



# ALTAIR

Altair PBS Professional 2021.1.2

Installation & Upgrade Guide

You are reading the Altair PBS Professional 2021.1.2

## **Installation & Upgrade Guide (IG)**

Updated 9/20/21

Copyright © 2003-2021 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®", and "PBS Pro™", "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](mailto: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](http://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](http://www.pbsworks.com)

### **Sales**

[pbssales@altair.com](mailto:pbssales@altair.com) 248.614.2400

Please send any questions or suggestions for improvements to [agu@altair.com](mailto:agu@altair.com).

## Technical Support

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

Location	Telephone	e-mail
Australia	+1 800 174 396	anz-pbssupport@india.altair.com
China	+86 (0)21 6117 1666	pbs@altair.com.cn
France	+33 (0)1 4133 0992	pbssupport@europe.altair.com
Germany	+49 (0)7031 6208 22	pbssupport@europe.altair.com
India	+91 80 66 29 4500 +1 800 208 9234 (Toll Free)	pbs-support@india.altair.com
Italy	+39 800 905595	pbssupport@europe.altair.com
Japan	+81 3 6225 5821	pbs@altairjp.co.jp
Korea	+82 70 4050 9200	support@altair.co.kr
Malaysia	+91 80 66 29 4500 +1 800 425 0234 (Toll Free)	pbs-support@india.altair.com
North America	+1 248 614 2425	pbssupport@altair.com
Russia	+49 7031 6208 22	pbssupport@europe.altair.com
Scandinavia	+46 (0)46 460 2828	pbssupport@europe.altair.com
Singapore	+91 80 66 29 4500 +1 800 425 0234 (Toll Free)	pbs-support@india.altair.com
South Africa	+27 21 831 1500	pbssupport@europe.altair.com
South America	+55 11 3884 0414	br_support@altair.com
UK	+44 (0)1926 468 600	pbssupport@europe.altair.com



# Contents

About PBS Documentation	vii
<b>1 PBS Architecture</b>	<b>1</b>
1.1 What is PBS?	1
1.2 PBS Daemons	1
1.3 PBS Commands	4
1.4 Scheduling Jobs	5
<b>2 Pre-Installation Steps</b>	<b>7</b>
2.1 Prerequisites for Running PBS	7
2.2 Important Considerations	12
2.3 PBS Configurations for Windows	13
<b>3 Installation</b>	<b>19</b>
3.1 Overview of Installation	19
3.2 Licenses	19
3.3 Major Steps for Installing PBS Professional	20
3.4 All Installations	20
3.5 Installing via RPM on Linux Systems	23
3.6 Installing via dpkg on Ubuntu	37
3.7 Installing PBS on Windows Hosts	37
<b>4 Communication</b>	<b>45</b>
4.1 Communication Within a PBS Complex	45
4.2 Terminology	45
4.3 Prerequisites	45
4.4 Communication Parameters	45
4.5 Inter-daemon Communication Using TPP	49
4.6 Ports Used by PBS	58
4.7 PBS with Multihomed Systems	60
<b>5 Initial Configuration</b>	<b>63</b>
5.1 Validate the Installation	63
5.2 Support PBS Features	63

## Contents

---

<b>6</b>	<b>Upgrading</b>	<b>65</b>
6.1	Types of Upgrades	65
6.2	Differences from Previous Versions	66
6.3	Caveats and Advice	67
6.4	Introduction to Upgrading Under Linux	70
6.5	Overlay Upgrade Under Linux	70
6.6	Overlay Upgrade on One or More Machines Running Cpuset MoM	82
6.7	Migration Upgrade Under Linux	93
6.8	Upgrading a Windows/Linux Complex	109
6.9	Upgrading from an All-Windows Complex	125
6.10	After Upgrading	137
<b>7</b>	<b>Installing and Upgrading on Cray</b>	<b>139</b>
7.1	Installing PBS with Shasta	139
7.2	Prerequisites for Using Power Profiles with Cray XC	139
7.3	Support for IMPS and CLE 6 and 7	139
7.4	Caveats and Advice for Installing and Upgrading on the Cray XC	140
7.5	Installing PBS on the Cray XC	142
7.6	After Installing on the Cray XC	152
7.7	Upgrading on the Cray XC	153
7.8	After Upgrading on Cray XC	157
7.9	Cray XC References	158
<b>8</b>	<b>Starting &amp; Stopping PBS on Linux</b>	<b>159</b>
8.1	Automatic Start on Bootup	159
8.2	When to Restart PBS Daemons	159
8.3	Methods for Starting, Stopping, or Restarting PBS	160
8.4	Starting, Stopping, and Restarting PBS Daemons	163
8.5	Impact of Stop-Restart on Running Linux Jobs	169
<b>9</b>	<b>Starting &amp; Stopping MoM on Windows</b>	<b>171</b>
9.1	Automatic Start on Bootup	171
9.2	When to Restart PBS MoMs	171
9.3	Starting, Stopping, and Restarting PBS	171
9.4	Stopping PBS Using the <code>qterm</code> Command	174
9.5	Impact of Stop-Restart on Running Windows Jobs	174
	<b>Index</b>	<b>177</b>

# 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**

Optional arguments are enclosed in square brackets. For example, in the `qstat` man page, the `-E` option is shown this way:

`qstat [-E]`

## About PBS Documentation

---

To use this option, you would type:

```
qstat -E
```

### Variable Arguments

Variable arguments (where you fill in the variable with the actual value) such as a job ID or vnode name are enclosed in angle brackets. Here's an example from the `pbsnodes` man page:

```
pbsnodes -v <vnode>
```

To use this command on a vnode named "my\_vnode", you'd type:

```
pbsnodes -v my_vnode
```

### Optional Variables

Optional variables are enclosed in angle brackets inside square brackets. In this example from the `qstat` man page, the job ID is optional:

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

To query the job named "1234@my\_server", you would type this:

```
qstat 1234@my_server
```

### Literal Terms

Literal terms appear exactly as they should be used. For example, to get the version for a command, you type the command, then "--version". Here's the syntax:

```
qstat --version
```

And here's how you would use it:

```
qstat --version
```

### Multiple Alternative Choices

When there are multiple options and you should choose one, the options are enclosed in curly braces. For example, if you can use either "-n" or "--name":

```
{-n | --name}
```

## List of PBS Professional Documentation

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

### *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 Big Book*

All your favorite PBS guides in one place: *Installation & Upgrade*, *Administrator's*, *Hooks*, *Reference*, *User's*, *Programmer's*, *Cloud*, *Budget*, and *Simulate* guides in a single book.

### *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*



## About PBS Documentation

---

How to write and use hooks for PBS Professional. For the PBS administrator.

### *PBS Professional Reference Guide*

Covers PBS reference material: the PBS commands, resource, attributes, configuration files, etc.

### *PBS Professional User's Guide*

How to submit, monitor, track, delete, and manipulate jobs. For the job submitter.

### *PBS Professional Programmer's Guide*

Discusses the PBS application programming interface (API). For integrators.

### *PBS Professional Manual Pages*

PBS commands, resources, attributes, APIs.

### *PBS Professional Licensing Guide*

How to configure licensing for PBS Professional. For the PBS administrator.

### *PBS Professional Cloud Guide*

How to configure and use the PBS Professional Cloud feature.

### *PBS Professional Budgets Guide*

How to configure Budget and use it to track and manage resource usage by PBS jobs.

### *PBS Professional Simulate Guide*

How to configure and use the PBS Professional Simulate feature.

## 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](mailto:pbssales@altair.com).

## About PBS Documentation

---

# PBS Architecture

## 1.1 What is PBS?

PBS Professional is a distributed workload management system for managing and monitoring your computational workload. PBS consists of daemons and commands that you use to manage jobs on one or more machines. You can use PBS to do tasks such as submitting, querying, altering, monitoring, moving, and deleting jobs.

## 1.2 PBS Daemons

You use one PBS server to manage a group of machines. The server coordinates with one or more schedulers to schedule where and when jobs run. Each machine where jobs run is managed by a MoM. Communication between server, schedulers, and MoMs is handled by one or more communication daemons. We call each instance of server, schedulers, MoMs, and communication daemons a *PBS complex*.

PBS daemons live in `PBS_EXEC/sbin`.

### 1.2.1 Server

The PBS server receives incoming job submissions, holds jobs that are waiting for execution, sends jobs for execution when it's their turn, and ensures that work is completed by monitoring the complex for failures and rerunning jobs when necessary. Commands communicate with the server, even if they affect other daemons. The server executable is named `pbs_server`; it is located in `$PBS_EXEC/sbin/pbs_server`.

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

For more about the server, see ["Configuring the Server and Queues" on page 19 in the PBS Professional Administrator's Guide](#).

### 1.2.2 Schedulers

PBS has a default scheduler; if you want to schedule individual partitions separately, you can add any number of additional schedulers, called *multischeds*. Each PBS scheduler follows its own scheduling policy.

Each scheduler daemon implements a policy that you define that controls when each job is run and on which resources. See ["About Schedulers" on page 91 in the PBS Professional Administrator's Guide](#).

### 1.2.3 MoM

The MoM daemon places each job into execution when it receives a copy of the job from the server. MoM creates a new session that is as identical to a user login session as is possible. For example, if the user's login shell is `csh`, then MoM creates a session in which `.login` is run as well as `.cshrc`. MoM also returns the job's output to the user. One MoM runs on each computer executing PBS jobs. These computers are called *execution hosts*.

For a complete description of configuring MoM, see ["Configuring MoMs and Vnodes" on page 35 in the PBS Professional Administrator's Guide](#).

