
<td>The exact number of occurrences. Number up to 4 digits in length. Format: integer.</td>
<td>count_spec</td>
<td>No default</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interval_spec</td>
<td>Specifies interval. Format is one or both of: BYDAY = MO</td>
<td>TU</td>
<td>WE</td>
<td>TH</td>
<td>FR</td>
<td>SA</td>
<td>SU or BYHOUR = 0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>until_spec</td>
<td>Occurrences will start up to but not after date and time specified. Format: YYYY-MM-DD[THHMSS]</td>
<td>until_spec</td>
<td>No default</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>User</th>
<th>Op</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>reserve_start</td>
<td>The date and time when the reservation period for the reservation begins.</td>
<td>Timestamp.</td>
<td></td>
<td></td>
<td>No default</td>
<td></td>
<td>r, r, r</td>
<td>w, w, w</td>
<td></td>
</tr>
<tr>
<td>reserve_state</td>
<td>The state of the reservation.</td>
<td>String</td>
<td></td>
<td></td>
<td>NO RESV_NONE</td>
<td>Reservation state constant: pbs.RESV_STATE_NONE</td>
<td>r, r, r</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UN RESV_UNCONFIRMED</td>
<td>Reservation state constant: pbs.RESV_STATE_UNCONFIRMED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CO RESV_CONFIRMED</td>
<td>Reservation state constant: pbs.RESV_STATE_CONFIRMED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WT RESV_WAIT</td>
<td>Reservation state constant: pbs.RESV_STATE_WAIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TR RESV_TIME_TO_RUN</td>
<td>Reservation state constant: pbs.RESV_STATE_TIME_TO_RUN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RN RESV_RUNNING</td>
<td>Reservation state constant: pbs.RESV_STATE_RUNNING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FN RESV_FINISHED</td>
<td>Reservation state constant: pbs.RESV_STATE_FINISHED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BD RESV_BEING_DELETED</td>
<td>Reservation state constant: pbs.RESV_STATE_BEING_DELETED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DE RESV_DELETED</td>
<td>Reservation state constant: pbs.RESV_STATE_DELETED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DJ RESV_DELETING_JOBS</td>
<td>Reservation state constant: pbs.RESV_STATE_DELETING_JOBS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DG DEGRADED</td>
<td>Reservation state constant: pbs.RESV_STATE_DEGRADED</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Attributes

## Reservation Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>User</th>
<th>Other</th>
<th>Managed</th>
</tr>
</thead>
<tbody>
<tr>
<td>reserve_substate</td>
<td>The substate of the reservation or occurrence. The substate is used internally by PBS.</td>
<td>Integer</td>
<td>No</td>
<td>default</td>
<td>int</td>
<td></td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>reserve_type</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Resource_List</td>
<td>The list of resources allocated to the reservation. Jobs running in the reservation cannot use in aggregate more than the specified amount of a resource.</td>
<td>String</td>
<td>No</td>
<td>default</td>
<td></td>
<td>pbs.pbs_resource</td>
<td>r, r</td>
<td>r</td>
<td>w, w</td>
</tr>
<tr>
<td>resv_nodes</td>
<td>The list of each vnode and the resources allocated from it to satisfy the chunks requested for this reservation or occurrence.</td>
<td>String</td>
<td>No</td>
<td>default</td>
<td></td>
<td>pbs.exec_vnode</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>server</td>
<td>Name of server.</td>
<td>String</td>
<td>No</td>
<td>default</td>
<td></td>
<td>pbs.server</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>User_List</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Variable_List</td>
<td>Not used</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
# 6.9 Queue Attributes

In the following table, Queue Type indicates the type of queue to which the attribute applies: R (routing), E (execution):