## 1.2.4 Communication Daemon

The *communication daemon*, `pbs_comm`, handles communication between the other PBS daemons. For a complete description, see [section 4.5, "Inter-daemon Communication Using TPP", on page 49](#).

## 1.2.5 Typical Daemon Placements

### 1.2.5.1 Linux Layouts

The PBS server, scheduler, and communication daemons run on a Linux host. One or more communication daemons run on other Linux hosts, if there are enough MoMs in the complex to require additional comm daemons. Typical layouts:

- A cluster of MPI-connected execution hosts where each host runs a MoM
- One or more Cray computers
- One or more HPE execution hosts, where each host is managed by a MoM and is made up of multiple blades
- Individual execution hosts on a network
- Any or all of the above

### 1.2.5.2 Windows Layouts

#### 1.2.5.2.i Linux-Windows Complex

A Linux-Windows complex has a Linux server/scheduler/communication host and Windows execution and client hosts.

#### 1.2.5.2.ii Mixed-mode Complex

A mixed-mode complex has a Linux server/scheduler/communication host, Linux execution and client hosts, and Windows execution and client hosts.

### 1.2.6 Single Execution System

You can install and run all PBS components on a single machine. The following illustration shows how communication works when PBS is on a single host:

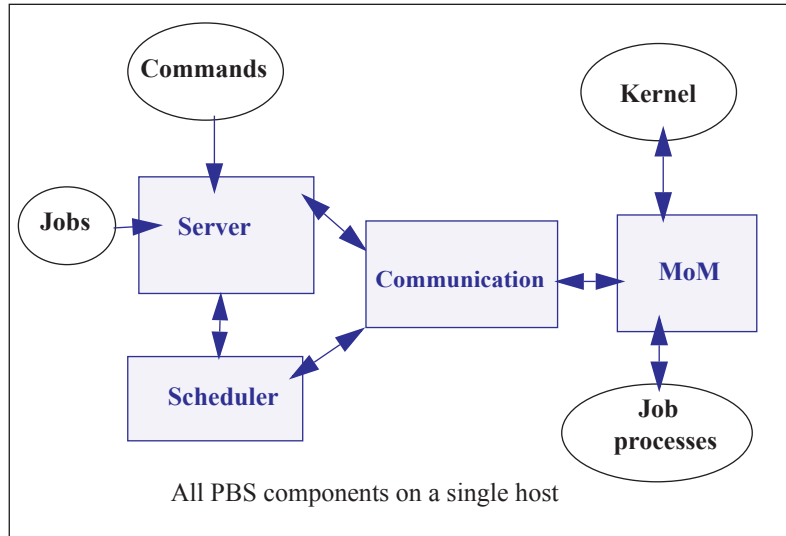


Figure 1-1:PBS daemons on a single execution host

### 1.2.7 Single Execution System with Front End

The PBS server and scheduler (pbs\_server and pbs\_sched) can run on one system and jobs can execute on another. 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:

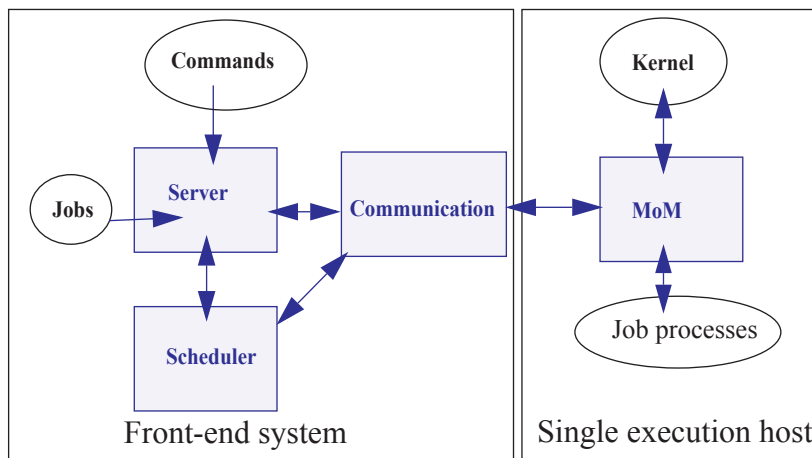


Figure 1-2:PBS daemons on single execution system with front end

## 1.2.8 Multiple Execution Systems

When you run PBS on several systems, the server (`pbs_server`), the scheduler (`pbs_sched`), and the communication daemon (`pbs_comm`) are installed on a front end system, and a MoM (`pbs_mom`) is installed and run on each execution host. The following diagram illustrates this:

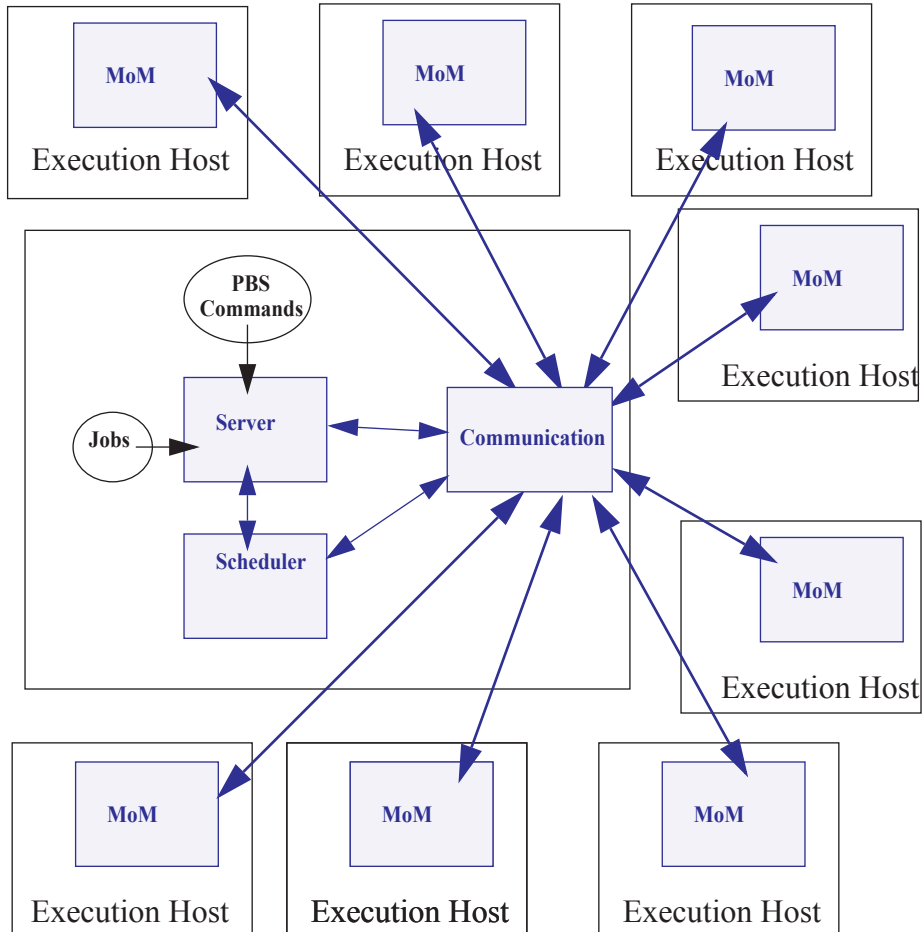


Figure 1-3: Typical PBS daemon locations for multiple execution hosts

## 1.3 PBS Commands

PBS supplies command-line client commands that 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.

The privilege required to run each command varies with that command; see each command's description. PBS commands are described in ["PBS Commands" on page 21 of the PBS Professional Reference Guide](#).

---

## 1.4 Scheduling Jobs

PBS runs jobs only on the execution hosts in the complex (hosts running a MoM). Each job is placed on a host or hosts according to the job's request. The scheduler matches jobs with available resources such as CPUs, memory, required software, licenses, etc. The scheduler follows rules for selecting hosts and parts of hosts that match each job's request. Once the scheduler finds the resources that match a job's request, it allocates hosts or parts of hosts to the job, according to how the host is configured and what the job requested.

Each task from a job can be placed on a different host, or a different part of a host. Alternatively, all tasks can be run on a single host. The job can request exclusive use of each host or part of a host, or shared use with other jobs. For details, see ["Specifying Job Placement", on page 64 of the PBS Professional User's Guide](#).

Each scheduler can be configured so that it follows its own scheduling policy. Scheduling policy dictates which jobs are allowed to run where, who can use how much of what, etc. See ["Scheduling" on page 55 in the PBS Professional Administrator's Guide](#).





# Pre-Installation Steps

This chapter describes the steps to take before installing PBS. Make sure that your setup meets the requirements described here, and that you take the required steps to prepare for installing PBS.

## 2.1 Prerequisites for Running PBS

### 2.1.1 Run Same Version Within Complex

Do not mix different versions of PBS within a PBS complex. All machines using a particular PBS server (all machines in the same PBS complex) must run the exact same version of PBS, except for platform differences. Do not mix major, minor, or patch versions for any element of PBS such as daemons or commands. For example, do not run 2020.1.1 and 2020.1.2 in the same complex.

Do not mix different versions of PBS across PBS complexes, either.

### 2.1.2 Resources Required by PBS

The amount of memory required by the PBS server and scheduler depends on the number of hosts and the number of jobs to be queued or running. You will need less than 512 bytes per host. The number of jobs is the important factor, since each job needs about 10 KB at server startup and 5 KB when the server is running. The number of processors in the complex is not a factor.

#### 2.1.2.1 Memory Required By Server Running Hooks

A PBS server executing hook scripts can consume a larger amount of memory than one not executing hook scripts. For example, a system consisting of a server and a MoM on a Linux machine handling 10,000 short-running jobs being submitted, modified, and moved causing execution of qsub, qalter, and movejob hooks will use around 40 MB of memory in a span of 24 hours.