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl_group_enable</td>
<td>Controls whether group access to the queue obeys the access control list defined in the acl_groups queue attribute.</td>
<td>Boolean</td>
<td>R, E</td>
<td></td>
<td>When set to True, group access to the queue is limited according to the group access control list.</td>
<td>False; all groups allowed access</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td>acl_groups</td>
<td>List of groups which are allowed or denied access to this queue. The groups in the list are groups on the server host, not submitting hosts. List is evaluated left-to-right; first match in list is used.</td>
<td>String</td>
<td>R, E</td>
<td></td>
<td>When set to True, host access to the queue is limited according to the host access control list.</td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td>acl_host_enable</td>
<td>Controls whether host access to the queue obeys the access control list defined in the acl_hosts queue attribute.</td>
<td>Boolean</td>
<td>R, E</td>
<td></td>
<td>When set to True, user access to the queue is limited according to the user access control list.</td>
<td>False; all users allowed access</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>acl_hosts</td>
<td>List of hosts from which jobs may be submitted to this queue. List is evaluated left-to-right; first match in list is used.</td>
<td>String</td>
<td>R, E</td>
<td></td>
<td>When set to True, host access to the queue is limited according to the host access control list.</td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>acl_user_enable</td>
<td>Controls whether user access to the queue obeys the access control list defined in the acl_users queue attribute.</td>
<td>Boolean</td>
<td>R, E</td>
<td></td>
<td>When set to True, user access to the queue is limited according to the user access control list.</td>
<td>False; all users allowed access</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>acl_users</td>
<td>List of users allowed or denied access to this queue. List is evaluated left-to-right; first match in list is used.</td>
<td>String</td>
<td>R, E</td>
<td></td>
<td>When set to True, user access to the queue is limited according to the user access control list.</td>
<td>No default</td>
<td>pbs.acl</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>alt_router</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>backfill_depth</td>
<td>Specifies backfilling behavior for this queue. Sets the number of jobs that are to be backfilled around in this queue. Overrides backfill_depth server attribute. Recommendation: set this to less than 100.</td>
<td>Integer</td>
<td>E</td>
<td>&gt;=0</td>
<td>PBS backfills around the specified number of jobs.</td>
<td>Unset</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkpoint_min</td>
<td>Minimum number of minutes of CPU time or walltime allowed between checkpoints of a job. If a user specifies a time less than this value, this value is used instead. The value given in checkpoint_min is used for both CPU minutes and walltime minutes.</td>
<td>Integer</td>
<td>E</td>
<td>No default</td>
<td>pbs.duration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| default_chunk   | The list of resources which will be inserted into each chunk of a job’s select specification if the corresponding resource is not specified by the user. This provides a means for a site to be sure a given resource is properly accounted for even if not specified by the user. | String       | E          | No default      | pbs.pbs_resource Syntax: ```default_chunk.<resource name> = <value>[
default_chunk.<resource name> = <value>, ...
``` |               |                                 |      |      |     |
| enabled         | Specifies whether this queue accepts new jobs.                              | Boolean      | R, E       | True            | This queue is enabled. This queue accepts new jobs; new jobs can be enqueued. | False         | bool                           |      |      |     |
|                 |                                                                             |              |            | False           | This queue does not accept new jobs. |               |                                 |      |      |     |
| from_route_only | Specifies whether this queue accepts jobs only from routing queues, or from both execution and routing queues. | Boolean      | R, E       | True            | This queue accepts jobs only from routing queues. | False         | bool                           |      |      |     |
|                 |                                                                             |              |            | False           | This queue accepts jobs from both execution and routing queues as well as directly from submitter. |               |                                 |      |      |     |
| hasnodes        | Indicates whether vnodes are associated with this queue. Set by PBS.        | Boolean      | E          | False           | This attribute is set to True if there are vnodes associated with this queue. | False; no vnodes are associated with this queue | bool                           |      |      |     |
### Attributes

#### Queue Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Op</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>kill_delay</td>
<td>The time delay between sending SIGTERM and SIGKILL when a <code>qdel</code> command is issued against a running job.</td>
<td>Integer.</td>
<td>E</td>
<td></td>
<td></td>
<td>10 seconds</td>
<td>pbs.duration</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>max_array_size</td>
<td>The maximum number of subjobs that are allowed in an array job.</td>
<td>Integer</td>
<td>R, E</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>max_group_res</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum amount of the specified resource that any single group may consume in a complex.</td>
<td>String</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>max_group_res_soft</td>
<td>Old limit attribute. Incompatible with new limit attributes. The soft limit on the amount of the specified resource that any single group may consume in a complex. If a group is consuming more than this amount of the specified resource, their jobs are eligible to be preempted by jobs from groups who are not over their soft limit.</td>
<td>String</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>max_group_run</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs owned by users in a single group that are allowed to be running from this queue at one time.</td>
<td>Integer</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>max_group_run_soft</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs owned by users in a single group that are allowed to be running from this queue at one time. If a group has more than this number of jobs running, their jobs are eligible to be preempted by jobs from groups who are not over their soft limit.</td>
<td>Integer</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
### Attributes

#### Queue Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_queuable</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs allowed to reside in this queue at any given time.</td>
<td><code>Integer</code></td>
<td>R, E</td>
<td></td>
<td>No default (no limit)</td>
<td><code>int</code></td>
<td>r</td>
<td>r, w, r, w</td>
<td></td>
</tr>
<tr>
<td>max_queued</td>
<td>Limit attribute. The maximum number of jobs allowed to be queued in or running from this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td><code>Limit specification. See Chapter 7, &quot;Formats&quot;, on page 363</code></td>
<td>R, E</td>
<td></td>
<td>No default</td>
<td><code>pbs.pbs_resource</code></td>
<td>r</td>
<td>r, r, w, w</td>
<td></td>
</tr>
</tbody>
</table>
| max_queued_res   | Limit attribute. The maximum amount of the specified resource allowed to be allocated to jobs queued in or running from this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes. | `Limit specification. See Chapter 7, "Formats", on page 363
Syntax: max_queued_res.<resource name> = <value>` | R, E       |                              | No default                                | `pbs.pbs_resource` | r    | r, r, w, w |
| max_run          | Limit attribute. The maximum number of jobs allowed to be running from this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes. | `Format: Limit specification. See Chapter 7, "Formats", on page 363` | E          |                              | No default                                | `pbs.pbs_resource` | r    | r, r, w, w |
| max_run_res      | Limit attribute. The maximum amount of the specified resource allowed to be allocated to jobs running from this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes. | `Format: Limit specification. See Chapter 7, "Formats", on page 363
Syntax: max_run_res.<resource name> = <value>` | E          |                              | No default                                | `pbs.pbs_resource` | r    | r, r, w, w |

**Chapter 6**
### Attributes

#### Queue Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User Op Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_run_res_soft</td>
<td>Limit attribute. Soft limit on the amount of the specified resource allowed to be allocated to jobs running from this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td>Format: Limit specification. See Chapter 7, &quot;Formats&quot;, on page 363. Syntax: [max_run_res_soft.&lt;resource name&gt;=&lt;value&gt;]</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r r r w w</td>
</tr>
<tr>
<td>max_run_soft</td>
<td>Limit attribute. Soft limit on the number of jobs allowed to be running from this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td>Limit specification. See Chapter 7, &quot;Formats&quot;, on page 363. Syntax: [max_run_soft.&lt;resource name&gt;=&lt;value&gt;]</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r r r w w</td>
</tr>
<tr>
<td>max_running</td>
<td>Old limit attribute. Incompatible with new limit attributes. For an execution queue, this is the largest number of jobs allowed to be running at any given time. For a routing queue, this is the largest number of jobs allowed to be transiting from this queue at any given time.</td>
<td>Integer</td>
<td>R, E</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r r r w w</td>
</tr>
<tr>
<td>max_user_res</td>
<td>Old limit attribute. Incompatible with new limit attributes. The maximum amount of the specified resource that any single user may consume.</td>
<td>String. Syntax: [max_user_res.&lt;resource name&gt;=&lt;value&gt;] Example: set queue workq max_user_res.ncpus=6</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r r r w w</td>
</tr>
<tr>
<td>max_user_res_soft</td>
<td>Old limit attribute. Incompatible with new limit attributes. The soft limit on the amount of the specified resource that any single user may consume. If a user is consuming more than this amount of the specified resource, their jobs are eligible to be preempted by jobs from users who are not over their soft limit.</td>
<td>String. Syntax: [max_user_res_soft.&lt;resource name&gt;=&lt;value&gt;] Example: set queue workq max_user_res_soft.ncpus=3</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r r r w w</td>
</tr>
</tbody>
</table>
## Attributes

### Name |
**max_user_run** |
Old limit attribute. Incompatible with new limit attributes. The maximum number of jobs owned by a single user that are allowed to be running from this queue at one time. |
Integer |
E |
No default |
int |
User Op Mgr |
\- r r r r w w |

### Name |
**max_user_run_soft** |
Old limit attribute. Incompatible with new limit attributes. The soft limit on the number of jobs owned by any single user that are allowed to be running from this queue at one time. If a user has more than this number of jobs running, their jobs are eligible to be preempted by jobs from users who are not over their soft limit. |
Integer |
E |
No default |
int |
User Op Mgr |
\- r r r r w w |

### Name |
**node_group_key** |
Specifies the resources to use for placement sets (node grouping). Overrides server’s node_group_key attribute. Specified resources must be of type string_array. |
String_array, Syntax: Comma-separated list of resource names. When specifying multiple resources, enclose value in double quotes. |
R E |
No default |
pbs.node_group_key |
User Op Mgr |
\- r r r r w w |

### Name |
**partition** |
Name of partition to which this queue is assigned. Cannot be set for routing queue. An execution queue cannot be changed to a routing queue while this attribute is set. |
String |
E |
No default |
str |
User Op Mgr |
\- r r r r w w |

### Name |
**Priority** |
The priority of this queue compared to other queues of the same type in this PBS complex. Priority can define a queue as an express queue. See preempt_queue_prio in Chapter 4, "Scheduler Parameters", on page 265. Used for execution queues only; the value of Priority has no meaning for routing queues. |
Integer |
E |
Valid values: -1024 to 1023 |
No default |
int |
User Op Mgr |
\- r r r r w w |

### Name |
**queued_jobs_threshold** |
Limit attribute. The maximum number of jobs allowed to be queued in this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes. |
R E |
No default |
pbs.pbs_resource Syntax: queued_jobs_threshold["<resource name>"="<value>"] where resource name is any built-in or custom resource |
User Op Mgr |
\- r r r r w w |
# Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>queued_jobs_threshold_res</td>
<td>Limit attribute. The maximum amount of the specified resource allowed to be allocated to jobs queued in this queue. Can be specified for projects, users, groups, or all. Cannot be used with old limit attributes.</td>
<td>Limit specification. See Chapter 7, &quot;Formats&quot;, on page 363. Syntax: &quot;queued_jobs_threshold_res.&lt;resource name&gt;=&lt;value&gt;&quot;</td>
<td>R, E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource Syntax: queued_jobs_threshold_res[&quot;&lt;resource name&gt; &quot;]=&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>queue_type</td>
<td>The type of this queue. This attribute must be explicitly set at queue creation.</td>
<td>String</td>
<td>R, E</td>
<td>e, execution</td>
<td>Execution queue</td>
<td>No default</td>
<td>PBS queue type constant: pbs.QUEUETYPE_EXECUTION</td>
<td>r</td>
<td>r</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r, route</td>
<td></td>
<td></td>
<td>Routing queue</td>
<td></td>
<td>PBS queue type constant: pbs.QUEUETYPE_ROUTE</td>
<td>r</td>
<td>r</td>
<td>R</td>
</tr>
<tr>
<td>require_cred</td>
<td>Specifies the credential type required. All jobs submitted to the named queue without the specified credential will be rejected. Not supported under Windows.</td>
<td>String</td>
<td>R, E</td>
<td>krb5</td>
<td></td>
<td>unset</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dce</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>R</td>
</tr>
<tr>
<td>require_cred_enable</td>
<td>Specifies whether the credential authentication method specified in the require_cred queue attribute is required for this queue. Not supported under Windows.</td>
<td>Boolean</td>
<td>R, E</td>
<td></td>
<td></td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>R</td>
</tr>
<tr>
<td>resources_assigned</td>
<td>The total for each kind of resource allocated to running and exiting jobs in this queue.</td>
<td>String. Syntax: resources_assigned.&lt;resource name&gt;=&lt;value&gt;&lt;new line&gt;resources_assigned.&lt;resource name&gt;=&lt;value&gt;&lt;new line&gt;...</td>
<td>E</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resource Syntax: resources_assigned[&quot;&lt;resource name&gt; &quot;]=&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>R</td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Oper</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>resources-available</td>
<td>The list of resources and amounts available to jobs running in this queue. The sum of the resource of each type used by all jobs running from this queue cannot exceed the total amount listed here.</td>
<td><code>String. Syntax: resources_available.&lt;resource name&gt;=&lt;value&gt;&lt;new line&gt; resources_available.&lt;resource name&gt;=&lt;value&gt;&lt;new line&gt;...</code></td>
<td>E</td>
<td>R, E</td>
<td>No default; zero usage</td>
<td>pbs.pbs_resource Syntax: `resources_available[&quot;&lt;resource name&quot;&gt;]={&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td>resources_default</td>
<td>The list of default resource values which are set as limits for a job residing in this queue and for which the job did not specify a limit. If not set, the default limit for a job is determined by the first of the following attributes which is set: server’s resources_default, queue’s resources_max, server’s resources_max. If none of these is set, the job gets unlimited resource usage.</td>
<td><code>String. Syntax: resources_default.&lt;resource name&gt;=&lt;value&gt;, resources_default.&lt;resource name&gt;=&lt;value&gt;, ...</code></td>
<td>R, E</td>
<td>No default</td>
<td>pbs.pbs_resource Syntax: `resources_default[&quot;&lt;resource name&quot;&gt;]={&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>resources_max</td>
<td>The maximum amount of each resource that can be requested by a single job in this queue. This queue value supersedes any server wide maximum limit.</td>
<td><code>String. Syntax: resources_max.&lt;resource name&gt;=&lt;value&gt;, resources_max.&lt;resource name&gt;=&lt;value&gt;, ...</code></td>
<td>R, E</td>
<td>No default; infinite usage</td>
<td>pbs.pbs_resource Syntax: `resources_max[&quot;&lt;resource name&quot;&gt;]={&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>resources_min</td>
<td>The minimum amount of each resource that can be requested by a single job in this queue.</td>
<td><code>String. Syntax: resources_min.&lt;resource name&gt;=&lt;value&gt;, resources_min.&lt;resource name&gt;=&lt;value&gt;, ...</code></td>
<td>R, E</td>
<td>No default; zero usage</td>
<td>pbs.pbs_resource Syntax: `resources_min[&quot;&lt;resource name&quot;&gt;]={&lt;value&gt; where resource name is any built-in or custom resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Queue Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Queue Type</th>
<th>Value or Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>User</th>
<th>Operator</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>route_destinations</td>
<td>The list of destinations to which jobs may be routed. Must be set to at least one valid destination.</td>
<td>String. Syntax: comma-separated strings: <code>&lt;queue name&gt;</code></td>
<td>R</td>
<td>No default</td>
<td>pbs.route_destinations</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td></td>
<td>Example: Q1, Q2@remote, Q3@remote:15501</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>route_held_jobs</td>
<td>Specifies whether jobs in the held state can be routed from this queue.</td>
<td>Boolean</td>
<td>R</td>
<td>When True, jobs with a hold can be routed from this queue.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>w, r</td>
</tr>
<tr>
<td>route_lifetime</td>
<td>The maximum time a job is allowed to reside in this routing queue. If a job cannot be routed in this amount of time, the job is aborted.</td>
<td>Integer. Units: Seconds</td>
<td>R</td>
<td>&gt;0 Jobs can reside for specified number of seconds</td>
<td>Unset; infinite</td>
<td>pbs.duration</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td></td>
<td>unset Infinite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>route_retry_time</td>
<td>Time delay between routing retries. Typically used when the network between servers is down.</td>
<td>Integer. Units: Seconds</td>
<td>R</td>
<td>30 seconds</td>
<td>pbs.duration</td>
<td>r</td>
<td></td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>route_waiting_jobs</td>
<td>Specifies whether jobs whose Execution_Time attribute value is in the future can be routed from this queue.</td>
<td>Boolean</td>
<td>R</td>
<td>When True, jobs with a future Execution_Time attribute can be routed from this queue.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>started</td>
<td>Specifies whether jobs in this queue can be scheduled for execution.</td>
<td>Boolean</td>
<td>R, E</td>
<td>When True, jobs in this queue can run</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>state_count</td>
<td>The number of jobs in each state currently residing in this queue.</td>
<td>String. Syntax: transiting=&lt;value&gt;, exiting=&lt;value&gt;, ...</td>
<td>R, E</td>
<td>No default</td>
<td>pbs.state_count</td>
<td>r</td>
<td></td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>total_jobs</td>
<td>The number of jobs currently residing in this queue.</td>
<td>Integer</td>
<td>R, E</td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td></td>
<td>r</td>
<td>r, r</td>
</tr>
</tbody>
</table>
## 6.10 Vnode Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>User</th>
<th>Option</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td></td>
<td>Information about this vnode. This attribute may be set by the manager to any string to inform users of any information relating to the node. If this attribute is not explicitly set, the PBS server will use the attribute to pass information about the node status, specifically why the node is down. If the attribute is explicitly set by the manager, it will not be modified by the server.</td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>current_aoe</td>
<td>String</td>
<td></td>
<td>The AOE currently instantiated on this vnode. Case-sensitive. Cannot be set on server's host.</td>
<td></td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>current_eoe</td>
<td>String</td>
<td></td>
<td>Current value of eoe on this vnode. We do not recommend setting this attribute manually.</td>
<td>Unset</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r, w</td>
</tr>
<tr>
<td>in_multivnode_host</td>
<td>Integer</td>
<td>Unset</td>
<td>Specifies whether a vnode is part of a multi-vnoded host. Used internally. Do not set.</td>
<td>Not part of a multi-vnoded host</td>
<td>int</td>
<td>r</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td>jobs</td>
<td>String</td>
<td></td>
<td>List of jobs running on this vnode. Syntax: <code>&lt;processor number&gt;/&lt;job ID&gt;, ...</code></td>
<td></td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>last_state_change_time</td>
<td>Timestamp.</td>
<td></td>
<td>Records the most recent time that this node changed state. Printed by qstat in human-readable Date format. Output in hooks as seconds since epoch.</td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r, r</td>
<td>r, w</td>
</tr>
</tbody>
</table>
# Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>last_used_time</td>
<td>Records the most recent time that this vnode finished being used for a job or reservation. Set at creation or reboot time. Updated when node is released early from a running job. Reset when node is ramped up.</td>
<td>Timestamp.</td>
<td>r</td>
<td>Time of vnode creation or node reboot.</td>
<td>int</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>license</td>
<td>Indicates whether this vnode is socket-licensed. Set by PBS.</td>
<td>Character</td>
<td>r</td>
<td>This vnode is socket licensed.</td>
<td>Unset</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>license_info</td>
<td>Number of socket licenses assigned to this vnode. Set by PBS.</td>
<td>Integer</td>
<td>r</td>
<td></td>
<td>Unset</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>lic_type</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>none</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maintenance_jobs</td>
<td>List of jobs that were running on this vnode, but have been suspended via the admin-suspend signal to qsig. Set by server.</td>
<td>String_array</td>
<td>r</td>
<td>No default</td>
<td></td>
<td>str</td>
<td></td>
<td></td>
<td>r</td>
</tr>
<tr>
<td>Mom</td>
<td>Hostname of host on which MoM daemon runs. Can be explicitly set by Manager only via qmgr, and only at vnode creation. The server can set this to the FQDN of the host on which MoM runs, if the vnode name is the same as the hostname.</td>
<td>String</td>
<td>r</td>
<td>Value of vnode resource (vnode name)</td>
<td></td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>name</td>
<td>The name of this vnode.</td>
<td>String</td>
<td>r</td>
<td>No default</td>
<td></td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>no_multinode_jobs</td>
<td>Controls whether jobs which request more than one chunk are allowed to execute on this vnode. Used for cycle harvesting.</td>
<td>Boolean</td>
<td>r</td>
<td>When set to True, jobs requesting more than one chunk are not allowed to execute on this vnode</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr Op Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntype</td>
<td>String</td>
<td>PBS</td>
<td>Normal vnode; not Globus</td>
<td>PBS</td>
<td>pbs.ND_PBS</td>
<td>r r r w</td>
</tr>
<tr>
<td>partition</td>
<td>String</td>
<td>globus</td>
<td>A Globus vnode (this functionality has been removed). Globus can still send jobs to PBS, but PBS no longer supports sending jobs to Globus.</td>
<td>No default</td>
<td>str</td>
<td>r r r w w</td>
</tr>
<tr>
<td>pbs_version</td>
<td>String</td>
<td></td>
<td>The version of PBS for this MoM</td>
<td>No default</td>
<td>str</td>
<td>r r r r</td>
</tr>
<tr>
<td>pcpus</td>
<td>Integer</td>
<td></td>
<td>The number of physical CPUs on this vnode. This is set to the number of CPUs available when MoM starts. For a multiple-vnode MoM, only the natural vnode has pcpus.</td>
<td>int</td>
<td>Number of CPUs on start-up</td>
<td>r r r</td>
</tr>
<tr>
<td>pnames</td>
<td>String</td>
<td></td>
<td>The list of resources being used for placement sets. Not used for scheduling; advisory only.</td>
<td>No default</td>
<td>str</td>
<td>r r r w w</td>
</tr>
<tr>
<td>Port</td>
<td>Integer</td>
<td></td>
<td>Port number on which MoM daemon listens. Can be explicitly set only via qmgr, and only at vnode creation.</td>
<td>15002</td>
<td>int</td>
<td>- r r r w</td>
</tr>
<tr>
<td>poweroff_eligible</td>
<td>Boolean</td>
<td>True</td>
<td>PBS can power this vnode on and off.</td>
<td>False</td>
<td>bool</td>
<td>r r r w</td>
</tr>
<tr>
<td>power_provisioning</td>
<td>Boolean</td>
<td>True</td>
<td>Power provisioning is enabled at this vnode.</td>
<td>False</td>
<td>bool</td>
<td>r r r w</td>
</tr>
<tr>
<td>Priority</td>
<td>Integer</td>
<td>[-1024, +1023] inclusive</td>
<td>The priority of this vnode compared with other vnodes.</td>
<td>No default</td>
<td>int</td>
<td>r r r w</td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>provision_enable</td>
<td>Controls whether this vnode can be provisioned. Cannot be set on server's host.</td>
<td>Boolean</td>
<td>True</td>
<td>This vnode may be provisioned.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td></td>
<td></td>
<td>False</td>
<td></td>
<td>This vnode may not be provisioned.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>queue</td>
<td>The queue with which this vnode is associated. Each vnode can be associated with at most 1 queue. Queues can be associated with multiple vnodes. Any jobs in a queue that has associated vnodes can run only on those vnodes. If a vnode has an associated queue, only jobs in that queue can run on that vnode.</td>
<td>String</td>
<td>&lt;name of queue&gt;</td>
<td>Only jobs in specified queue may run on this vnode.</td>
<td>No default</td>
<td>pbs.queue</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unset</td>
<td></td>
<td>Any job in any queue that does not have associated vnodes can run on this vnode.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>resources_assigned</td>
<td>The total amount of each resource allocated to running and exiting jobs and started reservations on this vnode.</td>
<td>String</td>
<td></td>
<td>Syntax: &lt;resources_assigned.&lt;resource name&gt;=&lt;value&gt;, ...&lt;/resources_assigned&gt;</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>resources_available</td>
<td>The list of resources and the amounts available on this vnode. If not explicitly set, the amount shown is that reported by the pbs_mom running on this vnode. If a resource value is explicitly set, that value is retained across restarts.</td>
<td>String</td>
<td></td>
<td>Syntax: &lt;resources_available.&lt;resource name&gt;=&lt;value&gt;, ...&lt;/resources_available&gt;</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>where resource name is any built-in or custom resource</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Vnode Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>resv</td>
<td>List of advance and standing reservations pending on this vnode.</td>
<td>String</td>
<td>Comma-</td>
<td>String, Comma-separated list of reservation IDs. Syntax:</td>
<td>No</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td>separated</td>
<td></td>
<td>&lt;reservation ID&gt;[, &lt;reservation ID&gt;, ...]</td>
<td>default</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>resv_enable</td>
<td>Controls whether the vnode can be used for advance and standing reservations.</td>
<td>Boolean</td>
<td></td>
<td>When set to True, this vnode can be used for reservations. Existing reservations are</td>
<td>True</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>r,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>honored when this attribute is changed from True to False.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sharing</td>
<td>Specifies whether more than one job at a time can use the resources of the vnode or the vnode’s host. Either (1) the vnode or host is allocated exclusively to one job, or (2) the vnode’s or host’s unused resources are available to other jobs. Can be set using pbs_mom -s insert only. Behavior of a vnode or host is determined by a combination of the sharing attribute and a job’s placement directive, defined as follows:</td>
<td>String</td>
<td>Example:</td>
<td>String Example: vnodename: sharing=force_excl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>vnodename:</td>
<td>sharing=force_excl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>default_shared Defaults to shared</td>
<td>default</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>default_excl Defaults to exclusive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>default_exclh Host Entire host is assigned to the job unless the job’s sharing request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>specifies otherwise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ignore_excl Overrides any job place=excl setting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>force_excl Overrides any job place=shared setting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>force_exclhost The entire host is assigned to the job, regardless of the job’s sharing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unset Defaults to shared</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Attributes

## Vnode Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr Op Mgr</th>
</tr>
</thead>
</table>

| Behavior of vnode: |

<table>
<thead>
<tr>
<th>Value of sharing</th>
<th>Placement Request (-lplace=)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vnode</td>
</tr>
<tr>
<td></td>
<td>not specified</td>
</tr>
<tr>
<td>not set</td>
<td>shared</td>
</tr>
<tr>
<td><strong>default_shared</strong></td>
<td>shared</td>
</tr>
<tr>
<td><strong>default_excl</strong></td>
<td>exclusive</td>
</tr>
<tr>
<td><strong>default_exclhost</strong></td>
<td>exclusive</td>
</tr>
<tr>
<td><strong>ignore_excl</strong></td>
<td>shared</td>
</tr>
<tr>
<td><strong>force_excl</strong></td>
<td>exclusive</td>
</tr>
<tr>
<td><strong>force_exclhost</strong></td>
<td>exclusive</td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>Shows or sets the state of the vnode.</td>
<td><code>String</code></td>
<td></td>
<td><code>busy</code> Vnode is reporting load average greater than allowed max. Can combine with <code>offline</code>.</td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>down</code> Node is not responding to queries from the server. Cannot be combined with <code>free, provisioning</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>free</code> Vnode is up and capable of accepting new job(s). Cannot be combined with other states.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>job-busy</code> All CPUs on the vnode are allocated to jobs. Can combine with: <code>offline, resv_exclusive</code>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>job-exclusive</code> Entire vnode is exclusively allocated to one job at the job's request. Can combine with <code>offline, resv_exclusive</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>offline</code> Jobs are not to be assigned to this vnode. Can combine: <code>busy, job-busy, job-exclusive, resv_exclusive</code>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>provisioning</code> Vnode is being provisioned. Cannot be combined with any other states.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>resv-exclusive</code> Running reservation has requested exclusive use of vnode. Can combine with <code>job-exclusive, offline</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>stale</code> Vnode was previously reported to server, but is no longer reported to server. Cannot combine with <code>free, provisioning</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>state-unknown</code> The server has never been able to contact the vnode. Either MoM is not running on the vnode, the vnode hardware is down, or there is a network problem.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>unresolvable</code> The server cannot resolve the name of the vnode.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><code>wait-provisioning</code> Vnode needs to be provisioned, but can't: limit reached for concurrent provisioning vnodes. Cannot be combined with other states. See <code>max_concurrent_provision</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>topology_info</td>
<td>Contains information intended to be used in hooks. Visible in and usable by hooks only.</td>
<td>XML string</td>
<td></td>
<td></td>
<td>Unset</td>
<td>str</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>


### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
</table>
| vnode_pool   | Cray only. Allows just one MoM, instead of all, to report inventory upon startup, allowing faster startup and less network communication between server and non-reporting MoMs. On each Cray, all MoMs must have same setting for this attribute. Can be set only at vnode creation; valid only on login nodes running a MoM. Not supported on non-Cray machines. | Integer| 0         | Unset; each MoM reports inventory separately  | 0       | int         | r   | r   | r, w
|              |                                                                                                                                                                                                             |        | >0        | Only one MoM per Cray reports inventory        | (Unset) |             |     |     |     |
## 6.11 Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account_Name</td>
<td>String used for accounting purposes. Can be used for fairshare.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>w</td>
<td>r</td>
</tr>
<tr>
<td>accounting_id</td>
<td>Accounting ID for tracking accounting data not produced by PBS.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>accru_type</td>
<td>Indicates what kind of time the job is accruing.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>2</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>0 (initial_time) Job is accruing initial time. Can occur when job is blocked by a runjob hook.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 (ineligible_time) Job is accruing ineligible time. Occurs when job or owner has hit limit.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 (eligible_time) Job is accruing eligible time. Occurs when job is blocked on resources.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 (run_time) Job is accruing run time. Occurs when job is running.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alt_id</td>
<td>For a few systems, the session ID is insufficient to track which processes belong to the job. Where a different identifier is required, it is recorded in this attribute. If set, it is also recorded in the end-of-job accounting record. For jobs running in CPU sets, the alt_id holds the set name in a form usable by the cpuset(1) command; this is cpuset=&lt;path to cpuset&gt;. On Windows, holds PBS home directory.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
### Attributes

#### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>array</td>
<td>Indicates whether this is a job array.</td>
<td>Boolean</td>
<td></td>
<td>Set to True if this is an array job.</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>s</td>
<td>r</td>
</tr>
<tr>
<td>array_id</td>
<td>Applies only to subjobs. Array identifier of subjob.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>array_index</td>
<td>Applies only to subjobs. Index number of subjob.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>array_indices_remaining</td>
<td>Applies only to job arrays. List of indices of subjobs still queued.</td>
<td>String. Range or list of ranges, e.g. 500, 552, 596-1000.</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>array_indices_submitted</td>
<td>Applies only to job arrays. Complete list of indices of subjobs given at submission time.</td>
<td>String. Given as range, e.g. 1-100</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.range</td>
<td>r</td>
<td>s</td>
<td>r</td>
</tr>
<tr>
<td>array_state_count</td>
<td>Applies only to job arrays. Lists number of subjobs in each state.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.state_count</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
### Attributes

#### Chapter 6

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>block</td>
<td>Specifies whether <code>qsub</code> will wait for the job to complete and return the exit value of the job. For X11 forwarding jobs, and jobs with <code>interactive</code> and/or <code>block</code> attributes set to <code>True</code>, the job's exit status is not returned.</td>
<td><strong>Boolean</strong></td>
<td></td>
<td></td>
<td><code>False</code></td>
<td><code>int</code></td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>Checkpoint</td>
<td>Determines when the job will be checkpointed. An Saction script is required to checkpoint the job.</td>
<td><strong>String</strong></td>
<td><code>c</code></td>
<td>Checkpoint at intervals, measured in CPU time, set on job’s execution queue. If no interval set at queue, job is not checkpointed.</td>
<td><code>u</code></td>
<td><code>pbs.checkpoint</code></td>
<td>r</td>
<td>w</td>
<td>w</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>c = &lt;minutes of CPU time&gt;</code></td>
<td>Checkpoint at intervals of specified number of minutes of job CPU time. This value must be &gt; 0. If interval specified is less than that set on job’s execution queue, queue's interval is used. Format: Integer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>w</code></td>
<td>Checkpoint at intervals, measured in walltime, set on job’s execution queue. If no interval set at queue, job is not checkpointed.</td>
<td><code>w = &lt;minutes of walltime&gt;</code></td>
<td><code>Integer</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>n</code></td>
<td>No checkpointing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>s</code></td>
<td>Checkpoint only when the server is shut down.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>u</code></td>
<td>Unset. Defaults to behavior when <code>interval</code> argument is set to <code>s</code>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>Comment about job. Informational only.</td>
<td><strong>String</strong></td>
<td></td>
<td></td>
<td><code>No default</code></td>
<td><code>str</code></td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
</tbody>
</table>

**Val / Opt:**
- `r`: Read
- `w`: Write
## Attributes

### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Ust</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctime</td>
<td>Timestamp; time at which the job was created.</td>
<td><code>Timestamp</code>. Printed by <code>qstat</code> in human-readable <code>Date</code> format. Output in hooks as seconds since epoch.</td>
<td></td>
<td></td>
<td></td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>depend</td>
<td>Specifies inter-job dependencies. No limit on number of dependencies.</td>
<td><code>String</code>. Syntax: <code>&lt;type&gt;::&lt;job ID&gt;[,&lt;job ID&gt; ...],[&lt;type&gt;::&lt;job ID&gt; ...]</code>. Must be enclosed in double quotes if it contains commas. Example: “before:123,456”</td>
<td>after:&lt;job ID list&gt;</td>
<td>This job may run at any point after all jobs in job ID list have started execution.</td>
<td>No default</td>
<td>pbs.depend</td>
<td>r, w, r, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>afterok:&lt;job ID list&gt;</td>
<td>This job may run only after all jobs in job ID list have terminated with no errors.</td>
<td>No default; no dependencies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>afterno:tok:&lt;job ID list&gt;</td>
<td>This job may run only after all jobs in job ID list have terminated with errors.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>after-any:&lt;job ID list&gt;</td>
<td>This job can run after all jobs in job ID list have finished execution, with or without errors. This job will not run if a job in the job ID list was deleted without ever having been run.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before:&lt;job ID list&gt;</td>
<td>Jobs in job ID list may start once this job has started.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>beforeok:&lt;job ID list&gt;</td>
<td>Jobs in job ID list may start once this job terminates without errors.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before:tok:&lt;job ID list&gt;</td>
<td>If this job terminates execution with errors, jobs in job ID list may begin.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before-any:&lt;job ID list&gt;</td>
<td>Jobs in job ID list may begin execution once this job terminates execution, with or without errors.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>on:&lt;count&gt;</td>
<td>This job may run after count dependences on other jobs have been satisfied. This type is used with one of the before types listed. Count is an integer greater than 0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>egroup</strong></td>
<td>If the job is queued, this attribute is set to the group name under which the job is to be run.</td>
<td><strong>String</strong></td>
<td></td>
<td>No default</td>
<td>str</td>
<td></td>
<td></td>
<td></td>
<td>r</td>
</tr>
<tr>
<td><strong>eligible_time</strong></td>
<td>The amount of wall clock wait time a job has accrued while the job is blocked waiting for resources. For a job currently accruing eligible_time, if we were to add enough of the right type of resources, the job would start immediately. Viewable via qstat -f.</td>
<td><strong>Duration</strong></td>
<td></td>
<td>Zero</td>
<td>pbs-duration</td>
<td></td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
</tbody>
</table>
| **Error_Path** | The final path name for the file containing the job’s standard error stream. See the qsub and qalter commands. | **String** Syntax: 
- `<hostname>[:]<path>` Path is relative to the current working directory of command executing on current host.  
- `<host-name>[:]<relative path>` Path is relative to user’s home directory on specified host.  
- `<host-name>[:]<absolute path>` Path is absolute path on named host.  
- No path Path is current working directory where qsub is executed.  
- Default path is current working directory where qsub is run. If the output path is specified, but does not include a filename, the default filename is `<job ID>.e<sequence number>`. |           | Default path is current working directory where qsub is run. If the output path is specified, but does not include a filename, the default filename is `<job ID>.e<sequence number>`. | str     |             | r   | r   | w   | w   | w   | w   | w   |
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>estimated</strong></td>
<td>List of estimated values for job. Used to report job’s <code>exec_vnode</code>, <code>start_time</code>, and <code>soft_walltime</code>. Can be set in a hook or via <code>qalter</code>, but PBS will overwrite the values.</td>
<td><code>exec_vnode</code></td>
<td>Syntax: <code>estimated.&lt;resource name&gt;==&lt;value&gt;</code>, <code>estimated.&lt;resource name&gt;==&lt;value&gt;</code>. <code>exec_vnode</code> is a string. <code>soft_walltime</code> is a duration. <code>start_time</code> is printed by <code>qstat</code> in human-readable Date format; <code>start_time</code> is output in hooks as seconds since epoch.</td>
<td>The estimated vnodes used by this job.</td>
<td>Unset</td>
<td>pbs.pbs_resource Syntax: <code>estimated.[&lt;resource name&gt;]==&lt;value&gt;</code>. <code>exec_vnode</code> is a string. <code>soft_walltime</code> is a duration. <code>start_time</code> is an int.</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td><strong>etime</strong></td>
<td>Timestamp; time when job became eligible to run, i.e. was enqueued in an execution queue and was in the “Q” state. Reset when a job moves queues, or is held then released. Not affected by <code>qalter</code>.</td>
<td>Timestamp.</td>
<td>Printed by <code>qstat</code> in human-readable Date format. Output in hooks as seconds since epoch.</td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td><strong>euser</strong></td>
<td>If the job is queued, this attribute is set to the user name under which the job is to be run.</td>
<td>String</td>
<td>No default</td>
<td>str</td>
<td>-</td>
<td>-</td>
<td>r</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>executable</strong></td>
<td>JSDL-encoded listing of job’s executable. Shown if job is submitted with “-- &lt;executable&gt; [&lt;arg list&gt;]”.</td>
<td>JSDL-encoded string. <code>&lt;jsdl-hpcpa:Executable&gt; &lt;name of executable&gt;</code> Example: if the executable is <code>ping</code>: <code>&lt;jsdl-hpcpa:Executable&gt;ping&lt;/jsdl-hpcpa:Executable&gt;</code></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>

The estimated vnodes used by this job. Can be set in a hook or via `qalter`, but PBS will overwrite the values. ```bash```  ```python```  

The estimated soft walltime for this job. Calculated when a job exceeds its `soft_walltime` resource. ```bash```  ```python```  

The estimated start time for this job. ```bash```  ```python```  

Timestamp; time when job became eligible to run, i.e. was enqueued in an execution queue and was in the “Q” state. Reset when a job moves queues, or is held then released. Not affected by `qalter`. ```bash```  ```python```  

If the job is queued, this attribute is set to the user name under which the job is to be run. ```bash```  ```python```  

JSDL-encoded listing of job’s executable. Shown if job is submitted with “-- <executable> [<arg list>]”. ```bash```  ```python```
## Attributes

### Chapter 6

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execution_Time</strong></td>
<td>Timestamp; time after which the job may execute. Before this time, the job remains queued in the (W)ait state. Can be set when stage-in fails and PBS moves job start time out 30 minutes to allow user to fix problem.</td>
<td><strong>Date/Time</strong>. See Chapter 7, &quot;Formats&quot;, on page 363.</td>
<td>Unset; no delay</td>
<td>int</td>
<td>r, r, r, w</td>
<td>w, w, w</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>exec_host</strong></td>
<td>If the job is running, this is set to the name of the host or hosts on which the job is executing.</td>
<td><strong>String</strong>. Syntax: <code>&lt;hostname&gt;/N[^C][+...]/</code>, where N is task slot number starting at 0, on that host, and C is the number of CPUs allocated to the job. *C does not appear if its value is 1.</td>
<td>No default</td>
<td>pbs.exec_host</td>
<td>r</td>
<td>r, i, r, i</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>exec_vnode</strong></td>
<td>List of chunks for the job. Each chunk shows the name of the vnode(s) from which it is taken, along with the host-level, consumable resources allocated from that vnode, and any AOE provisioned on this vnode for this job. If a vnode is allocated to the job but no resources from the vnode are used by the job, the vnode name appears alone. If a chunk is split across vnodes, the names of each vnode and its resources appear inside one pair of parentheses, joined with a plus (&quot;+&quot;) sign.</td>
<td>Each chunk is enclosed in parentheses. Chunks are connected by plus signs. Example: For a job which requested two chunks satisfied by resources from three vnodes, exec_vnode is: <code>(vnodeA:ncpus=N:mem=X)+ (nodeB:ncpus=P:mem=Y+ nodeC:mem=Z)</code>. For a job which requested one chunk and exclusive use of a 2-vnode host, where the chunk was satisfied by resources from one vnode, exec_vnode is: <code>(vnodeA:ncpus=N:mem=X)+(vnodeB)</code>.</td>
<td>No default</td>
<td>pbs.exec_vnode</td>
<td>r</td>
<td>r, w, r, w</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Exit_status
- **Description**: Exit status of job. Set to zero for successful execution. If any subjob of an array job has non-zero exit status, the array job has non-zero exit status.
- **Type**: Integer
- **Default**: No default
- **Python Type**: int
- **Access**: r, r, r

### forward_x11_cookie
- **Description**: Contains the X authorization cookie.
- **Type**: String
- **Default**: No default
- **Python Type**: str
- **Access**: r, r, r

### forward_x11_port
- **Description**: Contains the number of the port being listened to by the port forwarder on the submission host.
- **Type**: Integer
- **Default**: No default
- **Python Type**: int
- **Access**: r, r, r

### group_list
- **Description**: A list of group names used to determine the group under which the job runs. When a job runs, the server selects a group name from the list according to the following ordered set of rules:
  1. Select the group name for which the associated host name matches the name of the server host.
  2. Select the group name which has no associated host name.
  3. Use the login group for the user name under which the job will be run.
- **Type**: String
  **Syntax**: `<group name>[@<hostname>] [,<group name>[@<hostname>]...]`
  **Default**: No default
  **Python Type**: pbs.group_list
  **Access**: r, w, w, r, w

### Hashname
- **Description**: No longer used.
- **Type**: -
- **Default**: -
- **Python Type**: -
- **Access**: -

### Hold_Types
- **Description**: The set of holds currently applied to the job. If the set is not null, the job will not be scheduled for execution and is said to be in the held state. The held state takes precedence over the wait state.
- **Type**: String, made up of the letters 'n', 'o', 'p', 's', 'u'
- **Default**: n
- **Python Type**: pbs.hold_types
- **Access**: r, r, r, w, w, w

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit_status</td>
<td>Exit status of job. Set to zero for successful execution. If any subjob of an array job has non-zero exit status, the array job has non-zero exit status.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>forward_x11_cookie</td>
<td>Contains the X authorization cookie.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>forward_x11_port</td>
<td>Contains the number of the port being listened to by the port forwarder on the submission host.</td>
<td>Integer</td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
| group_list    | A list of group names used to determine the group under which the job runs. When a job runs, the server selects a group name from the list according to the following ordered set of rules:
  1. Select the group name for which the associated host name matches the name of the server host.
  2. Select the group name which has no associated host name.
  3. Use the login group for the user name under which the job will be run. | String  |           | `<group name>[@<hostname>] [,<group name>[@<hostname>]...]` | No default | pbs.group_list | r   | w   | w   |
| Hashname      | No longer used.                                                              | -      |           |                           | -      | -           | -   | -   | -   |
| Hold_Types    | The set of holds currently applied to the job. If the set is not null, the job will not be scheduled for execution and is said to be in the held state. The held state takes precedence over the wait state. | String  |           |                           | n      | pbs.hold_types | r   | r   | r   |
## Attributes

### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>interactive</strong></td>
<td>Specifies whether the job is interactive. Can be set, but not altered, by unprivileged user. When both this attribute and the block attribute are True, no exit status is returned. For X11 forwarding jobs, the job's exit status is not returned. Cannot be set using a PBS directive. Job arrays cannot be interactive.</td>
<td>Boolean</td>
<td></td>
<td>Set to True if this is an interactive job.</td>
<td>False</td>
<td>int</td>
<td>r</td>
<td>w</td>
<td>r</td>
</tr>
<tr>
<td><strong>jobdir</strong></td>
<td>Path of the job’s staging and execution directory on the primary execution host. Either user’s home, or private sandbox. Depends on value of sandbox attribute. Viewable via qstat -f.</td>
<td>String</td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td><strong>Job_Name</strong></td>
<td>The job name. See the qalter and qsub commands.</td>
<td>String</td>
<td></td>
<td>up to 236 characters, first character must be alphabetic or numeric</td>
<td>str</td>
<td>r, w, r, r</td>
<td>w</td>
<td>w</td>
<td>w</td>
</tr>
<tr>
<td><strong>Job_Owner</strong></td>
<td>The login name on the submitting host of the user who submitted the batch job.</td>
<td>String</td>
<td></td>
<td>Syntax: &lt;Username&gt;@&lt;submission host&gt;</td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>

**Example:**

- To set `interactive` to `True`, use the `qalter` command:

  ```bash
  qalter -N job1 -i
  ```

- To check the `interactive` attribute:

  ```bash
  qstat -f job1
  ```
# Attributes

## Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>job_state</td>
<td>The state of the job.</td>
<td>Character</td>
<td>B (Begun)</td>
<td>Job arrays only. Job array has begun execution.</td>
<td>No default</td>
<td>pbs.JOB_STATE_BEGUN</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>E (Exiting)</td>
<td>The job has finished, with or without errors, and PBS is cleaning up post-execution.</td>
<td></td>
<td>pbs.JOB_STATE_EXITING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F (Finished)</td>
<td>Job is finished. Job has completed execution, job failed during execution, or job was deleted.</td>
<td></td>
<td>pbs.JOB_STATE_FINISHED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H (Held)</td>
<td>The job is held.</td>
<td></td>
<td>pbs.JOB_STATE_HELD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>M (Moved)</td>
<td>The job has been moved to another server.</td>
<td></td>
<td>pbs.JOB_STATE_MOVED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Q (Queued)</td>
<td>The job resides in an execution or routing queue pending execution or routing. It is not in held or waiting state.</td>
<td></td>
<td>pbs.JOB_STATE_QUEUED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R (Running)</td>
<td>The job is in an execution queue and is running.</td>
<td></td>
<td>pbs.JOB_STATE_RUNNING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S (Suspended)</td>
<td>The job was executing and has been suspended. The job does not use CPU cycles or walltime.</td>
<td></td>
<td>pbs.JOB_STATE_SUSPEND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T (Transiting)</td>
<td>The job is being routed or moved to a new destination.</td>
<td></td>
<td>pbs.JOB_STATE_TRANSIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U (User suspended)</td>
<td>The job was running on a workstation configured for cycle harvesting and the keyboard/mouse is currently busy. The job is suspended until the workstation has been idle for a configured amount of time.</td>
<td></td>
<td>pbs.JOB_STATE_SUSPEND_UUSERACTIVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>W (Waiting)</td>
<td>The Execution_Time attribute contains a time in the future. Can be set when stage-in fails and PBS moves job start time out 30 minutes to allow user to fix problem.</td>
<td></td>
<td>pbs.JOB_STATE_WAITING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X (Expired)</td>
<td>Subjobs only. Subjob is finished (expired.)</td>
<td></td>
<td>pbs.JOB_STATE_EXPIRED</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

#### Chapter 6

#### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Join_Path</strong></td>
<td>Specifies whether the job’s standard error and standard output streams are to be merged and placed in the file specified in the <strong>Output_Path</strong> job attribute.</td>
<td>String</td>
<td></td>
<td><strong>eo</strong></td>
<td></td>
<td>pbs.join_path</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Standard output and standard error are merged, intermixed, into a single stream, which becomes standard error.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>oe</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Standard output and standard error are merged, intermixed, into a single stream, which becomes standard output.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>n</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Standard output and standard error are not merged.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Keep_Files</strong></td>
<td>Specifies whether the standard output and/or standard error streams are retained on the execution host in the job’s staging and execution directory after the job has executed. Otherwise these files are returned to the submission host. <strong>Keep_Files</strong> overrides the <strong>Output_Path</strong> and <strong>Error_Path</strong> attributes.</td>
<td>String</td>
<td></td>
<td><strong>o</strong></td>
<td></td>
<td>pbs.keep_files</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The standard output stream is retained. The filename is:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>job_name.o&lt;sequence_number&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>e</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The standard error stream is retained. The filename is:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>job_name.e&lt;sequence_number&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>eo, oe</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Both standard output and standard error streams are retained.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>d</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Output and error are written directly to their final destination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>n</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Neither stream is retained. Files are returned to submission host.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mail_Points</strong></td>
<td>Specifies state changes for which the server sends mail about the job.</td>
<td>String</td>
<td></td>
<td><strong>a</strong></td>
<td></td>
<td>pbs.mail_points</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mail is sent when job is aborted</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>b</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mail is sent at beginning of job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>e</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mail is sent at end of job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>j</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mail is sent for subjobs. Must be combined with one or more of <strong>a, b, and e</strong> options</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>n</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No mail is sent. Cannot be combined with other options.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mail_Users</strong></td>
<td>The set of users to whom mail is sent when the job makes state changes specified in the <strong>Mail_Points</strong> job attribute.</td>
<td>String</td>
<td>Syntax: <code>&lt;user-name&gt;@&lt;hostname&gt;[,&lt;username&gt;@&lt;hostname&gt;]</code> Must be enclosed in double quotes if it contains commas.</td>
<td>Job owner only</td>
<td>pbs.email_list</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
</tr>
</tbody>
</table>
# Attributes

## mtime
- **Name**: mtime
- **Description**: Timestamp; the time that the job was last modified, changed state, or changed locations.
- **Format**: 
  - Timestamp.
  - Printed by `qstat` in human-readable Date format.
  - Output in hooks as seconds since epoch.
- **Value/Option Description**: No default
- **Python Type**: int
- **Usr/Opt/Mgr**: r r r

## no_stdio_sockets
- **Description**: Not used.
- **Format**: 
  - String
  - Syntax: `no_stdio_sockets`
- **Value/Option Description**: No path
- **Python Type**: 
- **Usr/Opt/Mgr**: - - -

## Output_Path
- **Description**: The final path name for the file containing the job's standard output stream. See the `qsub` and `qalter` commands.
- **Format**: 
  - String
  - Syntax: `Output_Path`<relative path> /<absolute path> <host-name>:<relative path> [host-name]:<absolute path>
  - Path is relative to the current working directory of command executing on current host.
  - Path is absolute path on current host where command is executing.
  - Path is relative to user's home directory on specified host.
  - Path is absolute path on named host.
  - Path is current working directory where `qsub` is executed.
- **Value/Option Description**: Default path is current working directory where `qsub` is run.
  - If the output path is specified, but does not include a filename, the default filename is `<job ID>.OU`.
  - If the path name is not specified, the default filename is `<job name>.o<sequence number>`.
- **Python Type**: str
- **Usr/Opt/Mgr**: r r r r

## pcap_accelerator
- **Description**: Power attribute. Power cap for an accelerator. Corresponds to Cray `capmc set_power_cap --accel setting`. See `capmc` documentation.
- **Format**: 
  - Integer
  - Units: Watts
- **Value/Option Description**: Unset
- **Python Type**: int
- **Usr/Opt/Mgr**: r r r r
### Attributes

#### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val/Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcap_node</td>
<td>Power attribute. Power cap for a node. Corresponds to Cray capmc set_power_cap --node setting. See capmc documentation.</td>
<td>Integer</td>
<td></td>
<td>Unset</td>
<td>int</td>
<td></td>
<td>r,</td>
<td>r,</td>
<td>w,</td>
</tr>
<tr>
<td>pgov</td>
<td>Power attribute. Cray ALPS reservation setting for CPU throttling corresponding to p-governor. See BASIL 1.4 documentation.</td>
<td>String</td>
<td></td>
<td>Unset</td>
<td>str</td>
<td></td>
<td>r,</td>
<td>r,</td>
<td>w,</td>
</tr>
<tr>
<td>Priority</td>
<td>The scheduling priority for the job. Higher value indicates greater priority.</td>
<td>Integer</td>
<td>Syntax:</td>
<td>[-1024, +1023] inclusive</td>
<td>Unset</td>
<td>int</td>
<td>r,</td>
<td>r,</td>
<td>w,</td>
</tr>
<tr>
<td>project</td>
<td>The job’s project. A project is a way to tag jobs. Each job can belong to at most one project.</td>
<td>String</td>
<td></td>
<td>_pbs_project_default</td>
<td>str</td>
<td></td>
<td>r,</td>
<td>w,</td>
<td>w,</td>
</tr>
<tr>
<td>pset</td>
<td>Name of placement set used by the job.</td>
<td>String</td>
<td></td>
<td>Unset</td>
<td>str</td>
<td></td>
<td>r,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pstate</td>
<td>Power attribute. Cray ALPS reservation setting for CPU frequency corresponding to p-state. See BASIL 1.4 documentation.</td>
<td>String</td>
<td></td>
<td>Unset</td>
<td>str</td>
<td></td>
<td>r,</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>qtime</td>
<td>Timestamp.</td>
<td>Val</td>
<td>Time that the job entered the current queue. Printed by qstat in human-readable Date format. Output in hooks as seconds since epoch.</td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>queue</td>
<td>String</td>
<td>No default</td>
<td>Name of the queue in which the job currently resides.</td>
<td>pbs.queue</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>queue_rank</td>
<td>Integer</td>
<td>No default</td>
<td>A number indicating the job’s position within its queue. Only used internally by PBS.</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td>queue_type</td>
<td>Character</td>
<td>E</td>
<td>Execution queue</td>
<td>no default</td>
<td>pbs.QTYPE_EXECUTION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R</td>
<td>Routing queue</td>
<td></td>
<td>pbs.QTYPE_ROUTE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>release_nodes_on_stageout</td>
<td>Boolean</td>
<td>True</td>
<td>All of the job’s vnodes not on the primary execution host are released when stageout begins</td>
<td>False</td>
<td>bool</td>
<td>r</td>
<td>r</td>
<td>w</td>
</tr>
<tr>
<td></td>
<td></td>
<td>False</td>
<td>Job’s vnodes are released when the job finishes and MoM cleans up the job.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Ust</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove_Files</td>
<td>String</td>
<td>e</td>
<td>Standard error is removed upon job completion</td>
<td>Unset</td>
<td>str</td>
<td>r</td>
<td>w</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o</td>
<td>Standard output is removed upon job completion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>eo</td>
<td>Standard output and standard error are removed upon job completion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>oe</td>
<td>Standard output and standard error are removed upon job completion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>unset</td>
<td>Neither is removed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rerunable</td>
<td>Character</td>
<td>y</td>
<td>The job can be rerun.</td>
<td>y</td>
<td>bool</td>
<td>r</td>
<td>w</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td>n</td>
<td>Once the job starts running, it can never be rerun.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource_List</td>
<td>String. Syntax: Resource_List.&lt;resource name&gt;=&lt;value&gt;, Resource_List.&lt;resource name&gt;=&lt;value&gt;, ...</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.pbs_resourc e</td>
<td>r</td>
<td>w</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>resources_released</td>
<td>String. Syntax: (&lt;vnode&gt;:&lt;resource name&gt;=&lt;value&gt;:&lt;resource name&gt;=&lt;value&gt;:...)+(&lt;vnode&gt;:&lt;resource name&gt;=&lt;value&gt;:...))</td>
<td></td>
<td></td>
<td>No default</td>
<td>str</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Format</td>
<td>Val / Opt</td>
<td>Value/Option Description</td>
<td>Def Val</td>
<td>Python Type</td>
<td>USR</td>
<td>OPT</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
<td>-------------</td>
<td>---</td>
<td>-----</td>
</tr>
<tr>
<td>resource_released_list</td>
<td>Sum of each consumable resource requested by the job that was released when the job was suspended. Populated only when restrict_res_to_release_on_suspend server attribute is set. Set by server.</td>
<td><strong>String. Syntax:</strong> resource_released_list. &lt;resource name&gt;=&lt;value&gt;, resource_released_list.&lt;resource name&gt;=&lt;value&gt;, ...</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>resources_used</td>
<td>The amount of each resource used by the job.</td>
<td><strong>String. Syntax:</strong> List of resources_used.&lt;resource name&gt;=&lt;value&gt;, resources_used.&lt;resource name&gt;=&lt;value&gt; pairs. Example: resources_used.mem=2 mb</td>
<td>No default</td>
<td>pbs.pbs_resource</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td>run_count</td>
<td>The number of times the server thinks the job has been executed. The run_count attribute starts at zero. Job is held after 21 tries. Can be set via qsub, qalter, or a hook.</td>
<td><strong>Integer.</strong> Must be greater than or equal to zero.</td>
<td>Zero</td>
<td>int</td>
<td></td>
<td></td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>run_version</td>
<td>Used internally by PBS to track the instance of the job.</td>
<td><strong>Integer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sandbox</td>
<td>Specifies type of location PBS uses for job staging and execution. User-settable via qsub -Wsandbox=VALUE or via a PBS directive. See the $jobdir_root MoM configuration option in pbs_mom.8B.</td>
<td><strong>String</strong> PRIVATE, HOME or unset</td>
<td>Unset</td>
<td>str</td>
<td></td>
<td></td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>schedselect</td>
<td>The union of the select specification of the job, and the queue and server defaults for resources in a chunk.</td>
<td><strong>String</strong></td>
<td>No default</td>
<td>pbs.select</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Job Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>sched_hint</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>server</td>
<td>The name of the server which is currently managing the job. When the secondary server is running during failover, shows the name of the primary server. After a job is moved to another server, either via qmove or peer scheduling, shows the name of the new server.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.server</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>session_id</td>
<td>If the job is running, this is set to the session ID of the first executing task.</td>
<td></td>
<td></td>
<td></td>
<td>No default</td>
<td>int</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>Shell_Path_List</td>
<td>One or more absolute paths to the program(s) to process the job’s script file.</td>
<td>String</td>
<td></td>
<td></td>
<td>User’s login shell on execution host</td>
<td>pbs.path_list</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>stagein</td>
<td>The list of files to be staged in prior to job execution.</td>
<td>String</td>
<td></td>
<td></td>
<td>No default</td>
<td>pbs.staging_list</td>
<td>r</td>
<td>w</td>
<td>w</td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opt</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>stageout</td>
<td>The list of files to be staged out after job execution.</td>
<td>String</td>
<td>r, w, r</td>
<td></td>
<td>No default</td>
<td>pbs.staging_list</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stageout_status</td>
<td>Status of stageout. If stageout succeeded, this is set to 1. If stageout failed, this is set to 0. Available only for finished jobs. Displayed only if set. If stageout fails for any subjob of an array job, the value of Stageout_status is zero for the array job. Available only for finished jobs.</td>
<td>Integer</td>
<td>r, r, r</td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stime</td>
<td>Timestamp; time when the job started execution. Changes when job is restarted.</td>
<td>Timestamp</td>
<td>r, r, r</td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submit_arguments</td>
<td>Job submission arguments given on the qsub command line. Available for all jobs.</td>
<td>String</td>
<td>r, r, r</td>
<td></td>
<td>No default</td>
<td>str</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>substate</td>
<td>The substate of the job. The substate is used internally by PBS.</td>
<td>Integer</td>
<td>r, r, r</td>
<td></td>
<td>No default</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sw_index</td>
<td>No longer used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>topjob_ineligible</td>
<td>Allows administrators to mark this job as ineligible to be a top job.</td>
<td>Boolean</td>
<td></td>
<td></td>
<td></td>
<td>bool</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Def Val</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>umask</td>
<td>The initial umask of the job is set to the value of this attribute when the job is created. This may be changed by <code>umask</code> commands in the shell initialization files such as <code>.profile</code> or <code>.cshrc.</code></td>
<td><em>Decimal integer</em></td>
<td></td>
<td></td>
<td>077</td>
<td>int</td>
<td>r,</td>
<td>r,</td>
<td>r,</td>
</tr>
<tr>
<td>User_List</td>
<td>The list of users which determines the user name under which the job is run on a given host. No length limit. When a job is to be executed, the server selects a user name from the list according to the following ordered set of rules: 1. Select the user name from the list for which the associated host name matches the name of the server. 2. Select the user name which has no associated host name; the wild card name. 3. Use the value of Job_Owner as the user name.</td>
<td><em>String. Syntax: “&lt;username&gt;@&lt;hostname&gt;[,&lt;username&gt;@&lt;hostname&gt;...]” Must be enclosed in double quotes if it contains commas. May be up to 256 characters in length.</em></td>
<td></td>
<td>Value of Job_Owner job attribute</td>
<td></td>
<td>pbs.user_list</td>
<td>r,</td>
<td>r,</td>
<td>r,</td>
</tr>
<tr>
<td>Variable_List</td>
<td>List of environment variables set in the job’s execution environment. See the <code>qsub(1B)</code> command.</td>
<td><em>String. Syntax: “&lt;variable name&gt;=&lt;value&gt;[,&lt;variable name&gt;=&lt;value&gt;...]” Must be enclosed in double quotes if it contains commas.</em></td>
<td></td>
<td></td>
<td></td>
<td>pbs.pbs_resource</td>
<td>r,</td>
<td>r,</td>
<td>r,</td>
</tr>
</tbody>
</table>
### 6.12 Hook Attributes

An unset hook attribute takes the default value for that attribute.

Hook attributes can be set by root or the Admin at the local server only.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>alarm</td>
<td>Specifies the number of seconds to allow a hook to run before the hook times out.</td>
<td>Integer</td>
<td>Must be greater than zero.</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug</td>
<td>Specifies whether or not the hook produces debugging files under PBS_HOME/server_priv/hooks/tmp or PBS_HOME/mom_priv/hooks/tmp. Files are named hook_&lt;hook event&gt;<em>&lt;hook name&gt;</em>&lt;unique ID&gt;.in, .data, and .out. See &quot;Producing Files for Debugging&quot; on page 149 in the PBS Professional Hooks Guide.</td>
<td>Boolean</td>
<td>True</td>
<td>The hook leaves debugging files when it runs. False</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enabled</td>
<td>Determines whether or not a hook is run when its triggering event occurs.</td>
<td>Boolean</td>
<td>True</td>
<td>Hook runs when triggering event occurs. False</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

#### Hook Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
<th>Default Value</th>
<th>Python Type</th>
<th>Type</th>
<th>Usr</th>
<th>Opr</th>
<th>Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>List of events that trigger the hook. Can be operated on with the &quot;=&quot; &quot;+=&quot;, and &quot;+=&quot; operators. The provision event cannot be combined with any other events.</td>
<td>String array</td>
<td>queuejob</td>
<td>Triggered when job is queued</td>
<td>*** meaning hook is not triggered</td>
<td>str</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>modifyjob</td>
<td>Triggered when job is modified</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>movejob</td>
<td>Triggered when job is moved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>resvsub</td>
<td>Triggered when reservation is created</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>runjob</td>
<td>Triggered when job is run</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>periodic</td>
<td>Triggered periodically at server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>provision</td>
<td>Hook is master provisioning hook</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>execjob_begin</td>
<td>Triggered when MoM receives job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>execjob_prologue</td>
<td>Triggered just before first job process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>execjob_launch</td>
<td>Triggered just before executing user’s program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>execjob_attach</td>
<td>Triggered before running any execjob_prologue hooks, on each vnode where pbs_attach() runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>execjob_end</td>
<td>Triggered after job finishes or is killed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>execjob_preterm</td>
<td>Triggered just before job is killed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>execjob_epilogue</td>
<td>Triggered just after job runs successfully</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>exechost_periodic</td>
<td>Triggered at periodic interval on execution hosts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>exechost_startup</td>
<td>Triggered when MoM starts up or receives SIGHUP (Linux)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>***</td>
<td>Hook is not triggered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Attributes

### Hook Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Format</th>
<th>Val / Opt</th>
<th>Value/Option Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>fail_action</strong></td>
<td>String array</td>
<td>none</td>
<td>No action is taken.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>offline_vnodes</td>
<td>After unsuccessful hook execution, offline the vnodes managed by the MoM executing the hook. Only available for execjob_prologue, exechost_startup and execjob_begin hooks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>clear_vnodes_upon_recovery</td>
<td>After successful hook execution, clears vnodes previously offline via offline_vnodes fail action. Only available for exechost_startup hooks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scheduler_restart_cycle</td>
<td>After unsuccessful hook execution, restarts scheduling cycle. Only available for execjob_begin and execjob_prologue hooks.</td>
</tr>
<tr>
<td><strong>freq</strong></td>
<td>Integer</td>
<td></td>
<td>Number of seconds between triggers 120</td>
</tr>
<tr>
<td><strong>order</strong></td>
<td>Integer</td>
<td>Range: [-1000, 2000]</td>
<td>Indicates relative order of hook execution, for hooks sharing a trigger. Hooks with lower order values execute before those with higher values. Does not apply to periodic or exechost_periodic hooks.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>String</td>
<td>pbs</td>
<td>Hook is built in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>site</td>
<td>Hook is custom (site-defined)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pbsadmin</td>
<td>Hook runs as root</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pbsuser</td>
<td>Hook runs as owner of job</td>
</tr>
</tbody>
</table>
This chapter describes the formats used in PBS Professional.

### 7.1 List of Formats

#### Accounting Log Entry