#### 2.1.2.2 Memory Required for Job History

Enabling job history requires additional memory for the server. When the server is keeping job history, it needs 8k-12k of memory per job, instead of the 5k it needs without job history. Make sure you have enough memory: multiply the number of jobs being tracked by this much memory. For example, if you are starting 100 jobs per day, and tracking history for two weeks, you're tracking 1400 jobs at a time. On average, this will require 14.3M of memory.

If the server is shut down abruptly, there is no loss of job information. However, the server will require longer to start up when keeping job history, because it must read in more information.

#### 2.1.2.3 Amount of Memory in Complex

If the sum of all memory on all vnodes in a PBS complex is greater than 2 terabytes, then the server (`pbs_server`) and scheduler (`pbs_sched`) must be run on a 64-bit architecture host, using a 64-bit binary.

### 2.1.2.4 Adequate Space for Logfiles

PBS logging can fill up a filesystem. For customers running a large number of array jobs, we recommend that the filesystem where `$PBS_HOME` is located has at least 2 GB of free space for log files. It may also be necessary to rotate and archive log files frequently to ensure that adequate space remains available. (A typical PBS Professional complex will generate about 2 GB of log files for every 1,000,000 subjobs and/or jobs.)

### 2.1.2.5 Installation Disk Space

Make sure you have adequate disk space to install PBS. It is recommended to have at least 350 MB available, for installation alone.

### 2.1.2.6 Disk and Memory for Communication Daemon

By default, the communication daemon is installed on the server host.

Disk space used by the communication daemon is only for logfiles; make sure that your logging does not fill up the disk.

On any host running a communication daemon handling up to 5000 MoMs, make sure you have 500MB to 1GB of memory for the daemon.

### 2.1.2.7 Memory for Data Store

The data store itself requires around 100MB, but its size depends on the amount of memory required to store each job script. The total memory required is the size of all job scripts plus 100MB.

## 2.1.3 Name Resolution and Network Configuration

**Do NOT skip this section.** PBS cannot function if your hostname resolution or network is configured incorrectly.

### 2.1.3.1 Firewalls

PBS needs to be able to use any port for outgoing connections, but only specific ports for incoming connections. If you have firewalls running on the server or execution hosts, be sure to allow incoming connections on the appropriate ports for each host. By default, the PBS server and MoM daemons use ports 15001 through 15004 for incoming connections, the PBS communication daemon listens on port 17001, and daemons use any port below 1024 for outgoing connections. See [section 4.6, "Ports Used by PBS", on page 58](#) for a list of ports.

Firewall-based issues are often associated with server-MoM communication failures and messages such as 'premature end of message' in the log files.

To allow interactive jobs, make sure that the ephemeral port range in your firewall is open (make sure that MoMs can connect to an ephemeral port on submission hosts). Check your OS documentation for the correct range.

### 2.1.3.2 Network Tuning

Depending on your network, you may need to tune kernel settings or other configuration parameters. Make sure that your kernel settings support PBS. For example, check your IP tuning parameters, including UDP and TCP, and check your ARP, routing, and name resolution settings.

---

### 2.1.3.3 Planning for Number of Machines Connected to Complex

Configure your server host with sufficient ARP cache entries in order to allow at least one connection per ethernet address that will connect to the server or to which the server will connect. This includes execution hosts, client hosts, peered servers, storage machines, or machines where the scheduler may execute scripts. Check your ARP table tuning settings.

### 2.1.3.4 Required Name Resolution

Make sure that the following are true:

- Use only one canonical name per host. The canonical name must be unambiguous.
- On the server/scheduler/communication host, the short name must resolve to the correct IP address.
- On the server/scheduler/communication host, the IP address must reverse resolve to the canonical name.
- Make sure that different resolvers cannot disagree when resolving the server host, whether you are using `/etc/hosts`, DNS, LDAP, NIS, or something else.
- Every MoM must resolve each MoM to the same IP address that the server recognizes for that MoM. So if the server recognizes MoM A at IP address w.x.y.z, all other MoMs must resolve MoM A to w.x.y.z.
- Make sure that the IP address of each machine in the complex resolves to the fully qualified domain name for that machine, and vice versa. Forward and reverse hostname resolution must work consistently between all machines.
- The server must be able to look up the IP addresses for any execution host, any client host, and itself.
- Make sure that forward and reverse name lookup operate according to the IETF standard. The network on which you will be deploying PBS must be configured according to IETF standards.

### 2.1.3.5 Required Network Configuration

- PBS can use a static address mapping only.
- Communications between daemons must be robust and must have sufficient capacity. Make sure that your network does not present any limitations to PBS. For example, the ARP table size limit must not interfere when you have a large number of MoMs. Configure your server with sufficient ARP cache entries to allow at least one connection per ethernet address that will connect to the server or to which the server will connect. This includes execution hosts, client hosts, peered servers, storage machines, or machines where the scheduler may execute scripts. See [section 2.1.3.1, "Firewalls", on page 8](#).

### 2.1.3.6 Recommendations for Name Resolution and Network Configuration

- Test name resolution using the `ping` command.
- Test the connections between server and MoM daemons on every physical network. You should test TCP and UDP, and make sure that the connection can handle large packets. You can use a tool such as `ttcp`, with packets of size 16k, for testing.
- For multihomed MoMs, keep all PBS traffic on the same control network or subnet.
- Keep different types of traffic on separate interfaces to reduce jitter.
- When configuring `/etc/hosts`, do the following:
  - Use the server's FQDN as the first item on the first line on the PBS-to-PBS interface
  - Use different FQDNs as the first item on other lines
  - Use a name on only one line
- If you want redundancy in your network interface, consider using bonding. Aside from presenting a transparent interface, this can allow you to load-balance network traffic across different networks.
- If name resolution is a problem in a network that should be working, tell `nscd` not to cache the host name of the machine with the problem.
- If you are using `nscd` and you change an IP address or hostname, restart `nscd` on all hosts.

#### 2.1.3.6.i Recommendations for Name Resolution and Network Configuration on Windows

- On Windows, make sure the first nameserver resolves all the needed hostnames, including the server hostname and the domain controller host for active directory queries.
- On Windows, put explicit IP-to-hostname addresses in the `C:\windows\system32\drivers\etc\hosts` file. Otherwise your site will experience extreme slowdowns. If you make these changes to a running PBS complex, you must then restart all the PBS daemons (services).

### 2.1.3.7 Order of Operations for Name Resolution and Network Configuration

You can take care of some of the name resolution testing before you install PBS. However, you must do some testing using the `pbs_hostn` command, after you install PBS. The "[Initial Configuration](#)" chapter follows the "[Installation](#)" chapter, and includes steps to test name resolution. We include an overview of the whole process here for clarity:

1. Set up firewall
2. Set up name resolution
3. Test name resolution by using `ping` command; if necessary, fix & re-test
4. Install PBS
5. Test name resolution by using `pbs_hostn` command.
6. If name resolution does not work correctly:
  - a. Uninstall PBS
  - b. Fix name resolution
  - c. Install PBS
  - d. Test using `pbs_hostn`

---

### 2.1.3.8 Server Hostname

The `PBS_SERVER` entry in `pbs.conf` cannot be longer than 255 characters. If the short name of the server host resolves to the correct IP address, you can use the short name for the value of the `PBS_SERVER` entry in `pbs.conf`. If only the FQDN of the server host resolves to the correct IP address, you must use the FQDN for the value of `PBS_SERVER`.

### 2.1.3.9 Sockets

Some PBS processes cause network sockets to be opened between submission and execution hosts. For more information about these processes, see ["Sockets and Checkpointing" on page 433 in the PBS Professional Administrator's Guide](#). Make sure your network and firewalls are set up to handle sockets correctly.

### 2.1.3.10 Mounting NFS File Systems

Asynchronous writes to an NFS server can cause reliability problems. If using an NFS file system, mount the NFS file system synchronously (without caching.)

### 2.1.3.11 Making Ports Available

The ports used by the PBS daemons must be available during the installation. See [section 4.6, "Ports Used by PBS", on page 58](#).

## 2.1.4 HPE Prerequisites

### 2.1.4.1 HPE MPI Recommendation

For HPE MC990X, HPE Superdome Flex, and HPE 8600 machines, we recommend using HPE MPI.

As of PBS version 2020.1, `pbs_mom.cpuset` is no longer available. Instead, use standard MoM, and use the `cgroups` hook to manage `cgroups`.

### 2.1.4.2 Power File Requirement

When using PBS Power Provisioning on HPE, ensure that the following file exists:

```
/opt/clmgr/power-service
```

## 2.1.5 License Server Requirement

Make sure that the ALM license server is at version 14.5 before installing PBS.

## 2.1.6 System Clocks in Sync

We recommend that clocks on all participating systems be in sync.

## 2.1.7 User Requirements on Linux

### 2.1.7.1 User Accounts

Users who will submit jobs must have accounts at the server and at each execution host.

### 2.1.7.2 Linux User Authorization

When the user submits a job from a system other than the one on which the PBS server is running, system-level user authorization is required. This authorization is needed for submitting the job and for PBS to return output files (see also ["Managing Output and Error Files", on page 40 of the PBS Professional User's Guide](#) and ["Input/Output File Staging", on page 31 of the PBS Professional User's Guide](#)).

The username under which the job is to be executed is selected according to the rules listed under the "-u" option to `qsub`. The user submitting the job must be authorized to run the job under the execution user name (whether explicitly specified or not).

Such authorization is provided by any of the following methods:

1. The host on which `qsub` is run (i.e. the submission host) is trusted by the server. This permission may be granted at the system level by having the submission host as one of the entries in the server's `hosts.equiv` file naming the submission host. For file delivery and file staging, the host representing the source of the file must be in the receiving host's `hosts.equiv` file. Such entries require system administrator access.
2. The host on which `qsub` is run (i.e. the submission host) is explicitly trusted by the server via the user's `.rhosts` file in his/her home directory. The `.rhosts` must contain an entry for the system from which the job is submitted, with the user name portion set to the name under which the job will run. For file delivery and file staging, the host representing the source of the file must be in the user's `.rhosts` file on the receiving host. It is recommended to have two lines per host, one with just the "base" host name and one with the full hostname, e.g.: `host.domain.name`.
3. PBS may be configured to use the Secure Copy (`sftp`) for file transfers. The administrator sets up SSH keys as described in ["Enabling Passwordless Authentication" on page 567 in the PBS Professional Administrator's Guide](#). See also ["Setting File Transfer Mechanism" on page 561 in the PBS Professional Administrator's Guide](#).
4. User authentication may also be enabled by setting the server's `flatuid` attribute to `True`. See the `pbs_server_attributes(7B)` man page and ["Flatuid and Access" on page 378 in the PBS Professional Administrator's Guide](#). Note that `flatuid` may open a security hole in the case where a vnode has been logged into by someone impersonating a genuine user.

## 2.2 Important Considerations

### 2.2.1 Avoiding Datastore Corruption from Job Spool Files

Job spool files can fill up the `PBS_HOME` filesystem. This can corrupt the datastore and cause a failure that requires recovering from backups. Consider moving the spool directory to a dedicated file system, or using quotas. Note that if you use quotas on Cray ext3fs filesystems, the kernel enforces quotas correctly even though commands that display quotas may show stale data (when files are not closed.)

Job spool files are saved on the server on job rerun, and on the MoM for running jobs.

### 2.2.2 Using `noexec` on `/tmp`

If you need to have `noexec` on your `/tmp`, do one of the following:

- Set the `TMPDIR` environment variable; the shared library that is extracted to `/tmp/xf-dll` follows `TMPDIR` if it is set
- Install a soft link from `/tmp/xf-dll` pointing to a location on a filesystem that does not have the "noexec" mount flag

---

Why? The `ALSDK liblms-altair.so` self-extracts a DSO into `/tmp/xf-dll`, and then tries to map it. If it fails to do so because `noexec` is set, the `ALSDK` routines simply perform an `exit(1)`, which terminates the server, without any log message in the server log.

## 2.3 PBS Configurations for Windows

### 2.3.1 Definitions

#### Active Directory

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. Active Directory is fully integrated with DNS and TCP/IP; DNS is required. To be fully functional, the DNS server must support SRV resource records or service records.

#### Admin (Windows)

As referred to in various parts of this document, this is a user logged in from an account who is a member of any group that has full control over the local computer, domain controller, or is allowed to make domain and schema changes to the Active directory.

#### Administrators

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

#### Delegation

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, then delegation allows giving the account read access only to all domain users and groups information. This is done via the Delegation wizard.

#### Domain Admin Account

This is a domain account on Windows that is a member of the "Domain Admins" group.

#### Domain Admins

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

It is a domain account on Windows that is a member of the "Domain Users" group.

#### Domain Users

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.

#### Enterprise Admins

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.

#### Install Account, Installation Account

The account used by the person who installs PBS.



### Schema Admins

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.

### PBS service account

The account that is used to execute `pbs_mom` via the Service Control Manager on Windows. This account can have any name. The default name is `pbsadmin`.

## 2.3.2 Domained Environment Required

All Windows hosts and users must be in a dominated environment.

## 2.3.3 Permission Requirement

On Windows 7 and later with UAC enabled, if you will use the `cmd` prompt to operate on hooks, or for any privileged command such as `qmgr`, you must run the `cmd` prompt with option *Run as Administrator*.

## 2.3.4 Daemon Layout for Windows

As of PBS 19.4.1, all PBS complexes run the PBS server, scheduler, and comm daemons on Linux hosts. You can run all MoMs and client commands on Windows hosts, or some on Windows and some on Linux.

## 2.3.5 Windows Configuration in a Domained Environment

### 2.3.5.1 Machines

- Any Windows client commands and MoMs must run on a set of Windows machines networked in a single domain.
- The machines must be members of this one domain, and they must be dependent on a centralized database located on the primary/secondary domain controllers.
- The domain controllers must be running on a Server type of Windows host, using Active Directory configured in "native" mode.
- The choice of DNS must be compatible with Active Directory.
- The PBS server and scheduler run on a Linux host.
- PBS must not be installed or run on a Windows machine that is serving as the domain controller (running Active Directory) to the PBS hosts.

### 2.3.5.2 User Accounts

- Windows job submitters must have an account at all PBS hosts involved in a job: the server, the execution hosts, and the client host.
- All user accounts must be in the same domain as the Windows client and execution hosts.
- Each user must explicitly be assigned a HomeDirectory sitting on some network path. PBS does not support a HomeDirectory that is not network-mounted. PBS currently supports network-mounted directories that are using the Windows network share facility.
- If a user was not assigned a HomeDirectory, then PBS uses `PROFILE_PATH\My Documents\PBS Pro`, where `PROFILE_PATH` could be, for example, `"\Documents and Settings\username"`.



### 2.3.5.3 User Jobs

- All users must submit and run PBS jobs using only their domain accounts (no local accounts), and domain groups. If a user has both a domain account and local account, then PBS will ensure that the job runs under the domain account.
- Each user must always supply an initial password in order to submit jobs. This is done by running the [pbs\\_login](#) command at least once to supply the password that PBS will use to run the user's jobs.
- Access by jobs to network resources, such as a network drive, requires a password.
- All job scripts, as well as input, output, error, and intermediate files of a PBS job must reside in an NTFS directory.

### 2.3.6 User Authorization Under Windows

Windows job submitters must cache a password for authorization. To do this, each job submitter must run [pbs\\_login](#) at each client host initially and for each password change.

The user name under which the job is to be executed is selected according to the rules listed under the "-u" option to `qsub`. See ["qsub" on page 214 of the PBS Professional Reference Guide](#). The user submitting the job must be authorized to run the job under the execution user name (whether explicitly specified or not). Authorization is provided by either of the following methods:

#### 2.3.6.1 Requirements for Non-admin Users

Under Windows, if a user has a non-admin account, the server `hosts.equiv` file is used to determine whether that user can run a job at a given server.

The Windows `hosts.equiv` file determines the list of non-Administrator accounts that are allowed access to the local host, that is, the host containing this file. This file also determines whether a remote user is allowed to submit jobs to the local PBS server, with the user on the local host being a non-Administrator account.

This file is usually: `%WINDIR%\system32\drivers\etc\hosts.equiv`.

The format of the `hosts.equiv` file is as follows:

`[+|-] hostname username`

'+' means enable access, whereas '-' means to disable access. If '+' or '-' is not specified, then this implies enabling of access. If only `hostname` is given, then users logged into that host are allowed access to like-named accounts on the local host. If only `username` is given, then that user has access to all accounts (except Administrator-type users) on the local host. Finally, if both `hostname` and `username` are given, then user at that host has access to like-named account on local host.

The `hosts.equiv` file must be owned by an admin-type user or group, with write access granted to an admin-type user or group.

**Table 2-1: Requirements for Non-admin User to Submit Job**

File	Submission Host Username vs. Server Host Username	
	UserS Same as UserS	UserS Different from UserA
<code>hosts.equiv</code> on ServerA	<HostS>	<HostS> UserS

### 2.3.6.2 Requirements for Admin Users

For an admin account, `[PROFILE_PATH] . \rhosts` is used, and the server's `acl_roots` attribute must be set to allow job submissions.

**Table 2-2: Requirements for Admin User to Submit Job**

Location/Action	Submission Host Username vs. Server Host Username	
	UserS Same as UserS	UserS Different from UserA
<code>[PROFILE_PATH] \rhosts</code> contains	For UserS on ServerA, add <code>&lt;HostS&gt; UserS</code>	For UserA on ServerA, add <code>&lt;HostS&gt; UserS</code>
set ServerA's <code>acl_roots</code> attribute	<code>qmgr&gt; set server acl_roots=UserS</code>	<code>qmgr&gt; set server acl_roots=UserA</code>

### 2.3.7 Windows User HOMEDIR

Each Windows user must have a home directory (`HOMEDIR`) where their PBS job will initially be started. For jobs that do not have their staging and execution directories created by PBS, the home directory is also the starting location of file transfers when users specify relative path arguments to `qsub/qalter -W stagein/stageout` options.

PBS supports network mounted home directories.

#### 2.3.7.1 Configuring User HOMEDIR

The home directory can be configured by an Administrator by setting the user's `HomeDirectory` field in the user database, via the User Management Tool. It is important to include the drive letter when specifying the home directory path. The directory specified for the home folder must be accessible to the user. If the directory has incorrect permissions, PBS will be unable to run jobs for the user.

#### 2.3.7.2 Directory Must Exist Already

You must specify an already existing directory for home folder. If you don't, the system will create it for you, but set the permissions to that which will make it inaccessible to the user.

#### 2.3.7.3 Default Directory

If a user has not been explicitly assigned a home directory, then PBS will use this Windows-assigned default, local home directory as base location for its default home directory. More specifically, the actual home path will be:

```
[PROFILE_PATH] \My Documents \PBS Pro
```

For instance, if a `userA` has not been assigned a home directory, it will default to a local home directory of:

```
\Documents and Settings \userA \My Documents \PBS Pro
```

UserA's job will use the above path as working directory, and for jobs that do not have their staging and execution directories created by PBS, any relative pathnames in `stagein`, `stageout`, `output`, `error` file delivery will resolve to the above path.