```
logfile-date-time;record-type;id-string:message-text
```

Where:

- **logfile-date-time**
  - Date and time stamp in the format: `mm/dd/yyyy hh:mm:ss`
- **record-type**
  - A single character indicating the type of record
- **id-string**
  - The job or reservation identifier
- **message-text**
  - Format: blank-separated `keyword=value` fields.
  - Message text is ASCII text.
  - Content depends on the record type.

#### Attribute Name

- **PBS NAME**. Cannot be used for a vnode name.

#### Boolean

- Name of Boolean resource is a string.
- Values:
  - `TRUE, True, true, T, t, Y, y, 1`
  - `FALSE, False, false, F, f, N, n, 0`

#### Date

- `<Day of week> <Name of month> <Day of month> <HH:MM:SS> <YYYY>`
Datetime

A datetime is

$$\left[\left[CC\right]YY\left[MM\right]DD\right]hhmm[.SS]$$

where

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Century</td>
</tr>
<tr>
<td>YY</td>
<td>Year</td>
</tr>
<tr>
<td>MM</td>
<td>Month</td>
</tr>
<tr>
<td>DD</td>
<td>Day of month</td>
</tr>
<tr>
<td>hh</td>
<td>Hour</td>
</tr>
<tr>
<td>mm</td>
<td>Minute</td>
</tr>
<tr>
<td>SS</td>
<td>Second</td>
</tr>
</tbody>
</table>

When setting the value, each portion of the date defaults to the current date, as long as the next-smaller portion is in the future. For example, if today is the 3rd of the month and the specified day DD is the 5th, the month MM will be set to the current month.

If a specified portion has already passed, the next-larger portion will be set to one after the current date. For example, if the day DD is not specified, but the hour hh is specified to be 10:00 a.m. and the current time is 11:00 a.m., the day DD will be set to tomorrow.

Destination Identifier

String used to specify a particular destination. The identifier may be specified in one of three forms:

- `<queue name>`@<server name>
- `<queue name>`
- @<server name>

where `<queue name>` is an ASCII character string of up to 15 characters.

Valid characters are alphanumerics, the hyphen and the underscore. The string must begin with a letter.

Duration

A duration is a period of time, expressed either as

- An integer whose units are seconds
- ![hours:minutes:seconds.milliseconds]

in the form:

- ![HH:MM:SS.milliseconds]

Milliseconds are rounded to the nearest second.

Float

Floating point. Allowable values: `+[-] 0-9 [[0-9] ...][.] [[0-9] ...]`

Host Name

String of the form
name.domain

where domain is a hierarchical, dot-separated list of subdomains. Therefore, a host name cannot contain a dot, "." as a legal character other than as a subdomain separator.

The name must not contain the commercial at sign, ",@", as this is often used to separate a file from the host in a remote file name.

A host name cannot contain a colon, ":".

The maximum length of a host name supported by PBS is defined by PBS_MAXHOSTNAME, and is 255.

**Job Array Identifier**

Job array identifiers are a sequence number followed by square brackets:

<sequence number>[][.<server name>][@<server name>]

Example:

1234[

Note that some shells require that you enclose a job array ID in double quotes.

**Job Array Range**

<sequence number>[<first>-<last>][.<server name>][@<server name>]

first and last are the first and last indices of the subjobs.

**Job Identifier**

<sequence number>[.<server name>][@<server name>]

Format: [0-9]+[\[[0-9]+\]]..IP_HOSTNAME[.IP_DOMAINNAME]

Maximum value: 9999999; after this has been reached, job IDs start again at 0.

**Job Name, Job Array Name**

A job name or job array name can be at most 236 characters. It must consist of printable, non-whitespace characters. The first character must be alphabetic, numeric, plus sign ("+") dash or minus ("-"), or underscore ("_"). The name must not contain special characters.

Default: if a script is used to submit the job, the job’s name is the name of the script. If no script is used, the job’s name is "STDIN".
Limit Specification

\[ \text{limit specification}\! = \text{limit value}[, \text{limit specification}\! = \text{limit value}, ...] \]

where limit specification is:

Table 7-2: Limit Specification Syntax

<table>
<thead>
<tr>
<th>Limit Specification</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>o:PBS_ALL</td>
<td>Overall limit</td>
</tr>
<tr>
<td>u:PBS_GENERIC</td>
<td>Generic users</td>
</tr>
<tr>
<td>u:&lt;username&gt;</td>
<td>An individual user</td>
</tr>
<tr>
<td>g:PBS_GENERIC</td>
<td>Generic groups</td>
</tr>
<tr>
<td>g:&lt;group name&gt;</td>
<td>An individual group</td>
</tr>
<tr>
<td>p:PBS_GENERIC</td>
<td>Generic projects</td>
</tr>
<tr>
<td>p:&lt;project name&gt;</td>
<td>An individual project</td>
</tr>
</tbody>
</table>

- The limit specification can contain spaces anywhere except after the colon (";").
- If there are comma-separated limit specifications, the entire string must be enclosed in double quotes.
- A username, group name, or project name containing spaces must be enclosed in quotes.
- If a username, group name, or project name is quoted using double quotes, and the entire string requires quotes, the outer enclosing quotes must be single quotes. Similarly, if the inner quotes are single quotes, the outer quotes must be double quotes.
- PBS_ALL is a keyword which indicates that this limit applies to the usage total.
- PBS_GENERIC is a keyword which indicates that this limit applies to generic users, groups, or projects.
- When removing a limit, the limit value does not need to be specified.
- PBS_ALL and PBS_GENERIC are case-sensitive.

Format for setting a limit attribute:

```
set server <limit attribute> = "<limit specification>=<limit value>[, <limit specification>=<limit value>, ..."
set queue <queue name> <limit attribute> = "<limit specification>=<limit value>[, <limit specification>=<limit value>, ..."
```

For example, to set the max_queued limit on QueueA to 5 for total usage, and to limit user bill to 3:

```
Qmgr: s q QueueA max_queued = "[o:PBS_ALL=5], [u:bill =3]"
```

Examples of setting, adding, and removing:

```
Qmgr: set server max_run="[u:PBS GENERIC=2], [g:group1=10], [o:PBS_ALL = 100]"
Qmgr: set server max_run+="[u:user1=3], [g:PBS GENERIC=8]"
Qmgr: set server max_run-="[u:user2], [g:groupp3]"
Qmgr: set server max_run_res.ncpus="[u:PBS GENERIC=2], [g:group1=8], [o:PBS_ALL = 64]"
```

See "How to Set Limits at Server and Queues" on page 295 in the PBS Professional Administrator’s Guide.

logfile-date-time

Date and time stamp in the format:

`mm/dd/yyyy hh:mm:ss`
Long
Long integer. Allowable values: 0-9 [[0-9] ...], and + and -

pathname
All printable characters except for colon (";"), quotes(""""), and ampersand ("&")

PBS NAME
This is a generic term, used to describe various PBS entities. For example, attribute names are PBS NAMEs.
Must start with an alphabetic character, and may contain only the following: alpha-numeric, underscore ("_"), or dash ("-"").

PBS Password
The pbs_ds_password command generates passwords containing the following characters:
0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^&*()_+
When creating a password manually, do not use \ (backslash) or ' (backquote). This can prevent certain commands such as pbs_server, pbs_ds_password, and printjob from functioning properly, as they rely on connecting to the database.

Project Name
A project name can contain any characters except for the following: slash ("/"), left bracket ("[") , right bracket ("]"), double quote (""""), semicolon (";"), colon (:"), vertical bar ("|"), left angle bracket ("<"), right angle bracket (">"), plus ("+"), comma (","), question mark ("?"), and asterisk ("*").
Default value: "_pbs_project_default".

Queue Identifier
To specify a queue at the default server:
<queue name>
To specify all queues at a server:
@<server name>
To specify a queue at a specific server:
<queue name>@<server name>

Queue Name
PBS NAME

Reservation Name
Same as Job Name. See "Job Name, Job Array Name" on page 365.

Resource Name
PBS NAME
Resource names are case-insensitive.

Resource Value
• PBS NAME, or
• Anything inside double quotes
The format of each data type is defined for that data type. For example, float resources are defined above, in "Float" on page 364.
Size

Number of bytes or words. The size of a word is 64 bits.
Format: `<integer>[<suffix>]`
where `suffix` can be one of the following:

Table 7-3: Size in Bytes

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Meaning</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>b or w</td>
<td>Bytes or words</td>
<td>1</td>
</tr>
<tr>
<td>kb or kw</td>
<td>Kilobytes or kilowords</td>
<td>2 to the 10th, or 1024</td>
</tr>
<tr>
<td>mb or mw</td>
<td>Megabytes or megawords</td>
<td>2 to the 20th, or 1,048,576</td>
</tr>
<tr>
<td>gb or gw</td>
<td>Gigabytes or gigawords</td>
<td>2 to the 30th, or 1,073,741,824</td>
</tr>
<tr>
<td>tb or tw</td>
<td>Terabytes or terawords</td>
<td>2 to the 40th, or 1024 gigabytes</td>
</tr>
<tr>
<td>pb or pw</td>
<td>Petabytes or petawords</td>
<td>2 to the 50th, or 1,048,576 gigabytes</td>
</tr>
</tbody>
</table>

Default: `bytes`
Note that a scheduler rounds all resources of type `size` up to the nearest kb.

String

(Resource value)
Any character, including the space character.
Only one of the two types of quote characters, " or ' , may appear in any given value.
Values: `[ _a-zA-Z0-9][-_a-zA-Z0-9 ! # $ % ´ ( ) * + , - . / : ; < = > ? @ \ ^ _ ' { | } ~ ...]
String resource values are case-sensitive.

String Array
Comma-separated list of strings. Strings in `string_array` may not contain commas. No limit on length.

Subjob Identifier

Subjob identifiers are a sequence number followed by square brackets enclosing the subjob’s index:
`<sequence number>[<index>][.<server name>][@<server name>]`
Example:

1234[99]

Timestamp
Output format varies depending on context:
- Printed by `qstat` in human-readable `Date` format
- Output in hooks as seconds since epoch

Username
String up to 16 characters in length. PBS supports usernames containing any printable, non-whitespace character except the at sign (“@”). Your platform may place additional limitations on usernames.
Username, Windows

Must conform to the POSIX-1 standard for portability:

- The username must contain only alphanumeric characters, dot (.), underscore (_), and/or hyphen "-".
- The hyphen must not be the first letter of the username.
- If "@" appears in the username, it will assumed to be in the context of a Windows domain account: username@domainname.
- An exception to the above rule is the space character, which is allowed. If a space character appears in a username string, it will be displayed quoted and must be specified in a quoted manner.

Vnode Name

- For the natural vnode, the vnode name must conform to legal name for a host
- For other vnodes, the vnode name can be alphanumeric and any of these:
  - (dash)
  - (underscore)
  - (at sign)
  - [ (left bracket)
  - ] (right bracket)
  - # (hash)
  - ^ (caret)
  - / (slash)
  - \ (backslash)
  - . (period)
- Cannot be the same as an attribute name
This chapter lists and describes the states in PBS Professional.

## 8.1 Job States

Job states are abbreviated to one character.

### Table 8-1: Job States

<table>
<thead>
<tr>
<th>State</th>
<th>Numeric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>7</td>
<td>Job arrays only: job array is begun, meaning that at least one subjob has started</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>Job is exiting after having run</td>
</tr>
<tr>
<td>F</td>
<td>9</td>
<td>Job is finished. Job has completed execution, job failed during execution, or job was deleted.</td>
</tr>
<tr>
<td>H</td>
<td>2</td>
<td>Job is held. A job is put into a held state by the server or by a user or administrator. A job stays in a held state until it is released by a user or administrator.</td>
</tr>
<tr>
<td>M</td>
<td>8</td>
<td>Job was moved to another server</td>
</tr>
<tr>
<td>Q</td>
<td>1</td>
<td>Job is queued, eligible to run or be routed</td>
</tr>
<tr>
<td>R</td>
<td>4</td>
<td>Job is running</td>
</tr>
<tr>
<td>S</td>
<td>None; sub-state of Running</td>
<td>Job is suspended by scheduler. A job is put into the suspended state when a higher priority job needs the resources.</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>Job is in transition (being moved to a new location)</td>
</tr>
<tr>
<td>U</td>
<td>None; sub-state of Running</td>
<td>Job is suspended due to workstation becoming busy</td>
</tr>
<tr>
<td>W</td>
<td>3</td>
<td>Job is waiting for its requested execution time to be reached or job specified a stagein request which failed for some reason.</td>
</tr>
<tr>
<td>X</td>
<td>6</td>
<td>Subjobs only; subjob is finished (expired.)</td>
</tr>
</tbody>
</table>
## 8.1.1 Job Substates

Job substates are numeric:

<table>
<thead>
<tr>
<th>Substate Number</th>
<th>Substate Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Transit in, prior to waiting for commit</td>
</tr>
<tr>
<td>01</td>
<td>Transit in, waiting for commit</td>
</tr>
<tr>
<td>02</td>
<td>Transiting job outbound, not ready to commit</td>
</tr>
<tr>
<td>03</td>
<td>Transiting outbound, ready to commit</td>
</tr>
<tr>
<td>10</td>
<td>Job queued and ready for selection</td>
</tr>
<tr>
<td>11</td>
<td>Job queued, has files to stage in</td>
</tr>
<tr>
<td>13</td>
<td>Job waiting on sync start ready</td>
</tr>
<tr>
<td>14</td>
<td>Job staging in files before waiting</td>
</tr>
<tr>
<td>15</td>
<td>Job staging in files before running</td>
</tr>
<tr>
<td>16</td>
<td>Job stage in complete</td>
</tr>
<tr>
<td>20</td>
<td>Job held - user or operator</td>
</tr>
<tr>
<td>21</td>
<td>Job held waiting on sync regist</td>
</tr>
<tr>
<td>22</td>
<td>Job held - waiting on dependency</td>
</tr>
<tr>
<td>30</td>
<td>Job waiting until user-specified execution time</td>
</tr>
<tr>
<td>37</td>
<td>Job held - file stage in failed</td>
</tr>
<tr>
<td>41</td>
<td>Job sent to MoM to run</td>
</tr>
<tr>
<td>42</td>
<td>Running</td>
</tr>
<tr>
<td>43</td>
<td>Suspended by Operator or Manager</td>
</tr>
<tr>
<td>44</td>
<td>PBS no longer supports Globus. The Globus functionality has been <strong>removed</strong> from PBS. Job sent to run under Globus</td>
</tr>
<tr>
<td>45</td>
<td>Suspended by scheduler</td>
</tr>
<tr>
<td>50</td>
<td>Server received job obit</td>
</tr>
<tr>
<td>51</td>
<td>Staging out stdout/err and other files</td>
</tr>
<tr>
<td>52</td>
<td>Deleting stdout/err files and staged-in files</td>
</tr>
<tr>
<td>53</td>
<td>Mom releasing resources</td>
</tr>
<tr>
<td>54</td>
<td>Job is being aborted by server</td>
</tr>
<tr>
<td>56</td>
<td>(Set by MoM) Mother Superior telling sisters to kill everything</td>
</tr>
<tr>
<td>57</td>
<td>(Set by MoM) job epilogue running</td>
</tr>
<tr>
<td>58</td>
<td>(Set by MoM) job obit notice sent</td>
</tr>
<tr>
<td>59</td>
<td>Waiting for site &quot;job termination&quot; action script</td>
</tr>
</tbody>
</table>
8.2 Job Array States

Job array states map closely to job states except for the ‘B’ state. The ‘B’ state applies to job arrays and indicates that at least one subjob has left the queued state and is running or has run, but not all subjobs have run. Job arrays will never be in the ‘R’, ‘S’ or ‘U’ states.

### Table 8-3: Job Array States

<table>
<thead>
<tr>
<th>State</th>
<th>Numeric</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>7</td>
<td>The job array has started</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>All subjobs are finished and the server is cleaning up the job array</td>
</tr>
<tr>
<td>F</td>
<td>9</td>
<td>The job array is finished</td>
</tr>
<tr>
<td>H</td>
<td>2</td>
<td>The job array is held</td>
</tr>
<tr>
<td>Q</td>
<td>1</td>
<td>The job array is queued, or has been rerun</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>The job array is in transit between servers</td>
</tr>
<tr>
<td>W</td>
<td>3</td>
<td>The job array has a wait time in the future</td>
</tr>
</tbody>
</table>
8.3 Subjob States

Subjobs can be in one of six states, listed here.

<table>
<thead>
<tr>
<th>State</th>
<th>Numeric</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>5</td>
<td>Ending</td>
</tr>
<tr>
<td>F</td>
<td>9</td>
<td>Finished</td>
</tr>
<tr>
<td>Q</td>
<td>1</td>
<td>Queued</td>
</tr>
<tr>
<td>R</td>
<td>4</td>
<td>Running</td>
</tr>
<tr>
<td>S</td>
<td>None; sub-state of Running</td>
<td>Suspended</td>
</tr>
<tr>
<td>U</td>
<td>None; sub-state of Running</td>
<td>Suspended by keyboard activity</td>
</tr>
<tr>
<td>X</td>
<td>6</td>
<td>Expired or deleted; subjob has completed execution or been deleted</td>
</tr>
</tbody>
</table>

8.4 Server States

The state of the server is shown in the `server_state` server attribute. Possible values are shown in the following table:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hot_Start</td>
<td>The server has been started so that it will run first any jobs that were running when the server was shut down. Python type: <code>pbs.SV_STATE_HOT</code></td>
</tr>
<tr>
<td>Idle</td>
<td>The server is running. The scheduler is between scheduling cycles. Python type: <code>pbs.SV_STATE_IDLE</code></td>
</tr>
<tr>
<td>Scheduling</td>
<td>The server is running. The scheduler is in a scheduling cycle. Python type: <code>pbs.SV_STATE_ACTIVE</code></td>
</tr>
<tr>
<td>Terminating</td>
<td>The server is terminating. Python type: <code>pbs.SV_STATE_SHUTIMM</code> or <code>pbs.SV_STATE_SHUTSIG</code></td>
</tr>
<tr>
<td>Terminating_Delayed</td>
<td>The server is terminating in delayed mode. No new jobs will be run, and the server will shut down when the last running job finishes. Python type: <code>pbs.SV_STATE_SHUTDEL</code></td>
</tr>
</tbody>
</table>
8.5 Vnode States

If a vnode’s state attribute is unset, that is equivalent to the state being free. A vnode’s state is shown in its state attribute, which can take on zero or more of the values listed here. Some vnode state values can be set simultaneously. Values are:

<table>
<thead>
<tr>
<th>State</th>
<th>Set By</th>
<th>Description</th>
<th>Can Combine With these States</th>
</tr>
</thead>
<tbody>
<tr>
<td>busy</td>
<td>Server</td>
<td>Node is up and has load average greater than max_load, or is showing keyboard or mouse activity. When the loadave is above max_load, that node is marked busy. A scheduler won’t place jobs on a node marked busy. When the loadave drops below ideal_load, or when the mouse and keyboard have not shown any activity for a specified amount of time, the busy mark is removed. Consult your OS documentation to determine values that make sense.</td>
<td>offline maintenance</td>
</tr>
<tr>
<td>down</td>
<td>Server</td>
<td>Node is not usable. Existing communication lost between server and MoM.</td>
<td>maintenance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cannot be set with free</td>
</tr>
<tr>
<td>free</td>
<td>Server</td>
<td>Node is up and has available CPU(s). Server will mark a vnode “free” on first successful ping after vnode was “down”.</td>
<td>None</td>
</tr>
<tr>
<td>job-busy</td>
<td>Server</td>
<td>Node is up and all CPUs are allocated to jobs.</td>
<td>offline</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>resv-exclusive</td>
</tr>
<tr>
<td>job-exclusive</td>
<td>Server</td>
<td>Node is up and has been allocated exclusively to a single job.</td>
<td>offline</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>resv-exclusive</td>
</tr>
<tr>
<td>maintenance</td>
<td>Server</td>
<td>A vnode enters the maintenance state when any of its jobs is suspended with the admin-suspend signal. Other jobs running on this vnode continue to run; each job must be admin-suspended. The vnode leaves the maintenance state when the last job is resumed with the admin-resume signal. A scheduler does not start or resume jobs on a node in the maintenance state. Any reservations on vnodes in the maintenance state are marked degraded. PBS searches for alternate vnodes for those reservations.</td>
<td>down</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>offline</td>
</tr>
<tr>
<td>offline</td>
<td>Manager</td>
<td>Node is not usable. Jobs running on this vnode will continue to run. Used by Manager/Operator to mark a vnode not to be used for jobs.</td>
<td>busy</td>
</tr>
<tr>
<td></td>
<td>Operator</td>
<td></td>
<td>job-busy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>job-exclusive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>resv-exclusive</td>
</tr>
<tr>
<td>powered-off</td>
<td></td>
<td>Indicates that this vnode was powered off by PBS via power provisioning. This tells the scheduler that it can schedule jobs on this vnode; in that case PBS powers the vnode back up.</td>
<td></td>
</tr>
</tbody>
</table>
## Table 8-6: Vnode States

<table>
<thead>
<tr>
<th>State</th>
<th>Set By</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>powering-down</td>
<td></td>
<td>Indicates that this vnode is in the process of being powered down by PBS via power provisioning.</td>
</tr>
<tr>
<td>powering-on</td>
<td></td>
<td>Indicates that this vnode is in the process of being powered up by PBS via power provisioning.</td>
</tr>
<tr>
<td>provisioning</td>
<td>Server</td>
<td>A vnode is in the provisioning state while it is in the process of being provisioned. No jobs are run on vnodes in the provisioning state.</td>
</tr>
<tr>
<td>resv-exclusive</td>
<td>Server</td>
<td>Reservation has requested exclusive use of vnode, and reservation is running.</td>
</tr>
<tr>
<td>sleep</td>
<td>Server</td>
<td>Indicates that this vnode was ramped down or powered off via PBS power management. This tells the scheduler that it can schedule jobs on this vnode; in that case PBS powers the vnode back up.</td>
</tr>
</tbody>
</table>
| stale             | Server | MoM managing vnode is not reporting any information about this vnode, but was reporting it previously. Server can still communicate with MoM. A vnode becomes stale when:  
1. A vnode is defined in the server  
2. MoM starts or restarts and reports a set of vnodes according to her configuration  
3. A vnode which existed in the server earlier is not in the set being reported now by MoM  
4. That vnode is marked "stale"                                                                                                                                                     |
| state-unknown,    | Server | Node is not usable. Since server’s latest start, no communication with this vnode. May be network or hardware problem, or no MoM on vnode.                                                                  |
| down              |        |                                                                                                                                                                                                            |
| unresolvable      | Server | Server cannot resolve name of vnode                                                                                                                                                                         |
| wait-provisioning | Server | There is a limit on the maximum number of vnodes that can be in the provisioning state. This limit is specified in the server’s max_concurrent_provision attribute. If a vnode is to be provisioned, but cannot because the number of concurrently provisioning vnodes has reached the specified maximum, the vnode goes into the wait-provisioning state. No jobs are run on vnodes in the wait-provisioning state. |

### Notes:
- Cannot be set with any other states
- Cannot be set with job-exclusive, offline
- Cannot be set with free
8.6 Reservation States

The following table shows the list of possible states for a reservation. The states that you will usually see are CO, UN, BD, and RN, although a reservation usually remains unconfirmed for too short a time to see that state.

Table 8-7: Reservation States

<table>
<thead>
<tr>
<th>Code</th>
<th>Numeric</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>11</td>
<td>RESV_BEING_ALTERED</td>
<td>Transitory state; reservation is being altered</td>
</tr>
<tr>
<td>BD</td>
<td>7</td>
<td>RESV_BEING_DELETED</td>
<td>Transitory state; reservation is being deleted</td>
</tr>
<tr>
<td>CO</td>
<td>2</td>
<td>RESV_CONFIRMED</td>
<td>Reservation confirmed</td>
</tr>
<tr>
<td>DG</td>
<td>10</td>
<td>RESV_DEGRADED</td>
<td>Vnode(s) allocated to reservation unavailable</td>
</tr>
<tr>
<td>DE</td>
<td>8</td>
<td>RESV_DELETED</td>
<td>Transitory state; reservation has been deleted</td>
</tr>
<tr>
<td>DJ</td>
<td>9</td>
<td>RESV_DELETING_JOBS</td>
<td>Jobs remaining after reservation’s end time being deleted</td>
</tr>
<tr>
<td>FN</td>
<td>6</td>
<td>RESV_FINISHED</td>
<td>Transitory state; reservation’s end time has arrived and reservation will be deleted</td>
</tr>
<tr>
<td>NO</td>
<td>0</td>
<td>RESV_NONE</td>
<td>No reservation yet</td>
</tr>
<tr>
<td>RN</td>
<td>5</td>
<td>RESV_RUNNING</td>
<td>Time period from reservation’s start time to end time is being traversed</td>
</tr>
<tr>
<td>TR</td>
<td>4</td>
<td>RESV_TIME_TO_RUN</td>
<td>Transitory state; reservation’s start time has arrived</td>
</tr>
<tr>
<td>UN</td>
<td>1</td>
<td>RESV_UNCONFIRMED</td>
<td>Reservation not confirmed</td>
</tr>
<tr>
<td>WT</td>
<td>3</td>
<td>RESV_WAIT</td>
<td>Unused</td>
</tr>
</tbody>
</table>

8.6.1 Degraded Reservation Substates

The following table shows states and substates for degraded reservations:

Table 8-8: Degraded Reservation States and Substates

<table>
<thead>
<tr>
<th>Occurrence Type</th>
<th>State or Substate</th>
<th>Reservation Time in Future</th>
<th>Reservation Time Is Now</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advance Reservation</td>
<td>State</td>
<td>RESV_DEGRADED</td>
<td>RESV_RUNNING</td>
</tr>
<tr>
<td></td>
<td>Substate</td>
<td>RESV_DEGRADED</td>
<td>RESV_DEGRADED</td>
</tr>
<tr>
<td>Soonest Occurrence</td>
<td>State</td>
<td>RESV_DEGRADED</td>
<td>RESV_RUNNING</td>
</tr>
<tr>
<td></td>
<td>Substate</td>
<td>RESV_DEGRADED</td>
<td>RESV_DEGRADED</td>
</tr>
<tr>
<td>Non-soonest Occurrence Only</td>
<td>State</td>
<td>RESV_CONFIRMED</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Substate</td>
<td>RESV_DEGRADED</td>
<td>N/A</td>
</tr>
</tbody>
</table>
This chapter shows some configuration-specific scenarios which will hopefully clarify any configuration questions. Several configuration models are discussed, followed by several complex examples of specific features.

- Single Vnode System
- Single Vnode System with Separate PBS server
- Multi-vnode complex
- Complex Multi-level Route Queues (including group ACLs)
- Multiple User ACLs

For each of these possible configuration models, the following information is provided:

- General description for the configuration model
- Type of system for which the model is well suited
- Contents of server nodes file
- Any required server configuration
- Any required MoM configuration
- Any required scheduler configuration

### 9.1 Single Vnode System

Running PBS on a single vnode/host as a standalone system is the least complex configuration. This model is most applicable to sites who have a single large server system. In this model, all PBS components run on the same host, which is the same host on which jobs will be executed. The following illustration shows how communication works when PBS is on a single host in TPP mode. For more on TPP mode, see Chapter 4, "Communication", on page 47.

![Figure 9-1: PBS daemons on a single execution host](image-url)
For this example, let's assume we have a 32-CPU server machine named “mars”. We want users to log into mars and jobs will be run via PBS on mars.

In this configuration, the server's default `nodes` file (which should contain the name of the host on which the server was installed) is sufficient. Our example `nodes` file would contain only one entry: `mars`

The default MoM and scheduler configuration files, as well as the default queue/Server limits are also sufficient in order to run jobs. No changes are required from the default configuration, however, you may wish to customize PBS to your site.

### 9.2 Separate Server and Execution Host

A variation on the model presented above would be to provide a “front-end” system that ran the PBS server, scheduler, and communication daemons, and from which users submitted their jobs. Only the MoM would run on our execution server, mars. This model is recommended when the user load would otherwise interfere with the computational load on the server. The following illustration shows how communication works when the PBS server and scheduler are on a front-end system and MoM is on a separate host, in TPP mode. For more on TPP mode, see Chapter 4, "Communication", on page 47.

![PBS daemons on single execution system with front end](image)

In this case, the PBS `server_priv/nodes` file would contain the name of our execution server mars, but this may not be what was written to the file during installation, depending on which options were selected. It is possible the hostname of the machine on which the server was installed was added to the file, in which case you would need to use `qmgr(1B)` to manipulate the contents to contain one vnode: `mars`. If the default scheduling policy, based on available CPUs and memory, meets your requirements, no changes are required in either the MoM or scheduler configuration files.

However, if you wish the execution host (mars) to be scheduled based on load average, the following changes are needed. Edit MoM’s `mom_priv/config` file so that it contains the target and maximum load averages:

```
$ideal_load 30
$max_load 32
```

In the partition scheduler’s `sched_priv/sched_config` file, the following options need to be set:

```
load_balancing: True all
```
## 9.3 Multiple Execution Hosts

The multi-vnode complex model is a very common configuration for PBS. In this model, there is typically a front-end system as we saw in the previous example, with a number of back-end execution hosts. The PBS server, scheduler, and communication daemons are typically run on the front-end system, and a MoM is run on each of the execution hosts, as shown in the diagram to the right.

In this model, the server’s `nodes` file will need to contain the list of all the vnodes in the complex.

The MoM `config` file on each vnode will need two static resources added, to specify the target load for each vnode. If we assume each of the vnodes in our “planets” cluster is a 32-processor system, the following example shows what might be desirable ideal and maximum load values to add to the MoM `config` files:

```
$ideal_load 30
$max_load 32
```

Furthermore, suppose we want the partition scheduler to load balance the workload across the available vnodes, making sure not to run two jobs in a row on the same vnode. We accomplish this by editing the scheduler configuration file and enabling load balancing:

```
load_balancing: True all
smp_cluster_dist: round_robin
```

The following diagram illustrates this for an eight-host complex in TPP mode.
Chapter 9

Example Configurations

This diagram illustrates a multi-vnode complex TPP configuration wherein the server and scheduler daemons communicate with the MoMs on the execution hosts via the communication daemon. Jobs are submitted to the server, scheduled for execution by the partition scheduler, and then transferred to a MoM when it’s time to be run. MoM periodically sends status information back to the server, and answers resource requests from the scheduler.

9.4 Complex Multi-level Route Queues

There are times when a site may wish to create a series of route queues in order to filter jobs, based on specific resources, or possibly to different destinations. For this example, consider a site that has two large server systems, and a Linux cluster. The Administrator wants to configure route queues such that everyone submits jobs to a single queue, but the jobs get routed based on (1) requested architecture and (2) individual group IDs. In other words, users request the architecture they want, and PBS finds the right queue for them. Only groups “math”, “chemistry”, and “physics” are permitted to use either server systems; while anyone can use the cluster. Lastly, the jobs coming into the cluster should be divided into three separate queues for long, short, and normal jobs. But the “long” queue was created for the astronomy department, so only members of that group should be permitted into that queue. Given these requirements, let’s look at how we would set up such a collection of route queues. (Note that this is only one way to accomplish this task. There are various other ways too.)

First we create a queue to which everyone will submit their jobs. Let’s call it “submit”. It will need to be a route queue with three destinations, as shown:

```
Qmgr: create queue submit
Qmgr: set queue submit queue_type = Route
Qmgr: set queue submit route_destinations = server_1
Qmgr: set queue submit route_destinations += server_2
Qmgr: set queue submit route_destinations += cluster
Qmgr: set queue submit enabled = True
Qmgr: set queue submit started = True
```

Now we need to create the destination queues. (Notice in the above example, we have already decided what to call the three destinations: server_1, server_2, cluster.) First we create the server_1 queue, complete with a group ACL, and a specific architecture limit.

```
Qmgr: create queue server_1
Qmgr: set queue server_1 queue_type = Execution
Qmgr: set queue server_1 from_route_only = True
Qmgr: set queue server_1 resources_max.arch = linux
Qmgr: set queue server_1 resources_min.arch = linux
Qmgr: set queue server_1 acl_group_enable = True
Qmgr: set queue server_1 acl_groups = math
Qmgr: set queue server_1 acl_groups += chemistry
Qmgr: set queue server_1 acl_groups += physics
Qmgr: set queue server_1 enabled = True
Qmgr: set queue server_1 started = True
```
Next we create the queues for `server_2` and `cluster`. Note that the `server_2` queue is very similar to the `server_1` queue, only the architecture differs. Also notice that the `cluster` queue is another route queue, with multiple destinations.

```
Qmgr: create queue server_2
Qmgr: set queue server_2 queue_type = Execution
Qmgr: set queue server_2 from_route_only = True
Qmgr: set queue server_2 resources_max.arch = sv2
Qmgr: set queue server_2 resources_min.arch = sv2
Qmgr: set queue server_2 acl_group_enable = True
Qmgr: set queue server_2 acl_groups = math
Qmgr: set queue server_2 acl_groups += chemistry
Qmgr: set queue server_2 acl_groups += physics
Qmgr: set queue server_2 enabled = True
Qmgr: set queue server_2 started = True
Qmgr: create queue cluster
Qmgr: set queue cluster queue_type = Route
Qmgr: set queue cluster from_route_only = True
Qmgr: set queue cluster resources_max.arch = linux
Qmgr: set queue cluster resources_min.arch = linux
Qmgr: set queue cluster route_destinations = long
Qmgr: set queue cluster route_destinations += short
Qmgr: set queue cluster route_destinations += medium
Qmgr: set queue cluster enabled = True
Qmgr: set queue cluster started = True
```

In the cluster queue above, you will notice the particular order of the three destination queues (`long`, `short`, `medium`). PBS will attempt to route a job into the destination queues in the order specified. Thus, we want PBS to first try the `long` queue (which will have an ACL on it), then the `short` queue (with its short time limits). Thus any jobs that had not been routed into any other queues (server or cluster) will end up in the `medium` cluster queue. Now to create the remaining queues.

```
Qmgr: create queue long
Qmgr: set queue long queue_type = Execution
Qmgr: set queue long from_route_only = True
Qmgr: set queue long resources_max.cput = 20:00:00
Qmgr: set queue long resources_max.walltime = 20:00:00
Qmgr: set queue long resources_min.cput = 02:00:00
Qmgr: set queue long resources_min.walltime = 03:00:00
Qmgr: set queue long acl_group_enable = True
Qmgr: set queue long acl_groups = astrology
Qmgr: set queue long enabled = True
Qmgr: set queue long started = True
```
Example Configurations

Qmgr: create queue short
Qmgr: set queue short queue_type = Execution
Qmgr: set queue short from_route_only = True
Qmgr: set queue short resources_max.cput = 01:00:00
Qmgr: set queue short resources_max.walltime = 01:00:00
Qmgr: set queue short enabled = True
Qmgr: set queue short started = True
Qmgr: create queue medium
Qmgr: set queue medium queue_type = Execution
Qmgr: set queue medium from_route_only = True
Qmgr: set queue medium enabled = True
Qmgr: set queue medium started = True
Qmgr: set server default_queue = submit

Notice that the long and short queues have time limits specified. This will ensure that jobs of certain sizes will enter (or be prevented from entering) these queues. The last queue, medium, has no limits, thus it will be able to accept any job that is not routed into any other queue.

Lastly, note the last line in the example above, which specified that the default queue is the new submit queue. This way users will simply submit their jobs with the resource and architecture requests, without specifying a queue, and PBS will route the job into the correct location. For example, if a user submitted a job with the following syntax, the job would be routed into the server_2 queue:

```
qsub -l select=arch=sv2:ncpus=4 testjob
```

9.5 External Software License Management

PBS Professional can be configured to schedule jobs based on externally-controlled licensed software. A detailed example is provided in "Example of Floating, Externally-managed License with Features" on page 271 in the PBS Professional Administrator’s Guide.
A site may have a need to restrict individual users to particular queues. In the previous example we set up queues with group-based ACLs, in this example we show user-based ACLs. Say a site has two different groups of users, and wants to limit them to two separate queues (perhaps with different resource limits). The following example illustrates this.

```plaintext
Qmgr: create queue structure
Qmgr: set queue structure queue_type = Execution
Qmgr: set queue structure acl_user_enable = True
Qmgr: set queue structure acl_users = curly
Qmgr: set queue structure acl_users += jerry
Qmgr: set queue structure acl_users += larry
Qmgr: set queue structure acl_users += moe
Qmgr: set queue structure acl_users += tom
Qmgr: set queue structure resources_max.nodes = 48
Qmgr: set queue structure enabled = True
Qmgr: set queue structure started = True