---

Note that Windows can return as `PROFILE_PATH` one of the following forms:

```
\Documents and Settings\username
\Documents and Settings\username.local-hostname
\Documents and Settings\username.local-hostname.00N where N is a number
\Documents and Settings\username.domain-name
```

## 2.3.8 Windows Caveats

### 2.3.8.1 Installation of Microsoft Redistributable Pack

The PBS installer installs the Microsoft redistributable pack of `vc++` redistributable binaries into the system root (`C:\Windows`) directory.

### 2.3.8.2 Make Sure ComSpec Environment Variable Is Set

Check that in the `pbs_environment` file, the environment variable `ComSpec` is set to `C:\WINDOWS\system32\cmd.exe`. If it is not, set it to that value:

1. Change directory:  

```
cmd.admin> cd \Program Files\PBS\home
```
2. Edit the `pbs_environment` file:  

```
cmd.admin> edit pbs_environment
```
3. Add the following entry to the `pbs_environment` file:  

```
ComSpec=C:\WINDOWS\system32\cmd.exe
```
4. Restart the MoM:  

```
net stop pbs_mom
net start pbs_mom
```

Simply setting this variable inside a job script doesn't work. The `ComSpec` variable must be set before PBS executes `cmd`. `cmd` invokes the user's submission script.

### 2.3.8.3 Unsupported Windows Configurations

The following Windows configurations are currently unsupported:

- Using NIS/NIS+ for authentication on non-domain accounts.
- Using RSA SecurID module with Windows logons as a means of authenticating non-domain accounts.



# Installation

## 3.1 Overview of Installation

### 3.1.1 Prerequisite Reading

This chapter shows how to install PBS Professional. You should read the Release Notes and [Chapter 2, "Pre-Installation Steps", on page 7](#) before installing the software.

### 3.1.2 Replacing an Older Version of PBS

If you are installing on a system where PBS is already running, follow the instructions for an upgrade. Go to [Chapter 6, "Upgrading", on page 65](#).

### 3.1.3 Package Naming

Download the package for your platform from our website, and uncompress it. Packages are named like this:

```
PBSPro_<version>-<platform>_<hardware>.tar.gz.
```

For example, the PBS 19.2.2 package for CentOS 7 is named `PBSPro_19.2.2-CentOS7.tar.gz`. When you uncompress it, you'll find the following sub-package RPMs:

- Server/scheduler/MoM/communication/commands:  
`pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm`
- MoM/commands:  
`pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm`
- Commands:  
`pbspro-client-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm`

For example, for CentOS 7, the sub-packages are:

```
pbspro-server-19.2.2-<date etc.>-0.e17.x86_64.rpm  
pbspro-execution-19.2.2-<date etc.>-0.e17.x86_64.rpm  
pbspro-client-19.2.2-<date etc.>-0.e17.x86_64.rpm
```

## 3.2 Licenses

In order for a job to run, it must be running on a licensed host. Make sure that you have access to an Altair License Manager (ALM) license server that is hosting the licenses you need. Your license server can host either of these:

- Node licenses, which license a certain amount of hardware. Node licenses are obtained from Altair.
- Socket licenses, which are tied to hosts.

Each PBS complex can be licensed using PBSProNodes licenses or PBSProSockets licenses, but not both, so the ALM license server will provide one or the other. See the *PBS Works Licensing Guide*.

## 3.2.1 Licensing Caveats

If you do not tell PBS where to find the license server, the `pbs_license_info` attribute is left as is, which could be set to some previous value or unset. It is usually set to some previous value when doing an overlay or migration upgrade.

If the license server location is incorrectly initialized (e.g. the hostname or port number is incorrect), PBS may not be able to pinpoint the misconfiguration as the cause of the failure to reach a license server. The PBS server's first attempt to contact the license server results in the following message on the server's log file:

```
"unable to connect to license server at ..."
```

## 3.3 Major Steps for Installing PBS Professional

1. Set up your ALM license server with enough licenses for your site. See the *PBS Works Licensing Guide*.
2. Create accounts used by PBS. See [section 3.5.1.3, "Create PBS Data Service Management Account", on page 23](#) and [section 3.7.8, "Create Installation and Service Accounts", on page 39](#).
3. Download the correct PBS Professional package for each host. The PBS Professional package is available on the PBS download page at <https://secure.altair.com/UserArea/>.
4. Please read [section 3.4, "All Installations", on page 20](#). Then install PBS Professional on the server host and all execution hosts, without starting any daemons. For instructions, see [section 3.5, "Installing via RPM on Linux Systems", on page 23](#) or [section 3.7, "Installing PBS on Windows Hosts", on page 37](#).
5. Optionally, install additional communication daemons.
6. If you have additional communication daemons, start them using `systemd` or the PBS start/stop script. See [section 8.3, "Methods for Starting, Stopping, or Restarting PBS", on page 160](#).
7. Install PBS commands on any client hosts.
8. Start PBS on each execution host using `systemd` or the PBS start/stop script. See [section 8.3, "Methods for Starting, Stopping, or Restarting PBS", on page 160](#).
9. Start PBS on the server host using `systemd` or the PBS start/stop script. See [section 8.3, "Methods for Starting, Stopping, or Restarting PBS", on page 160](#).
10. Set the server's `pbs_license_info` attribute to point to the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```
11. Using the `qmgr` command, define the vnodes that the server will manage. See ["Creating Vnodes" on page 40 in the PBS Professional Administrator's Guide](#).
12. Perform post-installation tasks such as validation. See [Chapter 5, "Initial Configuration", on page 63](#).

## 3.4 All Installations

### 3.4.1 Automatic Installation of Database

Installing PBS automatically installs (and upgrades) the database used by PBS for its data store.

## 3.4.2 Choosing Installation Sub-package

On each PBS host, install the sub-package corresponding to the task(s) that host will perform. The task you give a host determines what we call the host. For example, a host that runs job tasks is called an "execution host". Sometimes there is more than one title that means the same thing; for example, some people call the server host the "headnode". Select the sub-package (or, for Windows, the installation option) that matches the desired task:

**Table 3-1: Choosing Installation Type**

Option	Host Role	Task	Package Contents	Parameters in pbs.conf For Default Start
1	Server host, headnode, front end machine	Runs server, scheduler, and communication daemons. Optionally runs MoM daemon. Client commands are included.  If using failover, install on both server hosts.	Server/scheduler/communication/MoM/client commands	PBS_START_SERVER=1 PBS_START_SCHED=1 PBS_START_COMM=1  To run MoM, add: PBS_START_MOM =1
2	Execution host, MoM host	Runs MoM. Executes job tasks. Client commands are included.  Install on each execution host.	Execution/client commands	PBS_START_MOM =1
3	Client host, submit host, submission host	Users can run PBS commands and view man pages.  Install on each client host.	Client commands	None

### 3.4.2.1 Pathname Conventions

The term *PBS\_HOME* refers to the location where the daemon/service configuration files, accounting logs, etc. are installed.

The term *PBS\_EXEC* refers to the location where the executable programs are installed.

## 3.4.3 Installing Additional Communication Daemons

By default, one communication daemon is installed on each server host. If you are configuring failover, your site will automatically have two communication daemons and all PBS daemons will automatically connect to them.

You may want to install additional communication daemons. For some rough guidelines on when you might want additional communication daemons, see [section 4.5.4, "Recommendations for Maximizing Communication Performance", on page 51](#).

To install just the communication daemon:

1. Download the appropriate PBS package
2. Uncompress the package
3. Make sure that parameters for `PBS_HOME`, `PBS_EXEC`, `PBS_LICENSE_INFO`, `PBS_SERVER` and `PBS_DATA_SERVICE_USER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
4. Install the server sub-package:

```
rpm -i
    <path/to/sub-package>pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

5. Edit `pbs.conf` to run only the communication daemon:

```
PBS_START_COMM=1
PBS_START_MOM=0
PBS_START_SCHED=0
PBS_START_SERVER=0
```

6. Start PBS:

```
systemctl start pbs
or
<path to script>/pbs start
```

7. Check to see that the communication daemon is running:

```
ps -ef | grep pbs
```

You should see that the `pbs_comm` daemon is running.

### 3.4.4 Deciding to Run a MoM After Installation

When you initially start PBS on a host that is configured not to run a MoM, PBS does not create MoM's home directory. If you later decide to run a MoM on this host:

1. Edit `pbs.conf` on that host and set `PBS_START_MOM=1`
2. You may find it helpful to source your `/etc/pbs.conf` file.
3. Run the `pbs_habitat` script:

```
$PBS_EXEC/libexec/pbs_habitat
```

4. Start PBS on the host:

```
systemctl start pbs
or
<path to start/stop script>/pbs start
```



## 3.4.5 Installation Method and Instructions by Platform

The procedure for installing PBS is the same on most platforms. Some platforms have a few minor differences, and some require special instructions. The following table lists instructions by platform:

**Table 3-2: Installation Method and Instructions by Platform**

Platform	Installation Method	Installation Instructions
RHEL	package manager, e.g. RPM	<a href="#">section 3.5, "Installing via RPM on Linux Systems", on page 23</a>
CentOS	package manager, e.g. RPM	<a href="#">section 3.5, "Installing via RPM on Linux Systems", on page 23</a>
HPE MC990X, Superdome Flex	package manager, e.g. RPM	<a href="#">section 3.5.3, "Installing on MC990X or Superdome Flex", on page 28</a>
HPE 8600	package manager, e.g. RPM	<a href="#">section 3.5.4, "Installing PBS on the HPE 8600", on page 30</a>
SuSE	package manager, e.g. RPM	<a href="#">section 3.5, "Installing via RPM on Linux Systems", on page 23</a>
CLE	RPM	<a href="#">"Installing and Upgrading on Cray" on page 139</a>
Ubuntu	deb	<a href="#">section 3.6, "Installing via dpkg on Ubuntu", on page 37</a>
Windows	PBS installation program provided by Altair	<a href="#">section 3.7, "Installing PBS on Windows Hosts", on page 37</a>

## 3.5 Installing via RPM on Linux Systems

### 3.5.1 Prerequisites for Installing on Linux Systems

#### 3.5.1.1 Prerequisite Reading

Please do not jump straight to this section in your reading. Before downloading and installing PBS, please make sure that you have read the following and taken any required steps:

- Prerequisites: All of [Section 2.1, "Prerequisites for Running PBS"](#), and [Section 2.1.7, "User Requirements on Linux"](#) and their subsections.
- Please read [Section 3.1, "Overview of Installation"](#).
- Make sure that you know how you will proceed by reading [Section 3.3, "Major Steps for Installing PBS Professional"](#).
- Please check all of [Section 3.4, "All Installations"](#) and its subsections to make sure you have prepared properly.

#### 3.5.1.2 Permissions

The location for the installation of the PBS Professional software binaries (PBS\_EXEC) and private directories (PBS\_HOME) must be owned and writable by root, and must not be writable by other users.

#### 3.5.1.3 Create PBS Data Service Management Account

Before you install PBS, you must create the PBS data service management account.

Note that there are two accounts related to the data service. Both have the same account name, but one is a Linux account and one is internal to the data service:

#### *PBS data service management account*

Created by administrator. Linux account with a Linux system password.

#### *Data service account*

Created by PBS on installation. Account that is internal to the data service, with its own data service password.

Used by PBS to log into and do operations on the data service. PBS maps this account to the PBS data service management account. Must have same name as PBS data service management account.

Create the PBS data service management account with the following characteristics:

- Non-root account
- Account must be for a system user; the UID must be less than 1000. Otherwise, the data service may be killed at inopportune times.
- Account is enabled
- If you are using failover, the UID of this account must be the same on both primary and secondary server hosts
- We recommend that the account is called *pbsdata*.
  - The installer looks for an account called *pbsdata*. If this account exists, the installer does not need to prompt for a username, and can install silently.
  - If you choose to use an account named something other than *pbsdata*, make sure you export an environment variable named `PBS_DATA_SERVICE_USER` with the value set to the desired existing PBS data service management account name.
- Root must be able to `su` to the PBS data service management account and run commands as that user. Do not add lines such as `'exec bash'` to the `.profile` of the PBS data service management account. If you want to use `bash` or similar, set this in the `/etc/passwd` file, via the OS tools for user management.
- The PBS data service management account must have a home directory.
- Do not put a CPU time limit on the data service Linux account. If you do, the datastore will die and kill the server.

### 3.5.1.4 Unset PBS\_EXEC Environment Variable

Unset the `PBS_EXEC` environment variable.

## 3.5.2 Generic Installation on Linux

For all platforms except those listed here, follow the generic instructions. The following platforms require their own steps:

- HPE MC990X and Superdome Flex: Go to [section 3.5.3, "Installing on MC990X or Superdome Flex", on page 28](#)
- HPE 8600: Go to [section 3.5.4, "Installing PBS on the HPE 8600", on page 30](#)

### 3.5.2.1 Downloading PBS

1. Download the PBS `tar.gz` package
2. Extract the tar file. For example:

```
tar zxvf PBSPro_<version>-linux26_i686.tar.gz
```

### 3.5.2.2 Setting Installation Parameters

Make sure that the `PBS_EXEC`, `PBS_HOME`, `PBS_LICENSE_INFO`, `PBS_SERVER` and `PBS_DATA_SERVICE_USER` parameters are specified at install time. PBS has default locations for `PBS_EXEC` and `PBS_HOME`, and a default value for `PBS_DATA_SERVICE_USER`, but you must specify the others.

You can override defaults at install time, in this order of precedence:

1. Via arguments to the package manager
2. Via environment variables
3. By specifying the desired parameters in `/etc/pbs.conf`. For details see ["The PBS Configuration File" on page 541 in the PBS Professional Administrator's Guide](#)

This table lists each parameter, its default value, and how it can be set at install time:

**Table 3-3: Setting Installation Parameters**

Parameter	Default Value	Specify via <code>pbs.conf</code>	Specify via Environment Variable	Specify via <code>rpm</code> Command
<code>PBS_DATA_SERVICE_USER</code>	<code>pbsdata</code>	No - ignored	Yes - environment variable only	No
<code>PBS_EXEC</code>	<code>/opt/pbs</code>	No - value in <code>pbs.conf</code> is overridden at install time. Note that changing this in <code>pbs.conf</code> breaks <code>rpm</code>	No - ignored	<code>--prefix &lt;location&gt;</code>
<code>PBS_HOME</code>	<code>/var/spool/pbs</code>	Yes	Yes	No
<code>PBS_LICENSE_INFO</code>	None	No - ignored	Yes - environment variable only. Can set <code>pbs_license_info</code> server attribute via <code>qmgr</code>	No
<code>PBS_SERVER</code>	For server installation: output of <code>hostname</code> command up to first period.  For all other installations: <code>"CHANGE_THIS_TO_PBS_PROD_SERVER_HOSTNAME"</code>	Yes	Yes	No

#### 3.5.2.2.i Caveats for Installation Parameters

Any `PBS_START_*` parameters set in the environment are not picked up and set in `pbs.conf`. You must specify these in `pbs.conf`; do not export them.

### 3.5.2.3 Installing on a Standalone Linux Machine

Make sure that you have covered the prerequisites in [section 3.5.1, "Prerequisites for Installing on Linux Systems", on page 23](#). The following example shows an installation on a single host on which all PBS components will run, and from which users will also submit jobs. The process may vary depending on the native package installer on your system.

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package
4. Make sure that parameters for PBS\_HOME, PBS\_EXEC, PBS\_LICENSE\_INFO, PBS\_SERVER and PBS\_DATA\_SERVICE\_USER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)

5. Install the server sub-package:

```
rpm -i
  <path/to/sub-package>pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

6. Edit pbs.conf to set PBS\_START\_MOM=1

7. Start PBS:

```
systemctl start pbs
```

or

```
<path to script>/pbs start
```

8. Check to see that the server, scheduler, MoM, and communication daemons are running:

```
ps -ef | grep pbs
```

You should see that the following daemons are running: pbs\_mom, pbs\_server, pbs\_sched, pbs\_comm

9. Make sure that user paths work, and submit sleep jobs. See [section 3.5.5, "Making User Paths Work", on page 36](#).

10. Verify that the jobs are running:

```
/opt/pbs/bin/qstat -a
```

11. Verify that you are running the correct version:

```
/opt/pbs/bin/qstat --version
```

12. Set the pbs\_license\_info server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

### 3.5.2.4 Installing on a Linux Cluster

Make sure that you have covered the prerequisites in [section 3.5.1, "Prerequisites for Installing on Linux Systems", on page 23](#).

You may or may not want to run batch jobs on the server/scheduler/communication host. First, install and start PBS on each execution host. Then install PBS on the server host. Follow these steps:

### 3.5.2.4.i Install PBS on Execution Hosts

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package
4. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
5. Install the PBS execution sub-package on each execution host:

```
rpm -i
  <path/to/sub-package>pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

6. Start PBS:

```
systemctl start pbs
```

or

```
<path to script>/pbs start
```

Instead of running the installer by hand on each machine, you can use a command such as `pdsh`. The one-line format for a non-default install is:

```
PBS_SERVER=<server name> PBS_HOME=<new home location> rpm -i --prefix <new exec location>
  pbspro-<sub-package>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 3.5.2.4.ii Install PBS on Server Host

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package
4. Make sure that parameters for PBS\_HOME, PBS\_EXEC, PBS\_LICENSE\_INFO, PBS\_SERVER and PBS\_DATA\_SERVICE\_USER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
5. If you want to run batch jobs on the front-end host, create or edit the `pbs.conf` file on the front-end machine so that a MoM runs there:

```
PBS_START_MOM=1
```

6. Install the server sub-package:

```
rpm -i
  <path/to/sub-package>pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 3.5.2.4.iii Start PBS on Server Host

Start PBS on the server machine by running `systemd` or the PBS start/stop script. If `/etc/init.d` exists, the script is in `/etc/init.d/pbs`, otherwise `/etc/rc.d/init.d/pbs`:

```
systemctl start pbs
```

or

```
<path to script>/pbs start
```

### 3.5.2.4.iv Configure Licensing

Set the `pbs_license_info` server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

### 3.5.2.4.v Install PBS on Client Hosts

Install PBS on each client host.

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package
4. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
5. Install the PBS client sub-package on each execution host:

```
rpm -i
    <path/to/sub-package>pbspro-client-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 3.5.2.4.vi Define Vnodes

Using the `qmgr` command, define the vnodes that the server will manage. See ["Creating Vnodes" on page 40 in the PBS Professional Administrator's Guide](#).

### 3.5.2.4.vii Check User Paths

Make sure that user paths work. See [section 3.5.5, "Making User Paths Work", on page 36](#).

## 3.5.3 Installing on MC990X or Superdome Flex

### 3.5.3.1 Prerequisites for Installing on a MC990X or Superdome Flex

Make sure that you have covered the prerequisites in [section 3.5.1, "Prerequisites for Installing on Linux Systems", on page 23](#). On these machines, you install the PBS server package and use `cgroups` to manage `cpusets`.

### 3.5.3.2 Download and Install the New PBS

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package
4. Make sure that parameters for PBS\_HOME, PBS\_EXEC, PBS\_LICENSE\_INFO, PBS\_SERVER and PBS\_DATA\_SERVICE\_USER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
5. Install the server sub-package:

```
rpm -i
    <path/to/sub-package>pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 3.5.3.3 Start PBS

1. Edit `pbs.conf` to set `PBS_START_MOM=1`
2. Start the PBS daemons by running `systemd` or the PBS start/stop script. The location of the script varies depending on system configuration.

```
systemctl start pbs
```

or

```
<path to script>/pbs start
```

### 3.5.3.4 Configure Licensing

Set the `pbs_license_info` server attribute to the location of the license server(s):

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

### 3.5.3.5 Test the New PBS

1. Check to see that the PBS daemons are running. You should see that there are four daemons running: `pbs_mom`, `pbs_server`, `pbs_sched`, `pbs_comm`:

```
ps -ef | grep pbs
```

2. Submit jobs as a normal user.

Submit a job to the default queue:

```
echo "sleep 60" | /opt/pbs/bin/qsub
```

3. Verify that the jobs are running:

```
/opt/pbs/bin/qstat -an
```

### 3.5.3.6 Configure Cgroups to Manage Cpusets

1. Make sure that your cgroups hook is enabled and that you can use cgroups. See ["Configuring and Using PBS with Cgroups" on page 573 in the PBS Professional Administrator's Guide](#).

1. Export the cgroups hook configuration file to `pbs_cgroups.json`:

```
# qmgr -c 'export hook pbs_cgroups application/x-config default' > pbs_cgroups.json
```

2. You can make the cgroups hook mimic the behavior of the cpuset MoM in previous versions:

- a. Create one vnode for each NUMA node. Edit `pbs_cgroups.json` as follows (important):

```
"vnode_per_numa_node" : true,
```

- b. Edit `pbs_cgroups.json` as follows (recommended):

```
"use_hyperthreads" : true,
```

3. If the cgroups memory subsystem is not mounted on the system, disable 'memory' in the cgroups hook configuration file:

- a. Check to see whether it is mounted:

```
# mount | grep cgroup | grep memory
```

If the memory subsystem is mounted, the command returns something like "cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)".

- b. If this returns empty, edit the `pbs_cgroups.json` file so that 'enabled' parameter for 'memory' under cgroup is *false*:

```
"cgroup": {
```

```
...
```

```
"memory": {
```

```
"enabled": false,
```

4. Import the modified configuration (make sure you use "x-config"):

```
# qmgr -c 'import hook pbs_cgroups application/x-config default pbs_cgroups.json'
```

### 3.5.3.7 Restart MoMs

On each execution host, restart MoM :

```
ps -eaf | grep pbs_mom
kill <MoM PID>
/opt/pbs/sbin/pbs_mom
```

## 3.5.4 Installing PBS on the HPE 8600

### 3.5.4.1 HPE 8600 Components

An 8600 system consists of one Admin node, one or more Service (login) nodes, and a set of one or more compute racks. Each compute rack consists of one or more IRU nodes and one or more compute nodes per IRU. The racks are diskless. The root file system of the IRU and compute nodes are mounted read-only from a NAS managed by the Admin node. There is a single image of the root file system for all of the compute nodes and a separate image for all of the IRU nodes. HPE Performance Cluster Manager node management commands are used to publish the image to the various nodes in a process that involves powering down the nodes, pushing a new image, and re-powering the nodes.

In a typical configuration, user home file systems are mounted from NAS, and each node has a separately mounted file system for `/var/spool`.

HPE follows a naming convention when preparing a system for shipment. Service nodes are named "service0", "service1", ... Compute nodes are named "rRiLnN" where 'R' is the rack number starting with 1; 'L' is the IRU node number within a rack starting with 0 in each rack; N is the node number, starting with 0, under the specific Rack Leader. For example, two racks with 2 IRUs per rack and 4 nodes per IRU are named:

**Table 3-4: Node Names**

IRU	Rack 1	Rack 2
IRU 0	<i>r1i0n0</i>	<i>r2i0n0</i>
	<i>r1i0n1</i>	<i>r2i0n1</i>
	<i>r1i0n2</i>	<i>r2i0n2</i>
	<i>r1i0n3</i>	<i>r2i0n3</i>
IRU 1	<i>r1i1n0</i>	<i>r2i1n0</i>
	<i>r1i1n1</i>	<i>r2i1n1</i>
	<i>r1i1n2</i>	<i>r2i1n2</i>
	<i>r1i1n3</i>	<i>r2i1n3</i>



### 3.5.4.2 Requirements for the HPE 8600 with HPE MPI

- Make sure that you have covered the prerequisites in [section 3.5.1, "Prerequisites for Installing on Linux Systems", on page 23](#).
- In order to run PBS on the HPE 8600 with HPE MPI, HPE Performance Cluster Manager node management tools must already be installed. You will be using the following HPE Performance Cluster Manager commands:

**Table 3-5: Performance Cluster Manager Commands**

Performance Cluster Manager Command	Description
<code>cnodes --ice-compute</code>	List the compute node names; useful in scripting operations
<code>cpower node off &lt;node name&gt;</code>	Powers down
<code>cpower node on &lt;node name&gt;</code>	Powers up the named nodes
<code>cimage --...</code>	Manages the file system image for the various nodes

- You must use the correct names for the Admin and Service nodes in any commands.

### 3.5.4.3 Choosing Whether PBS Will Manage Cpusets with HPE 8600 Running HPE MPI

You can use cpusets on an HPE 8600 running PBS, whether or not PBS manages the cpusets. If PBS manages the cpusets for you, that means that PBS dynamically creates a cpuset for each job and confines job processes to that cpuset. If PBS does not manage the cpusets for you, then jobs are not confined to cpusets. You can use the PBS cgroups hook to manage the cpusets on the 8600; see [section 3.5.4.10, "Configure Cgroups to Manage Cpusets", on page 35](#).

### 3.5.4.4 Installation of the PBS Server, Scheduler, and Communication Daemons

Install the PBS server, scheduler, communication daemon, and commands on a single service node; here we assume this node is "service0":

1. Log on to service0 as root.
2. Unzip and untar the appropriate package.
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, PBS\_LICENSE\_INFO, PBS\_SERVER and PBS\_DATA\_SERVICE\_USER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
4. Install the server sub-package:
 

```
rpm -i
<path/to/sub-package>pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
5. Do not start PBS

### 3.5.4.5 Installation of the PBS MoM

You install and configure MoM once on the root file system, then you push the image to all of the compute nodes by propagating it to the rack leaders. Then you reboot each node with the new image.

1. Log on to the Admin node as root.
2. Determine which image file is being used on the compute nodes. To list the nodes on rack 1:

```
cimage --list-nodes r1
```

It will show output in the form "*node: image\_name kernel*" similar to

```
r1i0n0: compute-sles15sp1 2.6.26.46-0.12-smp
```

Thus node r1i0n0 is running the image "compute-sles15sp1" and the kernel version "2.6.26.46-0.12-smp". For the remaining steps, it is assumed that those are the images and kernel available.

3. List the available images:

```
cimage --list-images
```

which will list the images available for the compute nodes. Each image may have multiple kernels.

4. Unless you are experienced in managing the image files, we suggest that you create a copy of the image in use and install PBS in that copy. To copy an image:

```
cinstallman --create-image --clone --source compute-sles15sp1 --image compute-sles15sp1pbs
```

5. The image file lives in the directory `/opt/clmgr/image/images`, so change into the `tmp` directory found in the new image just cloned:

```
cd /opt/clmgr/image/images/compute-sles15sp1pbs/tmp
```

6. Chroot to the new image file:

```
chroot /opt/clmgr/image/images/compute-sles15sp1pbs /bin/sh
```

The new root is in effect.

7. Download, unzip and untar the PBS package
8. Make sure that parameters for `PBS_HOME`, `PBS_EXEC` and `PBS_SERVER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
9. Install the PBS execution sub-package in the normal execution directory, `/opt/pbs`, in this system image:

```
rpm -i <path/to/sub-package>pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

10. Do not start PBS
11. Exit from the chroot shell and return to root's normal home directory.
12. Power down each rack of compute nodes:

```
for n in `cnodes --ice-compute` ; do
    cpower node off $n
done
```

13. Publish the new system image to the compute nodes:

```
cimage --push-rack compute-sles15sp1pbs r\*
```

This instruction will take several minutes to finish.

14. Set the new image and kernel to be booted. This need not be done if: (1) rather than cloning a new image, you have installed PBS into the image already running on the compute nodes; or (2) you are using an image that was already pushed to the nodes.