Qmgr: create queue engine
Qmgr: set queue engine queue_type = Execution
Qmgr: set queue engine acl_user_enable = True
Qmgr: set queue engine acl_users = bill
Qmgr: set queue engine acl_users += bobby
Qmgr: set queue engine acl_users += chris
Qmgr: set queue engine acl_users += jim
Qmgr: set queue engine acl_users += mike
Qmgr: set queue engine acl_users += rob
Qmgr: set queue engine acl_users += scott
Qmgr: set queue engine resources_max.nodes = 12
Qmgr: set queue engine resources_max.walltime=04:00:00
Qmgr: set queue engine enabled = True
Qmgr: set queue engine started = True
```
10

Run Limit Error Messages

This chapter lists the error messages generated when limits are exceeded. See "Managing Resource Usage By Users, Groups, and Projects, at Server & Queues" on page 286 in the PBS Professional Administrator’s Guide.

10.1 Run Limit Error Messages

When a job would exceed a limit by running, the job’s comment field is set to one of the following messages. The following table shows the limit attribute, where the limit is applied, to whom the limit is applied, and the message.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Where Applied</th>
<th>To What Applied</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_run</td>
<td>queue</td>
<td>o: PBS_ALL</td>
<td>Not Running: Queue &lt;queue name&gt; job limit has been reached.</td>
</tr>
<tr>
<td>max_run</td>
<td>server</td>
<td>o: PBS_ALL</td>
<td>Not Running: Server job limit has been reached.</td>
</tr>
<tr>
<td>max_run</td>
<td>server</td>
<td>p: PBS_GENERIC</td>
<td>Not Running: Project has reached server running limit.</td>
</tr>
<tr>
<td>max_run</td>
<td>queue</td>
<td>p: PBS_GENERIC</td>
<td>Not Running: Project has reached queue &lt;queue name&gt;’s running limit.</td>
</tr>
<tr>
<td>max_run</td>
<td>server</td>
<td>p: &lt;project name&gt;</td>
<td>Not Running: Server job limit reached for project &lt;project name&gt;</td>
</tr>
<tr>
<td>max_run</td>
<td>queue</td>
<td>p: &lt;project name&gt;</td>
<td>Not Running: Queue &lt;queue name&gt; job limit reached for project &lt;project name&gt;</td>
</tr>
<tr>
<td>max_run</td>
<td>queue</td>
<td>g: PBS_GENERIC</td>
<td>Not Running: Group has reached queue &lt;queue name&gt; running limit.</td>
</tr>
<tr>
<td>max_run</td>
<td>server</td>
<td>g: PBS_GENERIC</td>
<td>Not Running: Group has reached server running limit.</td>
</tr>
<tr>
<td>max_run</td>
<td>queue</td>
<td>u: PBS_GENERIC</td>
<td>Not Running: User has reached queue &lt;queue name&gt; running job limit.</td>
</tr>
<tr>
<td>max_run</td>
<td>server</td>
<td>u: PBS GENERIC</td>
<td>Not Running: User has reached server running job limit.</td>
</tr>
<tr>
<td>max_run</td>
<td>queue</td>
<td>g: &lt;group name&gt;</td>
<td>Queue &lt;queue name&gt; job limit reached for group &lt;G&gt;</td>
</tr>
<tr>
<td>max_run</td>
<td>server</td>
<td>g: &lt;group name&gt;</td>
<td>Server job limit reached for group &lt;G&gt;</td>
</tr>
<tr>
<td>max_run</td>
<td>queue</td>
<td>u: &lt;user name&gt;</td>
<td>Queue &lt;queue name&gt; job limit reached for user &lt;U&gt;</td>
</tr>
<tr>
<td>max_run</td>
<td>server</td>
<td>u: &lt;user name&gt;</td>
<td>Server job limit reached for user &lt;U&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>queue</td>
<td>o: PBS_ALL</td>
<td>Queue &lt;queue name&gt; job limit reached on resource &lt;resource name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>server</td>
<td>o: PBS_ALL</td>
<td>Server job limit reached on resource &lt;resource name&gt;</td>
</tr>
</tbody>
</table>
### Table 10-1: Job Run Limit Error Messages

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Where Applied</th>
<th>To What Applied</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_run_res</td>
<td>queue</td>
<td>p:PBS_GENERIC</td>
<td>Not Running: Queue &lt;queue name&gt; per-project limit reached on resource &lt;resource name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>server</td>
<td>p:PBS_GENERIC</td>
<td>Not Running: Server per-project limit reached on resource &lt;resource name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>server</td>
<td>p:&lt;project name&gt;</td>
<td>Not Running: would exceed project &lt;project_name&gt;'s limit on resource &lt;resource name&gt; in complex</td>
</tr>
<tr>
<td>max_run_res</td>
<td>queue</td>
<td>p:&lt;project name&gt;</td>
<td>Not Running: would exceed project &lt;project_name&gt;'s limit on resource &lt;resource name&gt; in queue &lt;queue-name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>queue</td>
<td>g: PBS_GENERIC</td>
<td>Queue &lt;queue name&gt; per-group limit reached on resource &lt;resource name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>server</td>
<td>g: PBS_GENERIC</td>
<td>Server per-group limit reached on resource &lt;resource name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>queue</td>
<td>u: PBS_GENERIC</td>
<td>Queue &lt;queue name&gt; per-user limit reached on resource &lt;resource name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>server</td>
<td>u: PBS_GENERIC</td>
<td>Server per-user limit reached on resource &lt;resource name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>queue</td>
<td>g:&lt;group name&gt;</td>
<td>would exceed group &lt;G&gt;'s limit on resource &lt;resource name&gt; in queue &lt;queue name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>server</td>
<td>g:&lt;group name&gt;</td>
<td>would exceed group &lt;G&gt;'s limit on resource &lt;resource name&gt; in complex</td>
</tr>
<tr>
<td>max_run_res</td>
<td>queue</td>
<td>u:&lt;user name&gt;</td>
<td>would exceed user &lt;U&gt;'s limit on resource &lt;resource name&gt; in queue &lt;queue name&gt;</td>
</tr>
<tr>
<td>max_run_res</td>
<td>server</td>
<td>u:&lt;user name&gt;</td>
<td>would exceed user &lt;U&gt;'s limit on resource &lt;resource name&gt; in complex</td>
</tr>
</tbody>
</table>
The following table lists all the PBS error codes, their textual names, and a description of each.

Table 11-1: Error Codes

<table>
<thead>
<tr>
<th>Error Name</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBSE_NONE</td>
<td>0</td>
<td>No error</td>
</tr>
<tr>
<td>PBSE_UNKJOBID</td>
<td>15001</td>
<td>Unknown Job Identifier</td>
</tr>
<tr>
<td>PBSE_NOATTR</td>
<td>15002</td>
<td>Undefined Attribute</td>
</tr>
<tr>
<td>PBSE_ATTRRO</td>
<td>15003</td>
<td>Attempt to set READ ONLY attribute</td>
</tr>
<tr>
<td>PBSE_IVALREQ</td>
<td>15004</td>
<td>Invalid request</td>
</tr>
<tr>
<td>PBSE_UNKREQ</td>
<td>15005</td>
<td>Unknown batch request</td>
</tr>
<tr>
<td>PBSE_TOOMANY</td>
<td>15006</td>
<td>Too many submit retries</td>
</tr>
<tr>
<td>PBSE_PERM</td>
<td>15007</td>
<td>No permission</td>
</tr>
<tr>
<td>PBSE_BADHOST</td>
<td>15008</td>
<td>Access from host not allowed</td>
</tr>
<tr>
<td>PBSE_JOBEXIST</td>
<td>15009</td>
<td>Job already exists</td>
</tr>
<tr>
<td>PBSE_SYSTEM</td>
<td>15010</td>
<td>System error occurred</td>
</tr>
<tr>
<td>PBSE_INTERNAL</td>
<td>15011</td>
<td>Internal server error occurred</td>
</tr>
<tr>
<td>PBSE_REGROUTE</td>
<td>15012</td>
<td>Parent job of dependent in route queue</td>
</tr>
<tr>
<td>PBSE_UNKSIG</td>
<td>15013</td>
<td>Unknown signal name</td>
</tr>
<tr>
<td>PBSE_BADATVAL</td>
<td>15014</td>
<td>Bad attribute value</td>
</tr>
<tr>
<td>PBSE_MODATRRUN</td>
<td>15015</td>
<td>Cannot modify attribute in run state</td>
</tr>
<tr>
<td>PBSE_BADSTATE</td>
<td>15016</td>
<td>Request invalid for job state</td>
</tr>
<tr>
<td>PBSE_UNKQUE</td>
<td>15018</td>
<td>Unknown queue name</td>
</tr>
<tr>
<td>PBSE_BADCRED</td>
<td>15019</td>
<td>Invalid Credential in request</td>
</tr>
<tr>
<td>PBSE_EXPIRED</td>
<td>15020</td>
<td>Expired Credential in request</td>
</tr>
<tr>
<td>PBSE_QUNOENB</td>
<td>15021</td>
<td>Queue not enabled</td>
</tr>
<tr>
<td>PBSE_QACCESS</td>
<td>15022</td>
<td>No access permission for queue</td>
</tr>
</tbody>
</table>
### Table 11-1: Error Codes

<table>
<thead>
<tr>
<th>Error Name</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBSE_BADUSER</td>
<td>15023</td>
<td>Missing userID, username, or GID. Returned under following conditions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. User does not have a password entry (getpwnam() returns null).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. User’s UID is zero and root isn’t allowed to run jobs (acl_roots).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. PBS_O_HOST is not set in the job.</td>
</tr>
<tr>
<td>PBSE_HOPCOUNT</td>
<td>15024</td>
<td>Max hop count exceeded</td>
</tr>
<tr>
<td>PBSE_QUEEXIST</td>
<td>15025</td>
<td>Queue already exists</td>
</tr>
<tr>
<td>PBSE_ATTRTYPE</td>
<td>15026</td>
<td>Incompatible queue attribute type</td>
</tr>
<tr>
<td>PBSE_OBJBUSY</td>
<td>15027</td>
<td>Object Busy</td>
</tr>
<tr>
<td>PBSE_QUENBIG</td>
<td>15028</td>
<td>Queue name too long</td>
</tr>
<tr>
<td>PBSE_NOSUP</td>
<td>15029</td>
<td>Feature/function not supported</td>
</tr>
<tr>
<td>PBSE_QUENOEN</td>
<td>15030</td>
<td>Can’t enable queue, lacking definition</td>
</tr>
<tr>
<td>PBSE_PROTOCOL</td>
<td>15031</td>
<td>Protocol (ASN.1) error. Message is distorted or truncated.</td>
</tr>
<tr>
<td>PBSE_BADATLST</td>
<td>15032</td>
<td>Bad attribute list structure</td>
</tr>
<tr>
<td>PBSE_NOCONNECTS</td>
<td>15033</td>
<td>No free connections</td>
</tr>
<tr>
<td>PBSE_NOSERVER</td>
<td>15034</td>
<td>No server to connect to</td>
</tr>
<tr>
<td>PBSE_UNKRESC</td>
<td>15035</td>
<td>Unknown resource</td>
</tr>
<tr>
<td>PBSE_EXCQRESC</td>
<td>15036</td>
<td>Job exceeds Queue resource limits</td>
</tr>
<tr>
<td>PBSE_QUENODFLT</td>
<td>15037</td>
<td>No Default Queue Defined</td>
</tr>
<tr>
<td>PBSE_NORERUN</td>
<td>15038</td>
<td>Job Not Rerunnable</td>
</tr>
<tr>
<td>PBSE_ROUTEREJ</td>
<td>15039</td>
<td>Route rejected by all destinations</td>
</tr>
<tr>
<td>PBSE_ROUTEEXPD</td>
<td>15040</td>
<td>Time in Route Queue Expired</td>
</tr>
<tr>
<td>PBSE_MOMREJECT</td>
<td>15041</td>
<td>Request to MoM failed</td>
</tr>
<tr>
<td>PBSE_BADSCRIPT</td>
<td>15042</td>
<td>(qsub) Cannot access script file</td>
</tr>
<tr>
<td>PBSE_STAGEIN</td>
<td>15043</td>
<td>Stage In of files failed</td>
</tr>
<tr>
<td>PBSE_RESCUNAV</td>
<td>15044</td>
<td>Resources temporarily unavailable</td>
</tr>
<tr>
<td>PBSE_BADGRP</td>
<td>15045</td>
<td>Bad Group specified</td>
</tr>
<tr>
<td>PBSE_MAXQUED</td>
<td>15046</td>
<td>Max number of jobs in queue</td>
</tr>
<tr>
<td>PBSE_CKPBSY</td>
<td>15047</td>
<td>Checkpoint Busy, may be retries</td>
</tr>
<tr>
<td>PBSE_EXLIMIT</td>
<td>15048</td>
<td>Limit exceeds allowable</td>
</tr>
<tr>
<td>PBSE_BADACCT</td>
<td>15049</td>
<td>Bad Account attribute value</td>
</tr>
<tr>
<td>PBSE_ALRDYEXIT</td>
<td>15050</td>
<td>Job already in exit state</td>
</tr>
<tr>
<td>Error Name</td>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>PBSE_NOCOPYFILE</td>
<td>15051</td>
<td>Job files not copied</td>
</tr>
<tr>
<td>PBSE_CLEANEDOUT</td>
<td>15052</td>
<td>Unknown job id after clean init</td>
</tr>
<tr>
<td>PBSE_NOSYNCHMSTR</td>
<td>15053</td>
<td>No Master in Sync Set</td>
</tr>
<tr>
<td>PBSE_BADDEPEND</td>
<td>15054</td>
<td>Invalid dependency</td>
</tr>
<tr>
<td>PBSE_DUPLIST</td>
<td>15055</td>
<td>Duplicate entry in List</td>
</tr>
<tr>
<td>PBSE_DISPROTO</td>
<td>15056</td>
<td>Bad DIS based Request Protocol</td>
</tr>
<tr>
<td>PBSE_EXECTHERE (Obsolete)</td>
<td>15057</td>
<td>Cannot execute there (Obsolete; no longer used.)</td>
</tr>
<tr>
<td>PBSE_SISREJECT</td>
<td>15058</td>
<td>Sister rejected</td>
</tr>
<tr>
<td>PBSE_SISCOMM</td>
<td>15059</td>
<td>Sister could not communicate</td>
</tr>
<tr>
<td>PBSE_SVRDOWN</td>
<td>15060</td>
<td>Request rejected -server shutting down</td>
</tr>
<tr>
<td>PBSE_CKPSHORT</td>
<td>15061</td>
<td>Not all tasks could checkpoint</td>
</tr>
<tr>
<td>PBSE_UNKNODE</td>
<td>15062</td>
<td>Named vnode is not in the list</td>
</tr>
<tr>
<td>PBSE_UNKNODEATR</td>
<td>15063</td>
<td>Vnode attribute not recognized</td>
</tr>
<tr>
<td>PBSE_NONODES</td>
<td>15064</td>
<td>Server has no vnode list</td>
</tr>
<tr>
<td>PBSE_NODENBIG</td>
<td>15065</td>
<td>Node name is too big</td>
</tr>
<tr>
<td>PBSE_NODEEXIST</td>
<td>15066</td>
<td>Node name already exists</td>
</tr>
<tr>
<td>PBSE_BADNDATEVAL</td>
<td>15067</td>
<td>Bad vnode attribute value</td>
</tr>
<tr>
<td>PBSE_MUTUALEX</td>
<td>15068</td>
<td>State values are mutually exclusive</td>
</tr>
<tr>
<td>PBSE_GMODERR</td>
<td>15069</td>
<td>Error(s) during global mod of vnodes</td>
</tr>
<tr>
<td>PBSE_NORELYMOM</td>
<td>15070</td>
<td>Could not contact MoM</td>
</tr>
<tr>
<td>Reserved</td>
<td>15076</td>
<td>Not used.</td>
</tr>
<tr>
<td>PBSE_TOOLATE</td>
<td>15077</td>
<td>Reservation submitted with a start time that has already passed</td>
</tr>
<tr>
<td>PBSE_genBatchReq</td>
<td>15082</td>
<td>Batch request generation failed</td>
</tr>
<tr>
<td>PBSE_mgrBatchReq</td>
<td>15083</td>
<td><code>qmgr</code> batch request failed</td>
</tr>
<tr>
<td>PBSE_UNKRESVID</td>
<td>15084</td>
<td>Unknown reservation ID</td>
</tr>
<tr>
<td>PBSE_delProgress</td>
<td>15085</td>
<td>Delete already in progress</td>
</tr>
<tr>
<td>PBSE_BADTDSPEC</td>
<td>15086</td>
<td>Bad time specification(s)</td>
</tr>
<tr>
<td>PBSE_RESVMSG</td>
<td>15087</td>
<td>So reply_text can return a msg</td>
</tr>
<tr>
<td>PBSE_BADNODESPEC</td>
<td>15089</td>
<td>Node(s) specification error</td>
</tr>
<tr>
<td>PBSE_LICENSESPEC</td>
<td>15090</td>
<td>Licensed CPUs exceeded</td>
</tr>
<tr>
<td>PBSE_LICENSEINV</td>
<td>15091</td>
<td>License is invalid</td>
</tr>
</tbody>
</table>
## Table 11-1: Error Codes

<table>
<thead>
<tr>
<th>Error Name</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBSE_RESVAUTH_H</td>
<td>15092</td>
<td>Host not authorized to make AR</td>
</tr>
<tr>
<td>PBSE_RESVAUTH_G</td>
<td>15093</td>
<td>Group not authorized to make AR</td>
</tr>
<tr>
<td>PBSE_RESVAUTH_U</td>
<td>15094</td>
<td>User not authorized to make AR</td>
</tr>
<tr>
<td>PBSE_R_UID</td>
<td>15095</td>
<td>Bad effective UID for reservation</td>
</tr>
<tr>
<td>PBSE_R_GID</td>
<td>15096</td>
<td>Bad effective GID for reservation</td>
</tr>
<tr>
<td>PBSE_IBMSPSWITCH</td>
<td>15097</td>
<td>IBM SP Switch error</td>
</tr>
<tr>
<td>PBSE_LICENSEUNAV</td>
<td>15098</td>
<td>Floating License unavailable</td>
</tr>
<tr>
<td></td>
<td>15099</td>
<td>UNUSED</td>
</tr>
<tr>
<td>PBSE_RESCNOTSTR</td>
<td>15100</td>
<td>Resource is not of type string</td>
</tr>
<tr>
<td>PBSE_SSIGNON_UNSET_REJECT</td>
<td>15101</td>
<td>rejected if SVR_ssignon_enable not set</td>
</tr>
<tr>
<td>PBSE_SSIGNON_SET_REJECT</td>
<td>15102</td>
<td>rejected if SVR_ssignon_enable set</td>
</tr>
<tr>
<td>PBSE_SSIGNON_BAD_TRANSITION1</td>
<td>15103</td>
<td>bad attempt: true to false</td>
</tr>
<tr>
<td>PBSE_SSIGNON_NOCONNECT_DEST</td>
<td>15105</td>
<td>couldn't connect to destination host during a user migration request</td>
</tr>
<tr>
<td>PBSE_SSIGNON_NO_PASSWORD</td>
<td>15106</td>
<td>no per-user/per-server password</td>
</tr>
<tr>
<td>PBSE_MaxArraySize</td>
<td>15107</td>
<td>max array size exceeded</td>
</tr>
<tr>
<td>PBSE_INVALSELECTRES esc</td>
<td>15108</td>
<td>resource invalid in select spec</td>
</tr>
<tr>
<td>PBSE_INVALJOBRES esc</td>
<td>15109</td>
<td>invalid job resource</td>
</tr>
<tr>
<td>PBSE_INVALNODEPLACE</td>
<td>15110</td>
<td>node invalid w/place</td>
</tr>
<tr>
<td>PBSE_PLACENOS elect</td>
<td>15111</td>
<td>cannot have place w/o select</td>
</tr>
<tr>
<td>PBSE_INRECTHOP</td>
<td>15112</td>
<td>too many indirect resource levels</td>
</tr>
<tr>
<td>PBSE_INDRECTBT</td>
<td>15113</td>
<td>target resource undefined</td>
</tr>
<tr>
<td>PBSE_NGBLU GENE</td>
<td>15114</td>
<td>No node_group_enable on BlueGene</td>
</tr>
<tr>
<td>PBSE_NODESTALE</td>
<td>15115</td>
<td>Cannot change state of stale vnode</td>
</tr>
<tr>
<td>PBSE_DUPRES C</td>
<td>15116</td>
<td>cannot dupe resource within a chunk</td>
</tr>
<tr>
<td>PBSE_CONNFULL</td>
<td>15117</td>
<td>server connection table full</td>
</tr>
<tr>
<td>PBSE_LICENSE_MIN_BADVAL</td>
<td>15118</td>
<td>bad value for pbs_license_min</td>
</tr>
<tr>
<td>PBSE_LICENSE_MAX_BADVAL</td>
<td>15119</td>
<td>bad value for pbs_license_max</td>
</tr>
<tr>
<td>PBSE_LICENSE_LINGER_BADVAL</td>
<td>15120</td>
<td>bad value for pbs_license_linger_time</td>
</tr>
<tr>
<td>PBSE_LICENSE_SERVER_DOWN</td>
<td>15121</td>
<td>License server is down</td>
</tr>
<tr>
<td>PBSE_LICENSE_BAD_ACTION</td>
<td>15122</td>
<td>Not allowed action with licensing</td>
</tr>
<tr>
<td>PBSE_BAD_FORMULA</td>
<td>15123</td>
<td>invalid sort formula</td>
</tr>
<tr>
<td>PBSE_BAD_FORMULA_KW</td>
<td>15124</td>
<td>invalid keyword in formula</td>
</tr>
</tbody>
</table>
## Table 11-1: Error Codes

<table>
<thead>
<tr>
<th>Error Name</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBSE_BAD_FORMULA_TYPE</td>
<td>15125</td>
<td>invalid resource type in formula</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_YEARLY</td>
<td>15126</td>
<td>reservation duration exceeds 1 year</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_MONTHLY</td>
<td>15127</td>
<td>reservation duration exceeds 1 month</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_WEEKLY</td>
<td>15128</td>
<td>reservation duration exceeds 1 week</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_DAILY</td>
<td>15129</td>
<td>reservation duration exceeds 1 day</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_HOURLY</td>
<td>15130</td>
<td>reservation duration exceeds 1 hour</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_MINUTELY</td>
<td>15131</td>
<td>reservation duration exceeds 1 minute</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_SECONDLY</td>
<td>15132</td>
<td>reservation duration exceeds 1 second</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_SYNTAX</td>
<td>15133</td>
<td>invalid recurrence rule syntax</td>
</tr>
<tr>
<td>PBSE_BAD_RRULE_SYNTAX2</td>
<td>15134</td>
<td>invalid recurrence rule syntax</td>
</tr>
<tr>
<td>PBSE_BAD_ICAL_TZ</td>
<td>15135</td>
<td>Undefined timezone info directory</td>
</tr>
<tr>
<td>PBSE_HOOKERROR</td>
<td>15136</td>
<td>error encountered related to hooks</td>
</tr>
<tr>
<td>PBSE_NEEDQUET</td>
<td>15137</td>
<td>need queue type set</td>
</tr>
<tr>
<td>PBSE_ETEERROR</td>
<td>15138</td>
<td>not allowed to alter attribute when <code>eligible_time_enable</code> is off</td>
</tr>
<tr>
<td>PBSE_HISTJOBID</td>
<td>15139</td>
<td>History job ID</td>
</tr>
<tr>
<td>PBSE_JOBHISTNOTSET</td>
<td>15140</td>
<td><code>job_history_enable</code> not SET</td>
</tr>
<tr>
<td>PBSE_MIXENTLIMS</td>
<td>15141</td>
<td>mixing old and new limit enforcement</td>
</tr>
<tr>
<td></td>
<td>15145</td>
<td>Server host not allowed to be provisioned</td>
</tr>
<tr>
<td></td>
<td>15146</td>
<td>While provisioning, provisioning attributes can't be modified</td>
</tr>
<tr>
<td></td>
<td>15147</td>
<td>State of provisioning vnode can't be changed</td>
</tr>
<tr>
<td></td>
<td>15148</td>
<td>Vnode can't be deleted while provisioning</td>
</tr>
<tr>
<td></td>
<td>15149</td>
<td>Attempt to set an AOE that is not in <code>resources_available.aoe</code></td>
</tr>
<tr>
<td></td>
<td>15150</td>
<td>Illegal job/reservation submission/alteration</td>
</tr>
<tr>
<td>PBSE_MOM_INCOMPLETE_HOOK</td>
<td>15167</td>
<td>Execution hooks not fully transferred to a particular MoM</td>
</tr>
<tr>
<td>PBSE_MOM_REJECT_ROOT_SCRIPTS</td>
<td>15168</td>
<td>A MoM has rejected a request to copy a hook-related file, or a job script to be executed by root</td>
</tr>
<tr>
<td>PBSE_HOOK_REJECT</td>
<td>15169</td>
<td>A MoM received a reject result from a mom hook</td>
</tr>
<tr>
<td>PBSE_HOOK_REJECT_RERUNJOB</td>
<td>15170</td>
<td>Hook rejection requiring a job to be rerun</td>
</tr>
<tr>
<td>PBSE_HOOK_REJECT_DELETEJOB</td>
<td>15171</td>
<td>Hook rejection requiring a job to be deleted</td>
</tr>
<tr>
<td>PBSE_JOBNBIG</td>
<td>15173</td>
<td>Submitted job or reservation name is too long</td>
</tr>
</tbody>
</table>

**Resource monitor specific error codes**
## Table 11-1: Error Codes

<table>
<thead>
<tr>
<th>Error Name</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBSE_RMUNKNOWN</td>
<td>15201</td>
<td>Resource unknown</td>
</tr>
<tr>
<td>PBSE_RMBADPARAM</td>
<td>15202</td>
<td>Parameter could not be used</td>
</tr>
<tr>
<td>PBSE_RMNOPARAM</td>
<td>15203</td>
<td>A needed parameter did not exist</td>
</tr>
<tr>
<td>PBSE_RMEXIST</td>
<td>15204</td>
<td>Something specified didn't exist</td>
</tr>
<tr>
<td>PBSE_RMSYSTEM</td>
<td>15205</td>
<td>A system error occurred</td>
</tr>
<tr>
<td>PBSE_RMPART</td>
<td>15206</td>
<td>Only part of reservation made</td>
</tr>
<tr>
<td>PBSE_SSIGNON_BAD_TRANSITION2</td>
<td>15207</td>
<td>bad attempt: false to true</td>
</tr>
<tr>
<td>PBSE_ALPSRELERR</td>
<td>15209</td>
<td>PBS is unable to release the ALPS reservation</td>
</tr>
</tbody>
</table>
When reading the PBS event logfiles, you may see messages of the form “Type 19 request received from PBS_Server...”. These “type codes” correspond to different PBS batch requests. The following table lists all the PBS type codes and the corresponding request of each.

**Table 12-1: Request Codes**

<table>
<thead>
<tr>
<th>Numeric Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PBS_BATCH_Connect</td>
</tr>
<tr>
<td>1</td>
<td>PBS_BATCH_QueueJob</td>
</tr>
<tr>
<td>2</td>
<td>UNUSED</td>
</tr>
<tr>
<td>3</td>
<td>PBS_BATCH_jobscript</td>
</tr>
<tr>
<td>4</td>
<td>PBS_BATCH_RdytoCommit</td>
</tr>
<tr>
<td>5</td>
<td>PBS_BATCH_Commit</td>
</tr>
<tr>
<td>6</td>
<td>PBS_BATCH_DeleteJob</td>
</tr>
<tr>
<td>7</td>
<td>PBS_BATCH_HoldJob</td>
</tr>
<tr>
<td>8</td>
<td>PBS_BATCH_LocateJob</td>
</tr>
<tr>
<td>9</td>
<td>PBS_BATCH_Manager</td>
</tr>
<tr>
<td>10</td>
<td>PBS_BATCH_MessJob</td>
</tr>
<tr>
<td>11</td>
<td>PBS_BATCH_ModifyJob</td>
</tr>
<tr>
<td>12</td>
<td>PBS_BATCH_MoveJob</td>
</tr>
<tr>
<td>13</td>
<td>PBS_BATCH_ReleaseJob</td>
</tr>
<tr>
<td>14</td>
<td>PBS_BATCH_Rerun</td>
</tr>
<tr>
<td>15</td>
<td>PBS_BATCH_RunJob</td>
</tr>
<tr>
<td>16</td>
<td>PBS_BATCH_SelectJobs</td>
</tr>
<tr>
<td>17</td>
<td>PBS_BATCH_Shutdown</td>
</tr>
<tr>
<td>18</td>
<td>PBS_BATCH_SignalJob</td>
</tr>
<tr>
<td>19</td>
<td>PBS_BATCH_StatusJob</td>
</tr>
<tr>
<td>20</td>
<td>PBS_BATCH_StatusQue</td>
</tr>
<tr>
<td>21</td>
<td>PBS_BATCH_StatusSvr</td>
</tr>
<tr>
<td>22</td>
<td>PBS_BATCH_TrackJob</td>
</tr>
<tr>
<td>23</td>
<td>PBS_BATCH_AsyrunJob</td>
</tr>
<tr>
<td>24</td>
<td>PBS_BATCH_Rescq</td>
</tr>
<tr>
<td>25</td>
<td>PBS_BATCH_ReservedResc</td>
</tr>
</tbody>
</table>
# Request Codes

## Table 12-1: Request Codes

<table>
<thead>
<tr>
<th>Numeric Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>PBS_BATCH_ReleaseResc</td>
</tr>
<tr>
<td>27</td>
<td>PBS_BATCH_FailOver</td>
</tr>
<tr>
<td>48</td>
<td>PBS_BATCH_SelStat</td>
</tr>
<tr>
<td>49</td>
<td>PBS_BATCH_AuthenUser</td>
</tr>
<tr>
<td>50</td>
<td>PBS_BATCH_OrderJob</td>
</tr>
<tr>
<td>51</td>
<td>PBS_BATCH_StatusNode</td>
</tr>
<tr>
<td>52</td>
<td>PBS_BATCH_RegistDep</td>
</tr>
<tr>
<td>54</td>
<td>PBS_BATCH_CopyFiles</td>
</tr>
<tr>
<td>55</td>
<td>PBS_BATCH_DelFiles</td>
</tr>
<tr>
<td>56</td>
<td>PBS_BATCH_JobObit</td>
</tr>
<tr>
<td>57</td>
<td>PBS_BATCH_MvJobFile</td>
</tr>
<tr>
<td>58</td>
<td>PBS_BATCH_StatusNode</td>
</tr>
<tr>
<td>59</td>
<td>PBS_BATCH_Disconnect</td>
</tr>
<tr>
<td>60</td>
<td>UNUSED</td>
</tr>
<tr>
<td>61</td>
<td>UNUSED</td>
</tr>
<tr>
<td>62</td>
<td>PBS_BATCH_JobCred</td>
</tr>
<tr>
<td>63</td>
<td>PBS_BATCH_CopyFiles_Cred</td>
</tr>
<tr>
<td>64</td>
<td>PBS_BATCH_DelFiles_Cred</td>
</tr>
<tr>
<td>65</td>
<td>PBS_BATCH_GSS_Context</td>
</tr>
<tr>
<td>66</td>
<td>UNUSED</td>
</tr>
<tr>
<td>67</td>
<td>UNUSED</td>
</tr>
<tr>
<td>68</td>
<td>UNUSED</td>
</tr>
<tr>
<td>69</td>
<td>UNUSED</td>
</tr>
<tr>
<td>70</td>
<td>PBS_BATCH_SubmitResv</td>
</tr>
<tr>
<td>71</td>
<td>PBS_BATCH_StatusResv</td>
</tr>
<tr>
<td>72</td>
<td>PBS_BATCH_DeleteResv</td>
</tr>
<tr>
<td>73</td>
<td>PBS_BATCH_UserCred</td>
</tr>
<tr>
<td>74</td>
<td>PBS_BATCH_UserMigrate</td>
</tr>
<tr>
<td>75</td>
<td>PBS_BATCH_ConfirmResv</td>
</tr>
<tr>
<td>80</td>
<td>PBS_BATCH_DefSchReply</td>
</tr>
<tr>
<td>81</td>
<td>PBS_BATCH_StatusSched</td>
</tr>
<tr>
<td>82</td>
<td>PBS_BATCH_StatusRsc</td>
</tr>
</tbody>
</table>
13.1 PBS Environment Variables

The following table lists the PBS environment variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Origin</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCPUS</td>
<td></td>
<td>Number of threads, defaulting to number of CPUs, on the vnode</td>
</tr>
<tr>
<td>OMP_NUM_THREADS</td>
<td></td>
<td>Same as NCPUS.</td>
</tr>
<tr>
<td>PBS ARRAY ID</td>
<td>Server</td>
<td>Identifier for job arrays. Consists of sequence number.</td>
</tr>
<tr>
<td>PBS ARRAY_INDEX</td>
<td>Server</td>
<td>Index number of subjob in job array.</td>
</tr>
<tr>
<td>PBS_CONF_FILE</td>
<td></td>
<td>Path to pbs.conf</td>
</tr>
<tr>
<td>PBS_CPUSETDEDICATED</td>
<td>Set by mpiexec</td>
<td>Asserts exclusive use of resources in assigned cpuset.</td>
</tr>
<tr>
<td>PBS_DEFAULT</td>
<td></td>
<td>Name of default PBS server</td>
</tr>
<tr>
<td>PBS_DATASERVICE_USER</td>
<td>Admin, during installation</td>
<td>Account used by data service.</td>
</tr>
<tr>
<td>PBS_ENVIRONMENT</td>
<td></td>
<td>Indicates job type: PBS_BATCH or PBS_INTERACTIVE</td>
</tr>
<tr>
<td>PBS_JOBCOOKIE</td>
<td></td>
<td>Unique identifier for inter-MoM job-based communication.</td>
</tr>
<tr>
<td>PBS_JOBDIR</td>
<td></td>
<td>Pathname of job-specific staging and execution directory</td>
</tr>
<tr>
<td>PBS_JOBID</td>
<td>Server</td>
<td>The job identifier assigned to the job or job array by the batch system.</td>
</tr>
<tr>
<td>PBS_JOBNAME</td>
<td>User</td>
<td>The job name supplied by the user.</td>
</tr>
<tr>
<td>PBS_MOMPORT</td>
<td></td>
<td>Port number on which this job’s MoMs will communicate.</td>
</tr>
<tr>
<td>PBS_NODEFILE</td>
<td></td>
<td>The filename containing a list of vnodes assigned to the job.</td>
</tr>
<tr>
<td>PBS_NODENUM</td>
<td></td>
<td>Logical vnode number of this vnode allocated to the job.</td>
</tr>
<tr>
<td>PBS_O_HOME</td>
<td>Submission environment</td>
<td>Value of HOME from submission environment.</td>
</tr>
<tr>
<td>PBS_O_HOST</td>
<td>Submission environment</td>
<td>The host name on which the qsub command was executed.</td>
</tr>
<tr>
<td>PBS_O_LANG</td>
<td>Submission environment</td>
<td>Value of LANG from submission environment.</td>
</tr>
</tbody>
</table>
### Table 13-1: PBS Environment Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Origin</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_O_LOGNAME</td>
<td>Submission env.</td>
<td>Value of LOGNAME from submission environment</td>
</tr>
<tr>
<td>PBS_O_MAIL</td>
<td>Submission env.</td>
<td>Value of MAIL from submission environment</td>
</tr>
<tr>
<td>PBS_O_PATH</td>
<td>Submission env.</td>
<td>Value of PATH from submission environment</td>
</tr>
<tr>
<td>PBS_O_QUEUE</td>
<td>Submission env.</td>
<td>The original queue name to which the job was submitted.</td>
</tr>
<tr>
<td>PBS_O_SHELL</td>
<td>Submission env.</td>
<td>Value of SHELL from submission environment</td>
</tr>
<tr>
<td>PBS_O_SYSTEM</td>
<td>Submission env.</td>
<td>The operating system name where qsub was executed.</td>
</tr>
<tr>
<td>PBS_O_TZ</td>
<td>Submission env.</td>
<td>Value of TZ from submission environment</td>
</tr>
<tr>
<td>PBS_O_WORKDIR</td>
<td>Submission env.</td>
<td>The absolute path of directory where qsub was executed.</td>
</tr>
<tr>
<td>PBS_QUEUE</td>
<td>Submission env.</td>
<td>The name of the queue from which the job is executed.</td>
</tr>
<tr>
<td>PBS_SERVER</td>
<td>Submission env.</td>
<td>The name of the default PBS server.</td>
</tr>
<tr>
<td>PBS_TASKNUM</td>
<td>Submission env.</td>
<td>The task (process) number for the job on this vnode.</td>
</tr>
<tr>
<td>PBS_TMPDIR</td>
<td></td>
<td>The job-specific temporary directory for this job.</td>
</tr>
</tbody>
</table>
The following table lists all the PBS files and directories; owner and permissions are specific to Linux systems.

Table 14-1: File Listing

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>/opt/pbs/default/etc/pbs_bootcheck.py</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4111</td>
</tr>
<tr>
<td>/var/tmp/pbs_bootcheck.py</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4111</td>
</tr>
<tr>
<td>/var/tmp/pbs_boot_check</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>188</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See &quot;Discovering Last Reboot Time of Server&quot; on page 543 in the PBS Professional Administrator’s Guide.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_EXEC/</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/bin</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/bin/nqs2pbs</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>16062</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbsdsh</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>111837</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbsnodes</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>153004</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbs_dataservice</td>
<td>root</td>
<td>-rwx------</td>
<td></td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbs_hostn</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>35493</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbs_rdel</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>151973</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbs_rstat</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>156884</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbs_rsub</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>167446</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbs_tclsh</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>857552</td>
</tr>
<tr>
<td>PBS_EXEC/bin/pbs_wish</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>1592236</td>
</tr>
<tr>
<td>PBS_EXEC/bin/printjob</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>42667</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qalter</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>210723</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qdel</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>164949</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qdisable</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>139559</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qenable</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>139558</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qhold</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>165368</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qmgr</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>202526</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qmove</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>160932</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qmsg</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>160408</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qorder</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>146393</td>
</tr>
<tr>
<td>Directory / File</td>
<td>Owner</td>
<td>Permission</td>
<td>Average Size</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qrerun</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>157228</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qrls</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>165361</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qrun</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>160978</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qselect</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>163266</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qsig</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>160083</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qstart</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>139589</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qstat</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>207532</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qstop</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>139584</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qsub</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>275460</td>
</tr>
<tr>
<td>PBS_EXEC/bin/qterm</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>132188</td>
</tr>
<tr>
<td>PBS_EXEC/bin/tracejob</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>64730</td>
</tr>
<tr>
<td>PBS_EXEC/bin/xpbs</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>817</td>
</tr>
<tr>
<td>PBS_EXEC/bin/xpbsmon</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>817</td>
</tr>
<tr>
<td>PBS_EXEC/etc</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/etc/modulefile</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>749</td>
</tr>
<tr>
<td>PBS_EXEC/etc/pbs_db_schema.sql</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10522</td>
</tr>
<tr>
<td>PBS_EXEC/etc/pbs_dedicated</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>557</td>
</tr>
<tr>
<td>PBS_EXEC/etc/pbs_holidays</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2612</td>
</tr>
<tr>
<td>PBS_EXEC/etc/pbs_holidays.&lt;year&gt;</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2643</td>
</tr>
<tr>
<td>PBS_EXEC/etc/pbs_resource_group</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>657</td>
</tr>
<tr>
<td>PBS_EXEC/etc/pbs_sched_config</td>
<td>root</td>
<td>-r--r--r--</td>
<td>9791</td>
</tr>
<tr>
<td>PBS_EXEC/include</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/include/pbs_error.h</td>
<td>root</td>
<td>-r--r--r--</td>
<td>7543</td>
</tr>
<tr>
<td>PBS_EXEC/include/pbs_if1.h</td>
<td>root</td>
<td>-r--r--r--</td>
<td>17424</td>
</tr>
<tr>
<td>PBS_EXEC/include/rm.h</td>
<td>root</td>
<td>-r--r--r--</td>
<td>740</td>
</tr>
<tr>
<td>PBS_EXEC/include/tm.h</td>
<td>root</td>
<td>-r--r--r--</td>
<td>2518</td>
</tr>
<tr>
<td>PBS_EXEC/include/tm_.h</td>
<td>root</td>
<td>-r--r--r--</td>
<td>2236</td>
</tr>
<tr>
<td>PBS_EXEC/lib</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/libattr.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>390274</td>
</tr>
<tr>
<td>PBS_EXEC/lib/liblog.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>101230</td>
</tr>
<tr>
<td>PBS_EXEC/lib/libnet.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>145968</td>
</tr>
<tr>
<td>PBS_EXEC/lib/libpbs.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1815486</td>
</tr>
<tr>
<td>Directory / File</td>
<td>Owner</td>
<td>Permission</td>
<td>Average Size</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>PBS_EXEC/lib/libsite.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>132906</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI/pbsrun.ch_gm.init.in</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>9924</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI/pbsrun.ch_mx.init.in</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>9731</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI/pbsrun.gm_mpd.init.in</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10767</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI/pbsrun.intelmpi.init.in</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10634</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI/pbsrun.mpich2.init.in</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10694</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI/pbsrun.mx_mpd.init.in</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10770</td>
</tr>
<tr>
<td>PBS_EXEC/lib/MPI/sgiMPI.awk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6564</td>
</tr>
<tr>
<td>PBS_EXEC/lib/pbs_sched.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>822026</td>
</tr>
<tr>
<td>PBS_EXEC/lib/pm</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/pm/PBS.pm</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3908</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_acctname.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3484</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_after_depend.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>8637</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_auto_upd.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3384</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_before_depend.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>8034</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bin</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bin/xpbs_datadump</td>
<td>root</td>
<td>-rw-xr-xr-x</td>
<td>190477</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bin/xpbs_scriptload</td>
<td>root</td>
<td>-rw-xr-xr-x</td>
<td>173176</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bindings.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>26029</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/curve_down_arrow.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>320</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/curve_up_arrow.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>314</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/cyclist-only.xbm</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2485</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/Downarrow.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>299</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/hourglass.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>557</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/iconize.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>287</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/logo.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>67243</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/maximize.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>287</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/sm_down_arrow.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>311</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/bitmaps/sm_up_arrow.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>305</td>
</tr>
</tbody>
</table>
### Table 14-1: File Listing

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_bitmaps/Uparrow.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>293</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_box.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>25912</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_button.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>18795</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_checkpoint.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6892</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_common.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>25940</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_concur.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>8445</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_datetime.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4533</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_email_list.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3094</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_entry.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>12389</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_fileselect.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7975</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/after_depend.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1746</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/auto_update.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>776</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/before_depend.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1413</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/concur.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1383</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/datetime.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>698</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/delete.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>632</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/email.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>986</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/fileselect.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1655</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/hold.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>538</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/main.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>15220</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/message.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>677</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/misc.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4194</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/modify.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6034</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/move.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>705</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/notes.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3724</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/preferences.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1645</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/release.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>573</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.acctname.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>609</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.checkpoint.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1133</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.hold.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>544</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.jobname.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>600</td>
</tr>
<tr>
<td>Directory / File</td>
<td>Owner</td>
<td>Permission</td>
<td>Average Size</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.owners.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1197</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.priority.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>748</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.qtime.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>966</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.rerun.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>541</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.resources.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1490</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/select.states.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>562</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/signal.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>675</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/staging.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3702</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/submit.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>9721</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/terminate.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>635</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_help/trackjob.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2978</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_hold.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3539</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_jobname.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3375</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_listbox.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10544</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_main.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>24147</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_misc.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>14526</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_owners.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4509</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_pbs.tcl</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>52524</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_pref.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3445</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_preferences.tcl</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4323</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_prefsave.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1378</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_priority.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4434</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_galter.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>35003</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_gdel.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3175</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_qhold.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3676</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_qmove.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3326</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_qmsg.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4032</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_qrls.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3674</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_qsig.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>5171</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_gsub.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>37466</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_qterm.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3204</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_qtime.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>5790</td>
</tr>
<tr>
<td>Directory / File</td>
<td>Owner</td>
<td>Permission</td>
<td>Average Size</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_rerun.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2802</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_res.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4807</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_spinbox.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7144</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_staging.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>12183</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_state.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3657</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_text.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2738</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_trackjob.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>13605</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/pbs_vmgr.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1428</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/tclIndex</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>19621</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/xpbs.src.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>96666</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbs/xpbsrc</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2986</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_auto_upd.tk</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_bindings.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3281</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_cluster.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>9288</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_bitmaps</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_bitmaps/cyclist-only.xbm</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2485</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_bitmaps/hourglass.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>557</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_bitmaps/iconize.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>287</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_bitmaps/logo.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>67243</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_bitmaps/maximize.bmp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>287</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_box.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>15607</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_button.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7543</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_cluster.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>44406</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_color.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>5634</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_common.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>5716</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_dialog.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>8398</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_entry.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10697</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_expr.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6163</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_help</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_help/auto_update.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>624</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_help/main.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>15718</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_help/notes.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>296</td>
</tr>
<tr>
<td>Directory / File</td>
<td>Owner</td>
<td>Permission</td>
<td>Average Size</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_help/pref.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1712</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_help/prefQuery.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4621</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_help/prefServer.hlp</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1409</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_listbox.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10640</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_main.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6760</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_node.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>60640</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_pbs.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7090</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_pref.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>22117</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_preferences.tcl</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10212</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_prefsave.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1482</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_spinbox.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7162</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_system.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>47760</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/pbs_wmgr.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1140</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/tclIndex</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>30510</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/xpbsmon.src.tk</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>13999</td>
</tr>
<tr>
<td>PBS_EXEC/lib/xpbsmon/xpbsmonrc</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3166</td>
</tr>
<tr>
<td>PBS_EXEC/libexec/au-nodeupdate.pl</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>10506</td>
</tr>
<tr>
<td>PBS_EXEC/libexec/install_db</td>
<td>root</td>
<td>-rw-x------</td>
<td>10059</td>
</tr>
<tr>
<td>PBS_EXEC/libexec/pbs_habitat</td>
<td>root</td>
<td>-rw-x------</td>
<td>25568</td>
</tr>
<tr>
<td>PBS_EXEC/libexec/pbs_init.d</td>
<td>root</td>
<td>-rw-x------</td>
<td>29104</td>
</tr>
<tr>
<td>PBS_EXEC/libexec/pbs_postinstall</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/man</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3276</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/nqs2pbs</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>5376</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/pbsdeh.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2978</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/pbs_rdel.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2342</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/pbs_rstat.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2682</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/pbs_rspub.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>9143</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qalter.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>21569</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qdel.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3363</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qhold.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4323</td>
</tr>
</tbody>
</table>
## Table 14-1: File Listing

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_EXEC/man/man1/qmove.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3343</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qmsg.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3244</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qorder.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3028</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qrerun.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2965</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qr1s.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3927</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qselect.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>12690</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qsig.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3817</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qstat.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>15274</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/qsub.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>36435</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/xpbs.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>26956</td>
</tr>
<tr>
<td>PBS_EXEC/man/man1/xpbsmon.1B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>26365</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_alterjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>5475</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_connect.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3493</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_default.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2150</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_deljob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3081</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_disconnect.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1985</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_geterrmsg.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2473</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_holdjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3006</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_manager.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4337</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_movejob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3220</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_msgjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2912</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_orderjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2526</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_rerunjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2531</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_rescreserve.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4125</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_rlsjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3043</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_runjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3484</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_selectjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7717</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_sigjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3108</td>
</tr>
<tr>
<td>PBS EXEC/man/man3/pbs_stagein.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3198</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_statjob.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4618</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_statnode.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3925</td>
</tr>
</tbody>
</table>
Table 14-1: File Listing

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_EXEC/man/man3/pbs_statque.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4009</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_statserver.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3674</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_submit.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6320</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_submiserv.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3878</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/pbs_terminate.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3322</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/rpp.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6476</td>
</tr>
<tr>
<td>PBS_EXEC/man/man3/tm.3B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>11062</td>
</tr>
<tr>
<td>PBS_EXEC/man/man7</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/man/man7/pbs_job_attributes.7B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>15920</td>
</tr>
<tr>
<td>PBS_EXEC/man/man7/pbs_node_attributes.7B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7973</td>
</tr>
<tr>
<td>PBS_EXEC/man/man7/pbs_queue_attributes.7B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>11062</td>
</tr>
<tr>
<td>PBS_EXEC/man/man7/pbs_resources.7B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>22124</td>
</tr>
<tr>
<td>PBS_EXEC/man/man7/pbs_resv_attributes.7B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>11662</td>
</tr>
<tr>
<td>PBS_EXEC/man/man7/pbs_server_attributes.7B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>14327</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/mpiexec.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4701</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs-report.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>19221</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbsfs.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3703</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbsnodes.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3441</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbsrun.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>20937</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbsrun_unwrap.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2554</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbsrun_wrap.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3855</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_attach.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3790</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_hostn.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2781</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_idled.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2628</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_lamboot.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2739</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_migrate_users.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2519</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_mom.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>23496</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_mpihp.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4120</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_mpilam.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2647</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_mpirun.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3130</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_password.8B</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3382</td>
</tr>
</tbody>
</table>
### Table 14-1: File Listing

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_EXEC/man/man8/pbs_poe.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>3973</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_probe.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>3344</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_sched_cc.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>6731</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_server.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>7914</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_tclsh.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>2475</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_tmrsh.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>3556</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/pbs_wish.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>2123</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/printjob.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>2823</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/qdisable.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>3104</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/qenable.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>2937</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/qmgr.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>7282</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/qrun.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>2850</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/qstart.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>2966</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/qstop.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>2963</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/qterm.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>4839</td>
</tr>
<tr>
<td>PBS_EXEC/man/man8/tracejob.8B</td>
<td>root</td>
<td>-rw-r-r--</td>
<td>4664</td>
</tr>
<tr>
<td>PBS_EXEC/pgsql</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/sbin</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>6296</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs-report</td>
<td>root</td>
<td>-rw-r-xr-x</td>
<td>663707</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbsfs</td>
<td>root</td>
<td>-rw-r-xr-x</td>
<td>38688</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_demux</td>
<td>root</td>
<td>-rw-r-xr-x</td>
<td>99373</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_idled</td>
<td>root</td>
<td>-rw-r-xr-x</td>
<td>133142</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_iff</td>
<td>root</td>
<td>-rwx-------</td>
<td>839326</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_mom</td>
<td>root</td>
<td>-rwx-------</td>
<td>0</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_mom.cpuset</td>
<td>root</td>
<td>-rwx-------</td>
<td>0</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_mom.standard</td>
<td>root</td>
<td>-rwx-------</td>
<td>0</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_probe</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>83108</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_rcp</td>
<td>root</td>
<td>-rwxr-xr-x</td>
<td>75274</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_sched</td>
<td>root</td>
<td>-rwx-------</td>
<td>705478</td>
</tr>
<tr>
<td>PBS_EXEC/sbin/pbs_server</td>
<td>root</td>
<td>-rwx-------</td>
<td>1133650</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/bin</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
</tbody>
</table>
### Table 14-1: File Listing

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_EXEC/tcltk/bin/tclsh8.3</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>552763</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/bin/wish8.3</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1262257</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/include</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/include/tcl.h</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>57222</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/include/tclDecls.h</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>123947</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/include/tk.h</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>47420</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/include/tkDecls.h</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>80181</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/libtcl8.3.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>777558</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/libtclstub8.3.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1832</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/libtk8.3.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1021024</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/libtkstub8.3.a</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3302</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/tcl8.3</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/tclConfig.sh</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>7076</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/tk8.3</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/lib/tkConfig.sh</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>3822</td>
</tr>
<tr>
<td>PBS_EXEC/tcltk/license.terms</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>2233</td>
</tr>
<tr>
<td>PBS_HOME</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/aux</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/checkpoint</td>
<td>root</td>
<td>drwx------</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/datastore</td>
<td>data service account</td>
<td>-rwx------</td>
<td></td>
</tr>
<tr>
<td>PBS_HOME/mom_logs</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/mom_priv</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/mom_priv/config</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>18</td>
</tr>
<tr>
<td>PBS_HOME/mom_priv/jobs</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/mom_priv/mom.lock</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4</td>
</tr>
<tr>
<td>PBS_HOME/pbs_environment</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>0</td>
</tr>
<tr>
<td>PBS_HOME/sched_log</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/sched_priv</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/sched_priv/dedicated_time</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>557</td>
</tr>
<tr>
<td>PBS_HOME/sched_priv/holidays</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>1228</td>
</tr>
</tbody>
</table>
# Table 14-1: File Listing

<table>
<thead>
<tr>
<th>Directory / File</th>
<th>Owner</th>
<th>Permission</th>
<th>Average Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_HOME/sched_priv/resource_group</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>0</td>
</tr>
<tr>
<td>PBS_HOME/sched_priv/sched.lock</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>4</td>
</tr>
<tr>
<td>PBS_HOME/sched_priv/sched_config</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>6370</td>
</tr>
<tr>
<td>PBS_HOME/sched_priv/sched_out</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>0</td>
</tr>
<tr>
<td>PBS_HOME/server_logs</td>
<td>root</td>
<td>drwxr-xr-x</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv</td>
<td>root</td>
<td>drwxr-x---</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/accounting</td>
<td>root</td>
<td>drwxr-x---</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/acl_groups</td>
<td>root</td>
<td>drwxr-x---</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/acl_hosts</td>
<td>root</td>
<td>drwxr-x---</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/acl_svr</td>
<td>root</td>
<td>drwxr-x---</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/acl_svr/managers</td>
<td>root</td>
<td>-rw-------</td>
<td>13</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/acl_users</td>
<td>root</td>
<td>drwxr-x---</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/jobs</td>
<td>root</td>
<td>drwxr-x---</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/license_file</td>
<td>root</td>
<td>-rw-r--r--</td>
<td>34</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/queues/newqueue</td>
<td>root</td>
<td>-rw-------</td>
<td>303</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/queues/workq</td>
<td>root</td>
<td>-rw-------</td>
<td>303</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/resourcedef</td>
<td>root</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PBS_HOME/server_priv/server.lock</td>
<td>root</td>
<td>-rw-------</td>
<td>4</td>
</tr>
<tr>
<td>PBS_HOME/server_priv/svrlive</td>
<td>root</td>
<td>-rw-------</td>
<td></td>
</tr>
<tr>
<td>PBS_HOME/server_priv/tracking</td>
<td>root</td>
<td>-rw-------</td>
<td>0</td>
</tr>
<tr>
<td>PBS_HOME/spool</td>
<td>root</td>
<td>drwxrwxrwt</td>
<td>4096</td>
</tr>
<tr>
<td>PBS_HOME/undelivered</td>
<td>root</td>
<td>drwxrwxrwt</td>
<td>4096</td>
</tr>
</tbody>
</table>
15 Introduction to PBS

15.1 Acknowledgements

PBS Professional is the enhanced commercial version of the PBS software originally developed for NASA. The NASA version had a number of corporate and individual contributors over the years, for which the PBS developers and PBS community is most grateful. Below we provide formal legal acknowledgements to corporate and government entities, then special thanks to individuals.

The NASA version of PBS contained software developed by NASA Ames Research Center, Lawrence Livermore National Laboratory, and MRJ Technology Solutions. In addition, it included software developed by the NetBSD Foundation, Inc., and its contributors as well as software developed by the University of California, Berkeley and its contributors.

Other contributors to the NASA version of PBS include Bruce Kelly and Clark Streeter of NERSC; Kent Crispin and Terry Heidelberg of LLNL; John Kochmar and Rob Pennington of Pittsburgh Supercomputing Center; and Dirk Grunwald of University of Colorado, Boulder. The ports of PBS to the Cray T3e and the IBM SP SMP were funded by DoD USAERDC; the port of PBS to the Cray SV1 was funded by DoD MSIC.

No list of acknowledgements for PBS would possibly be complete without special recognition of the first two beta test sites. Thomas Milliman of the Space Sciences Center of the University of New Hampshire was the first beta tester. Wendy Lin of Purdue University was the second beta tester and holds the honor of submitting more problem reports than anyone else outside of NASA.
action RG-256
$alps_client RG-256
$alps_release_timeout RG-256
$checkpoint_path RG-257
$clihost RG-257
$cpuset_error_action RG-257
$cpumult RG-258
$dcerefresh_delta RG-258
$enforce RG-258
$jobdir_root RG-259
$logevent RG-260
$max_check_poll RG-260
$max_load RG-260
$min_check_poll RG-261
$pbs_accounting_workload_mgmt RG-261
$sprologinalarm RG-261
$reject_root_scripts RG-261
$restart_background RG-261
$restart_transmogrify RG-262
$restrict_user RG-262
$restrict_user_exceptions RG-262
$restricted RG-262
$suspendsig RG-262
$tmpdir RG-263
$usecp RG-263
$vnodedef_additive RG-263
$wallmult RG-263

A
accelerator RG-280
accelerator_memory RG-280
accelerator_model RG-280
accept an action RG-1
access
  by group RG-7
  by user RG-17
  from host RG-7
  to a queue RG-7
  to a reservation RG-1
  to the Server RG-1
access control list RG-1
account RG-1
Account_Name
  job attribute RG-341
accounting

ACL RG-1, RG-379, RG-382, RG-383, RG-385
acl_group_enable
  queue attribute RG-324
acl_groups
  queue attribute RG-324
acl_host_enable RG-295
  queue attribute RG-324
acl_host_moms_enable RG-295
acl_hosts
  queue attribute RG-324
acl_resv_group_enable
  server attribute RG-295
acl_resv_groups
  server attribute RG-295
acl_resv_host_enable
  server attribute RG-295
acl_resv_hosts
  server attribute RG-295
acl_resv_user_enable
  server attribute RG-296
acl_resv_users
  server attribute RG-296
acl_roots
  server attribute RG-296
acl_user_enable
  queue attribute RG-324
  server attribute RG-296
acl_users
  queue attribute RG-324
  server attribute RG-296
action RG-1
  accept RG-1
  reject RG-14
active (failover) RG-1
Active Directory RG-1
Admin RG-1
administrator RG-2
Administrators RG-2
advance reservation RG-2, RG-392
Index

aggressive_provision RG-273
alarm
  hook attribute RG-360
alt_id
  job attribute RG-341
Ames Research Center RG-411
AOE RG-2
aoe RG-281
API RG-2
application checkpoint RG-2
application operating environment RG-2
arch RG-281
argument_list
  job attribute RG-342
array
  job attribute RG-342
array job RG-2, RG-8
array_id
  job attribute RG-342
array_index
  job attribute RG-342
array_indices_remaining
  job attribute RG-342
array_indices_submitted
  job attribute RG-342
array_state_count
  job attribute RG-342
attribute
  definition RG-2
  rerunnable RG-14
attribute name
  format RG-363
Authorized_Groups
  reservation attribute RG-317
Authorized_Hosts
  reservation attribute RG-317
Authorized_Users
  reservation attribute RG-318
avoid_provision RG-273

B
backfill RG-266
backfill_depth
  queue attribute RG-324
  server attribute RG-296
backfill_prime RG-266
Backfilling RG-2
batch job RG-8
batch processing RG-2
block
  job attribute RG-343
Boolean
  format RG-363

borrowing vnode RG-2
built-in hook RG-2
built-in resource RG-3
busy RG-375
by_queue RG-266

C
Checkpoint
  job attribute RG-343
checkpoint RG-256, RG-390, RG-402, RG-409
  restart RG-14
  restart file RG-14
  restart script RG-14
checkpoint and abort RG-3
checkpoint and restart RG-3
checkpoint/restart RG-3
checkpoint_abort RG-3, RG-256
checkpoint_min
  queue attribute RG-325
chunk RG-3
chunk set RG-3
chunk-level resource RG-3
cluster RG-3
commands RG-3
comment
  job attribute RG-343
  scheduler attribute RG-314
  server attribute RG-297
  vnode attribute RG-333
communication daemon RG-333
complex RG-4
configuration file
  Version 1 RG-17
  Version 2 RG-18
consumable resource RG-4
CPU RG-4
cpus_per_ssnodes RG-266
cpuset_create_flags RG-257
cpuset_destroy_delay RG-257
cput RG-281
creating a hook RG-4
crtime
  job attribute RG-344
  reservation attribute RG-318
current_aoe
  reservation attribute RG-333
current_eoe RG-333
custom resource RG-4

D
date
  format RG-363
datetime
format [RG-364]
debug
  hook attribute [RG-360]
dedicated_prefix [RG-266]
default_chunk
  queue attribute [RG-325]
  server attribute [RG-297]
default_qdel_arguments
  server attribute [RG-297]
default_qsub_arguments
  server attribute [RG-297]
default_queue
  server attribute [RG-297]
degraded reservation [RG-14]
delegation [RG-4]
depend
  job attribute [RG-344]
destination
    definition [RG-4]
destination identifier [RG-4]
    format [RG-364]
destination queue [RG-4]
destination server [RG-4]
directive [RG-5]
do_not_span_psets
    scheduler attribute [RG-314]
Domain Admin Account [RG-5]
Domain Admins [RG-5]
Domain User Account [RG-5]
Domain Users [RG-5]
down [RG-375]

E
egroup
  job attribute [RG-345]
eligible_time
  job attribute [RG-345]
eligible_time_enable
  server attribute [RG-298]
enabled
  hook attribute [RG-360]
  queue attribute [RG-325]
Endpoint [RG-5]
energy [RG-281]
Enterprise Admins [RG-5]
entity [RG-5]
entity share [RG-5]
Environment Variables [RG-397]
eoe [RG-281]
error codes [RG-389]
Error_Path
  job attribute [RG-345]
est_start_time_freq

server attribute [RG-298]
estimated
  job attribute [RG-346]
etime
  job attribute [RG-346]
euser
  job attribute [RG-346]
Event [RG-5]
event
  hook attribute [RG-361]
exec_host
  job attribute [RG-347]
exec_vnode [RG-282]
  job attribute [RG-347]
executable
  job attribute [RG-346]
execution event hooks [RG-5]
execution host [RG-6]
execution queue [RG-6]
Execution_Time
  job attribute [RG-347]
Exit_status
  job attribute [RG-348]
express_queue [RG-271]
externally-provided resources [RG-255]

F
fail_action
  hook attribute [RG-362]
failover [RG-6]
  idle [RG-7]
    primary scheduler [RG-13]
    primary server [RG-13]
    secondary scheduler [RG-15]
    secondary server [RG-15]
failure action [RG-6]
fair_share [RG-266]
Fairshare [RG-6]
fairshare [RG-271]
fairshare_decay_factor [RG-267]
fairshare_decay_time [RG-267]
fairshare_enforce_no_shares [RG-267]
fairshare_entity [RG-267]
fairshare_perc [RG-268]
fairshare_usage_res [RG-267]
File
  stage out [RG-16]
file [RG-282]
  stage in [RG-16]
  vnodedefs [RG-18]
file staging [RG-6]
Files
  MoM config [RG-381]
Index

nodes RG-380
finished jobs RG-6
flatuid
server attribute RG-298
FLicenses
server attribute RG-298
float
format RG-364
floating license RG-6
format
accounting log entry RG-363
attribute name RG-363
Boolean RG-363
date RG-363
datetime RG-364
destination identifier RG-364
float RG-364
host name RG-364
job array identifier RG-365
job array name RG-365
job array range RG-365
job identifier RG-365
job name RG-365
limit specification RG-366
logfile-date-time RG-366
pathname RG-367
PBS NAME RG-367
PBS password RG-367
project name RG-367
queue identifier RG-367
queue name RG-367
reservation name RG-367
size RG-368
string resource value RG-368
string_array RG-368
subjob identifier RG-368
username RG-368
Windows RG-369
vnode name RG-369
forward_x11_cookie job attribute RG-348
forward_x11_port job attribute RG-348
free RG-375
freq
hook attribute RG-362
from_route_only queue attribute RG-325
furnishing queue RG-6

access RG-7
ID (GID) RG-7
group limit RG-7
group_list
job attribute RG-348

H
half_life RG-267
hasnodes
queue attribute RG-325
hbmem RG-282
help_starving_jobs RG-267
history jobs RG-7
hold RG-7
Hold_Types
job attribute RG-348
Hook RG-7
hook
creating RG-4
importing RG-7, RG-8
provisioning RG-13
hooks
accept RG-1
action RG-1
execution event RG-5
non-job event RG-11
pre-execution event RG-13
reject action RG-14
host RG-7, RG-282
access RG-7
host name
format RG-364
Hot_Start RG-374
HTT RG-7

I
Idle RG-374
idle (failover) RG-7
importing a hook RG-7, RG-8
in_multivnode_host
vnode attribute RG-333
index
subjob RG-16
indirect resource RG-8
InfiniBand RG-48, RG-49
installation account RG-8
instance RG-11
interactive
job attribute RG-349
reservation attribute RG-319
Interactive job RG-8

G
global resource RG-7
group RG-7
Index

J
job
  attribute RG-14
  batch RG-8
  identifier RG-9
  kill RG-9
  owner RG-11
  rerunnable RG-14
  route RG-14
  state RG-9
job array RG-8
  identifier RG-8
  range RG-8
  subjob RG-16
  subjob index RG-16
job array identifier
  format RG-365
job array name
  format RG-365
job array range
  format RG-365
job ID RG-9
job identifier
  format RG-365
job name
  format RG-365
Job Submission Description Language RG-9
Job Substates RG-372
job_history_duration
  server attribute RG-299
job_history_enable
  server attribute RG-299
Job_Name
  job attribute RG-349
Job_Owner
  job attribute RG-349
job_priority RG-268
job_requeue_timeout
  server attribute RG-299
job_sort_formula
  server attribute RG-299
job_sort_formula_threshold
  scheduler attribute RG-314
job_sort_key RG-268
job_state
  job attribute RG-350
job.busy RG-375
jobdir
  job attribute RG-349
job-exclusive RG-375
jobs
  moved RG-11
  vnode attribute RG-333
jobscript_max_size
  server attribute RG-299
job-wide resource RG-9
Join_Path
  job attribute RG-351
JSDL RG-9
K
Keep_Files
  job attribute RG-351
key RG-268
kill job RG-9
kill_delay
  queue attribute RG-326
L
last_state_change_time RG-333
last_used_time RG-334
Leaf RG-9
license
  external RG-384
  token RG-17
  vnode attribute RG-334
license server RG-9
license server configuration
  redundant RG-14
License Server List Configuration RG-9
license_info
  vnode attribute RG-334
Limit
  Generic project limit RG-6
  Individual project limit RG-8
  overall RG-11
limit RG-10
  generic group limit RG-6
  generic user limit RG-7
  group limit RG-7
  individual group limit RG-8
  individual user limit RG-8
  project RG-13
  user limit RG-17
limit specification
  format RG-366
load balance RG-10
load_balancing RG-269
load_balancing_rr RG-269
local resource RG-10
log_events
  server attribute RG-300
log_filter RG-269
logfile-date-time
  format RG-366
Index

M
mail_from
    server attribute RG-300
Mail_Points
    job attribute RG-351
    reservation attribute RG-319
Mail_Users
    job attribute RG-351
    reservation attribute RG-319
maintenance RG-375
maintenance_jobs RG-334
Manager RG-10
managers
    server attribute RG-301
managing vnode RG-10
master provisioning script RG-10
master script RG-10
max_array_size
    queue attribute RG-326
    server attribute RG-301
max_concurrent_provision
    server attribute RG-301
max_group_res
    queue attribute RG-326
max_group_res_soft
    queue attribute RG-326
max_group_run
    queue attribute RG-326
max_group_run_soft
    queue attribute RG-326
max_queuable
    queue attribute RG-327
max_queued
    queue attribute RG-327
max_queue_res
    queue attribute RG-327
max_run
    queue attribute RG-327
max_run_res
    queue attribute RG-327
max_run_res_soft
    queue attribute RG-328
max_soft
    queue attribute RG-328
max_running
    queue attribute RG-328
max_starve RG-269
max_user_res
    queue attribute RG-328
max_user_res_soft
    queue attribute RG-328
max_user_run
    queue attribute RG-329
max_user_run_soft
    queue attribute RG-335
max_walltime RG-282
mem RG-283
mem_per_ssinode RG-269
memory-only vnode RG-10
memreserved RG-261
min_walltime RG-283
MOM RG-10
    subordinate RG-17
Mom
    vnode attribute RG-334
mom_resources RG-269
monitoring RG-10
Mother Superior RG-10
moved jobs RG-11
mpiprocs RG-283
MRJ Technology Solutions RG-411
mtime
    job attribute RG-352
    reservation attribute RG-319
multinodebusy RG-256
multi-vnode complex RG-381

N
naccelerators RG-284
name
    vnode attribute RG-334
NASA
    and PBS RG-411
nchunk RG-284
NCPUS RG-397
ncpus RG-285
nice RG-285
no_multinode_jobs
    vnode attribute RG-334
no_stdio_sockets
    job attribute RG-352
node
    definition RG-11
node_group_key
    queue attribute RG-329
    server attribute RG-305
node_sort_key RG-269
nodect RG-285
nodes RG-285
non-consumable resource RG-11
non-job event hooks RG-11
nonprimetime_prefix RG-270
normal_jobs RG-271
nppcu RG-11
nqs2pbs RG-26
ntype
    vnode attribute RG-335
Index

O
object RG-11
occurrence of a standing reservation RG-11
offline RG-375
OMP_NUM_THREADS RG-397
ompthreads RG-285
only Explicit_psets
  scheduler attribute RG-314
Operator RG-11
operators
  server attribute RG-305
opt_backfill_fuzzy
  scheduler attribute RG-314
order
  hook attribute RG-362
Output_Path
  job attribute RG-352
overall limit RG-11
owner RG-11

P
parameter RG-11
partition RG-329, RG-335
  scheduler attribute RG-315
pathname
  format RG-367
PBS RG-397
pbs RG-28
PBS administrator RG-2
PBS entity RG-5, RG-12
pbs Module RG-12
PBS NAME
  format RG-367
PBS object RG-11, RG-12
PBS password
  format RG-367
PBS Professional RG-12
PBS_ARRAY ID RG-397
PBS_ARRAY_INDEX RG-397
pbs_attach RG-55
pbs_comm RG-57
PBS_CONF_FILE RG-397
PBS_CPUTSET DEDICATED RG-397
pbs_dataservice RG-60
pbs_ds_password DEDICATED RG-397
PBS_ENVIRONMENT RG-397
PBS_EXEC RG-12
PBS_HOME RG-12
pbs_hostn RG-63
pbs_idled RG-64
pbs_interactive RG-67
PBS_JOBCOOKIE RG-397
PBS_JOBID RG-397
PBS_JOBNAME RG-397
pbs_lamboot RG-68
pbs_license_info
  server attribute RG-306
pbs_license_linger_time
  server attribute RG-306
pbs_license_max
  server attribute RG-306
pbs_license_min
  server attribute RG-306
pbs_migrate_users RG-69
pbs_mkdirs RG-71
pbs_mom RG-72
PBS_MOMPORT RG-397
pbs_mpihp RG-77
pbs_mpilam RG-79
pbs_mpirun RG-80
PBS_NODENUM RG-397
PBS_O_HOME RG-397
PBS_O_HOST RG-397
PBS_O_LANG RG-397
PBS_O_LOGNAME RG-398
PBS_O_MAIL RG-398
PBS_O_PATH RG-398
PBS_O_QUEUE RG-398
PBS_O_SHELL RG-398
PBS_O_SYSTEM RG-398
PBS_O_TZ RG-398
PBS_O_WORKDIR RG-398
pbs_password RG-82
pbs_probe RG-84
pbs_python RG-86
PBS QUEUE RG-398
pbs_rdel RG-92
pbs_rstat RG-95
pbs_rsub RG-97
pbs_sched RG-104
PBS_SERVER RG-398
pbs_server RG-106
PBS_TASKNUM RG-398
pbs_tclsh RG-116
PBS_TMPDIR RG-398
pbs_tmrsh RG-117
pbs_version
  scheduler attribute RG-315
  server attribute RG-306
  vnode attribute RG-335
pbs_wish RG-119, RG-121
pbsadmin RG-12
PBScrayhost RG-285
PBScraylabel RG-286
PBScraylabel RG-286
PBScrayord RG-286
PBScrayseg RG-286
Index

pbsdsh RG-29
pbsfs RG-31
pbshook RG-11
pbsnodes RG-35
pbsrun RG-40
pbsrun_unwrap RG-50
pbsrun_wrap RG-51
pcap_accelerator RG-352
pcap_node RG-353
pcpus
  vnode attribute RG-335
peput RG-286
Peer scheduling RG-12
pgov RG-353
p-governor RG-353
placement
  set RG-12
Placement pool RG-12
Placement set series RG-12
pmem RG-287
pnames
  vnode attribute RG-335
policy RG-12
  scheduling RG-15
Port
  vnode attribute RG-335
POSIX RG-12
power_provisioning
  server attribute RG-306
  vnode attribute RG-335
poweroff_eligible
  vnode attribute RG-335
PPU RG-12
preempt RG-13
preempt_checkpoint RG-270
preempt_fairshare RG-270
preempt_order RG-270
preempt_prio RG-271
preempt_queue_prio RG-271
preempt_requeue RG-271
preempt_sort RG-272
preempt_starving RG-272
preempt_suspend RG-272
preempt_targets RG-287
preemption
  level RG-13
  method RG-13
Preemption target RG-13
preemptive_sched RG-270
pre-execution event hooks RG-13
primary execution host RG-13
primary scheduler RG-13
primary server RG-13
prime_spill RG-272
primetime_prefix RG-272
printjob RG-122
Priority
  job attribute RG-353
queue attribute RG-329
  vnode attribute RG-335
Project RG-13
project
  job attribute RG-353
Project limit RG-13
project name
  format RG-367
provision RG-13
provision_enable
  vnode attribute RG-336
provision_policy RG-272
provisioned vnode RG-13
provisioning RG-376
  hook RG-13
provisioning tool RG-13
pset
  job attribute RG-353
pstate RG-353
pulling queue RG-13
pvmem RG-287
python_restart_max_hooks
  server attribute RG-307
python_restart_max_objects
  server attribute RG-307
python_restart_min_interval
  server attribute RG-307

Q
qalter RG-124
qdel RG-137
qdisable RG-140
qenable RG-142
qhold RG-144
qmgr RG-146, RG-380
qmove RG-167
qmsg RG-169
qorder RG-171
qrerun RG-173
qrls RG-175
qrun RG-177
qselect RG-181
qsig RG-187
qstart RG-190
qstat RG-192
qstop RG-205
qsub RG-207
qterm RG-226
qtime
Index

job attribute  RG-354
query_other_jobs  RG-307
server attribute  RG-308
queue
access to a  RG-1
definition  RG-14
execution  RG-6
furnishing  RG-6
job attribute  RG-354
pulling  RG-13
reservation attribute  RG-320
routing  RG-15
vnode attribute  RG-336
queue identifier
format  RG-367
queue name
format  RG-367
queue_rank
job attribute  RG-354
queue attribute  RG-330
queue_type
job attribute  RG-354
queue attribute  RG-330
queued_jobs_threshold
queue attribute  RG-329
queued_jobs_threshold_res
queue attribute  RG-330
server attribute  RG-308
queuing  RG-14
Quick Start Guide  RG-vi

R
redundant license server configuration  RG-14
reject an action  RG-14
release_nodes_on_stageout  RG-354
requeue  RG-14
require_cred
queue attribute  RG-330
server attribute  RG-308
require_cred_enable
queue attribute  RG-330
server attribute  RG-308
Rerunnable
job attribute  RG-355
reservation
access to a  RG-1
advance  RG-2
degradation  RG-14
degraded  RG-4
instance  RG-11
occurrence  RG-11
soonest occurrence  RG-16
standing  RG-16
instance  RG-11
soonest occurrence  RG-16
reservation degradation  RG-14
reservation name
format  RG-367
reserve_count
reservation attribute  RG-320
reservation_duration
reservation attribute  RG-320
reserve_end
reservation attribute  RG-320
reserve_ID
reservation attribute  RG-320
reserve_index
reservation attribute  RG-320
Reserve_Name
reservation attribute  RG-320
Reserve_Owner
reservation attribute  RG-321
reserve_retry
reservation attribute  RG-321
reserve_retry_cutoff
server attribute  RG-308
reserve_retry_init
server attribute  RG-308
reserve_rule
reservation attribute  RG-321
reserve_start
reservation attribute  RG-322
reserve_state
reservation attribute  RG-322
reserve_substate
reservation attribute  RG-322
Resource  RG-14
resource
built-in  RG-3
consumable  RG-4
custom  RG-4
indirect  RG-8
job-wide  RG-9
non-consumable  RG-11
shared  RG-15
Resource_List
job attribute  RG-355
reservation attribute  RG-323
Resource_List.eoe  RG-281
resource_unset_infinite  RG-273
resources  RG-273
resources_assigned
queue attribute  RG-330
server attribute  RG-309
vnode attribute  RG-336
resources_available
queue attribute  RG-331
Index

server attribute  RG-309  
  vnode attribute  RG-336  
  resources_available.eoe  RG-281  
  resources_default  
    queue attribute  RG-331  
    server attribute  RG-309  
  resources_max  
    queue attribute  RG-331  
    server attribute  RG-310  
  resources_min  
    queue attribute  RG-331  
    resources_used  
      job attribute  RG-356  
      restart  RG-14, RG-256  
      restart file  RG-14  
      restart script  RG-14  
      restrict_res_to_release_on_suspend  RG-310  
      resv  
        vnode attribute  RG-337  
        RESV_BEING_DELETED  RG-377  
        RESV_CONFIRMED  RG-377  
        RESV_DEGRADED  RG-377  
        RESV_DELETED  RG-377  
        RESV_DELETING_JOBS  RG-377  
        resv_enable  
          vnode attribute  RG-337  
          RESV_FINISHED  RG-377  
          resv_nodes  
            reservation attribute  RG-323  
            RESV_NONE  RG-377  
            resv_post_processing_time  
              server attribute  RG-310  
            RESV_RUNNING  RG-377  
            RESV_TIME_TO_RUN  RG-377  
            RESV_UNCONFIRMED  RG-377  
            RESV_WAIT  RG-377  
            resv-exclusive  RG-376  
            round_robin  RG-273  
            route  RG-14  
            route queue  RG-379, RG-382  
            route_destinations  
              queue attribute  RG-332  
              route_held_jobs  
                queue attribute  RG-332  
                route_lifetime  
                  queue attribute  RG-332  
                  route_retry_time  
                    queue attribute  RG-332  
                    route_waiting_jobs  
                      queue attribute  RG-332  
                      routing queue  RG-15  
                      rpp_highwater  
                      server attribute  RG-309  
                      rpp_max_pkt_check  RG-310  
                      rpp_retry  
                        server attribute  RG-310  
                        run_count  RG-134, RG-221  
                          job attribute  RG-356  
                          run_version  
                            job attribute  RG-356  
  S  
    sandbox  RG-221  
      job attribute  RG-356  
      sched_cycle_length  
        scheduler attribute  RG-315  
      sched_host  
        scheduler attribute  RG-315  
      sched_log  
        scheduler attribute  RG-315  
      sched_port  
        scheduler attribute  RG-315  
      sched_preempt_enforce_resumption  
        scheduler attribute  RG-315  
      sched_priv  
        scheduler attribute  RG-316  
      schedselect  
        job attribute  RG-356  
      Scheduler  RG-15  
        policies  RG-15  
        scheduler_iteration  
          scheduler attribute  RG-315  
          server attribute  RG-311  
      Scheduling  RG-374  
        scheduling  
          policy  RG-12, RG-15  
          scheduler attribute  RG-315  
          server attribute  RG-311  
      Schema Admins  RG-15  
      secondary scheduler  RG-15  
      secondary server  RG-15  
      sequence number  RG-15  
      Server  RG-15  
        access to the  RG-1  
        server  
          job attribute  RG-357  
          reservation attribute  RG-323  
          server_dyn_res  RG-274  
          server_softlimits  RG-271  
          server_state  
            server attribute  RG-312  
            session_id  
              job attribute  RG-357  
              set_power_cap  RG-353  
              shared resource  RG-15
Index

sharing
  vnode attribute RG-337
Shell_Path_List
  job attribute RG-357
shrink-to-fit job RG-15
single_signon_password_enable
  server attribute RG-312
sister RG-15
sisterhood RG-15
site RG-287
size
  format RG-368
smp_cluster_dist RG-274
Snapshot checkpoint RG-16
soft_walltime RG-288
software RG-287
soonest occurrence RG-16
sort_by RG-274
sort_priority RG-268
sort_queues RG-274
stage
  in RG-16
  out RG-16
stagein
  job attribute RG-357
stageout
  job attribute RG-358
Stageout_status
  job attribute RG-358
Staging and execution directory RG-16
stale RG-376
standing reservation RG-16
start_time RG-288
started
  queue attribute RG-332
starving_jobs RG-267, RG-271
state RG-16
  job RG-9
  scheduler attribute RG-316
  vnode attribute RG-339
state_count
  queue attribute RG-332
  server attribute RG-312
state-unknown, down RG-376
stime
  job attribute RG-358
Strict ordering RG-16
strict_fifo RG-274
strict_ordering RG-274
string resource value
  format RG-368
string_array
  format RG-368
subject RG-16
subjob RG-16
subjob identifier
  format RG-368
subjob index RG-16
Submit_arguments
  job attribute RG-358
subordinate MOM RG-17
substate
  job attribute RG-358
sw_index
  job attribute RG-358
sync_time RG-275
T
  task RG-17
  task placement RG-17
terminate RG-256
Terminating RG-374
Terminating_Delayed RG-374
Three-server Configuration RG-17
throughput_mode
  scheduler attribute RG-316
time-sharing RG-379, RG-380, RG-381
token RG-17
topjob_ineligible
  job attribute RG-358
topology_info
  vnode attribute RG-339
total_jobs
  queue attribute RG-332
  server attribute RG-313
TPP RG-17
tracejob RG-228
Type
  hook attribute RG-362
U
  UID RG-17
umask
  job attribute RG-359
unknown_shares RG-275
user
  access RG-17
  definition RG-17
  hook attribute RG-362
  ID RG-17
User Guide RG-vi
user limit RG-17
User_List
  job attribute RG-359
username
  format RG-368
  Windows
Index

format RG-369

V
Variable_List
  job attribute RG-359
vchunk RG-17
Version 1 configuration file RG-17
Version 2 configuration file RG-18
vmem RG-288
vnode RG-18, RG-288
  borrowing RG-2
  managing RG-10
  memory-only RG-10
vnode name
  format RG-369
vnode_pool RG-340
vnodedefs file RG-18
vntype RG-288
vp RG-18

W
wait-provisioning RG-376
walltime RG-289

X
xpbs RG-232
xpbsmon RG-243