```
cimage --set compute-sles15sp1pbs 2.6.26.46-0.12-smp r\*i\*n\*
```

15. Power up the compute nodes:

```
for n in `cnodes --ice-compute` ; do
    cpower node on $n
done
```

It will take several minutes for the compute nodes to reboot.

### 3.5.4.6 Start PBS Server

1. Log on to the Service node as root
2. On the Service node, start the PBS server, scheduler, and communication daemons:

```
systemctl start pbs
or
<path to script>/pbs start
```

### 3.5.4.7 Configure Licensing

Set the pbs\_license\_info server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

### 3.5.4.8 Add Compute Nodes

Using qmgr, add the compute nodes to the PBS configuration:

```
for N in `cnodes --ice-compute`
do
    qmgr -c "create node $N"
done
```

If you use the IP address for the name of the vnode:

1. Add PBS\_MOM\_NODE\_NAME=<IP address> to pbs.conf on the execution host
2. Restart MoM

### 3.5.4.9 Configuring Placement Sets on the HPE 8600

Placement sets improve job placement on execution nodes. If you want to use cgroups, you can generate placement set information. See ["Placement Sets" on page 170 in the PBS Professional Administrator's Guide](#).

Placement sets can be defined only after you have defined the compute nodes as in the previous section. Put placement set resource information in a Version 2 configuration file for each host. Make sure that the vnode names you use in your Version 2 configuration file are exactly the same as the names generated by the cgroups hook.

Steps to generate placement sets:

1. Shut down the server.
2. Add a resource named "router" (the script uses this exact name):  

```
Qmgr: create resource router type=string_array, flag=h
```
3. Restart the server
4. Generate your placement sets and set their resource values at vnodes; you can use the `sgiICEplacement.sh` script, which is in the `unsupported` directory, as an example
5. Verify the result:
  - a. Run the `pbsnodes -a` command
  - b. Look for the line `resources_available.router` at each vnode. The value assigned to the "router" resource should be in the form `"r#,r##"`, where *r* identifies the rack number and *i* identifies the IRU number.

### 3.5.4.10 Configure Cgroups to Manage Cpusets

Do the following steps as root on the server node (service0).

1. Make sure that cgroups subsystems including `cpuset` are mounted on the compute nodes. See ["Configuring and Using PBS with Cgroups" on page 573 in the PBS Professional Administrator's Guide](#).
2. Modify the cgroups hook configuration file:
  - a. Export the cgroups hook configuration file:
 

```
qmgr -c "export hook pbs_cgroups application/x-config default" > pbs_cgroups.json
```
  - b. Edit the cgroups configuration file. To get default cpuset behavior, set these:
 

```
"vnode_per_numa_node" : true,
"use_hyperthreads"     : true,
"ncpus_are_cores"     : false,
```

 We describe how to manage hyperthreading behavior in ["Configuring Hyperthreading Support" on page 584 in the PBS Professional Administrator's Guide](#).
  - c. If the cgroups `memory` subsystem is not mounted on the system, disable 'memory' in the cgroups hook configuration file. Check to see whether it is mounted:
 

```
# mount | grep cgroup | grep memory
```

 If the `memory` subsystem is mounted, the command returns something like `"cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)"`.
  - d. If this returns empty, edit the `pbs_cgroups.json` file so that 'enabled' parameter for 'memory' under `cgroup` is *false*:
 

```
"cgroup": {
  ...
  "memory": {
    "enabled": false,
```
  - e. Read in the updated cgroups hook configuration:
 

```
qmgr -c "import hook pbs_cgroups application/x-config default pbs_cgroups.json"
```
3. Enable the cgroups hook:
 

```
# qmgr -c "set hook pbs_cgroups enabled=true"
```
4. Restart the MoMs, using either `systemctl` or the start/stop script:
 

```
# for n in `cnodes --ice-compute`; do
  ssh $n "systemctl restart pbs"
done
or
# for n in `cnodes --ice-compute`; do
  ssh $n "<path to script>/pbs restart"
done
```
5. Check that you have created one vnode for each NUMA node, and that the vnode state is *free*:
 

```
# pbsnodes -av
```

---

## 3.5.5 Making User Paths Work

If you're installing PBS for the first time, make sure that user PATHs include the location of the PBS commands. If users already have paths to PBS commands, you can either make symbolic links so that users don't have to change their PATHs, or users can set their PATHs to the locations of the commands.

### 3.5.5.1 Setting User Paths to Location of Commands

Users should set their path to include `PBS_EXEC/bin` and `PBS_EXEC/sbin`. For example, if `PBS_EXEC` is `/opt/pbs`, by including `/opt/pbs/bin`, users will have PBS executables in their path.

### 3.5.5.2 Making Existing User Paths Work with New Location

You may need to make users' `PATH` variable point to the new `PBS_EXEC` directory, especially if `PBS_EXEC` is in a non-default location, or if you're using a new location. You can use symbolic links to enable users to access PBS commands via their current `PATH`:

```
<user PATH>/bin -> <PBS_EXEC>/bin
<user PATH>/sbin -> <PBS_EXEC>/sbin
```

For example if the old location was `/usr/pbs_bin`, create the link `/usr/pbs_bin/bin -> /opt/pbs/bin`.

### 3.5.5.3 Testing User Paths

- Test that a normal user can submit a job. As a normal user, type:  

```
echo "sleep 60" | /opt/pbs/bin/qsub
```

This submits a job to the queue named 'workq' (the queue that is automatically defined as the default queue)
- If you've changed the location of PBS commands and used symbolic links to allow users to keep their old PATHs, verify that the old paths work:  

```
echo "sleep 60" | <old user path>/bin/qsub
```

## 3.5.6 Caveats for Uninstalling on Linux

Using `rpm -e`, even on an older package than the one you are currently using, will cause any currently running PBS daemons to shut down, and will also remove the system `V` init and/or `systemd` service startup files. This will prevent PBS daemons from starting automatically at system boot time. If you wish to remove an older RPM without these effects, use `rpm -e --noscripts`.

## 3.6 Installing via dpkg on Ubuntu

To install PBS Professional on Ubuntu, use the following steps:

1. Choose the `.deb` package to install. Make sure it is appropriate for the host's function, which could be server, execution, or client host.

2. Use `dpkg -i` to install the `.deb` package:

```
dpkg -i <.deb package>
```

3. Update `/etc/pbs.conf`: set the `PBS_START_*` parameters to the appropriate values. Here is an example where one host will run all daemons:

```
PBS_EXEC=/opt/pbs
PBS_SERVER=<hostname>
PBS_START_SERVER=1
PBS_START_SCHED=1
PBS_START_COMM=1
PBS_START_MOM=1
PBS_HOME=/var/spool/pbs
PBS_CORE_LIMIT=unlimited
PBS_SCP=/usr/bin/scp
```

4. Each hostname **must** resolve to at least one non-loopback IP address. Typically, the default `/etc/hosts` file does not conform to this prerequisite, so you probably need to do additional network configuration to make PBS work on Ubuntu. You can do this by using DNS or by adding a new entry into `/etc/hosts` that associates the hostname with a non-loopback IP address. To update `/etc/hosts`:

Update the IP address for the server host:

```
127.0.0.1      localhost
192.168.238.135 <server hostname>
```

The following lines are desirable for IPv6-capable hosts:

```
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

5. If the PBS [Data service management account](#), usually called `pbsdata`, does not already exist, create it. See [section 3.5.1.3, "Create PBS Data Service Management Account"](#), on page 23.

6. Start PBS:

```
/etc/init/pbs start
```

## 3.7 Installing PBS on Windows Hosts

### 3.7.1 Daemon Layout

MoMs and client commands can run on Windows machines, but all other PBS components are installed on Linux hosts. Install the Windows MoM and client packages, and install your chosen Linux server/scheduler/comm package.

---

## 3.7.2 Prerequisites

Please do not jump straight to this section in your reading. Before downloading and installing PBS, please make sure that you have read the following and taken any required steps:

- Prerequisites: All of [Section 2.1, "Prerequisites for Running PBS"](#), [Section 2.3, "PBS Configurations for Windows"](#), [Section 2.3.6, "User Authorization Under Windows"](#), and [Section 2.3.8, "Windows Caveats"](#) and their subsections.
- Please read [Section 3.1, "Overview of Installation"](#).
- Please start your installation by following the steps in [Section 3.3, "Major Steps for Installing PBS Professional"](#).
- Please check all of [Section 3.4, "All Installations"](#) and its subsections to make sure you have prepared properly.

## 3.7.3 Default Installation Locations

On Windows systems, PBS is installed in `\Program Files (x86)\PBS\`.

Default installation directories:

PBS\_HOME: `C:\Program Files (x86)\PBS\home`

PBS\_EXEC: `C:\Program Files (x86)\PBS\exec`

## 3.7.4 Where to Run Daemons (Services)

When PBS is installed on a complex, the MoM must be run on each execution host. The server, scheduler, and communication daemons are installed on a Linux front-end system. The PBS Windows package contains the following:

- PBS Professional software
- Supporting text files (README etc.)

## 3.7.5 PBS Requirements on Windows

All Windows hosts in a PBS complex must be in the same domain.

PBS Professional is supported if the domain controller server is configured "native". Running PBS in an environment where the domain controllers are configured in "mixed-mode" is not supported.

You must install PBS Professional from an Administrator account.

Before you install PBS on Windows, make sure you are using the correct type of account. See [section 2.3.5, "Windows Configuration in a Domained Environment"](#), on page 14.

PBS Professional requires that the drive that PBS was installed under (e.g. `\Program Files\PBS` or `\Program Files (x86)\PBS`) be configured as an NTFS filesystem.

Before installing PBS Professional, be sure to uninstall any old PBS Professional files. For details see ["Uninstalling PBS Professional on Windows"](#) on page 44.

You can specify the destination folder for PBS using the "Ask Destination Path" dialog during setup.

## 3.7.6 Make Sure Hostnames Resolve Correctly

Make sure that all of your hosts consistently resolve to the correct IP addresses. Wrong IP address to hostname translation will cause errors for PBS.

Configure your system to talk to a properly configured and functioning DNS server.



On each host, add the correct host entries to the following files:

```
c:\windows\system32\drivers\etc\hosts
hosts.equiv
```

Make each `etc\hosts` file identical on each host, and make each `hosts.equiv` file identical on each host.

Example 3-1: Your server is `serverA`, your execution host is `exec01`, and your client hosts are `client001` and `client002`. Hostnames and IP addresses look like this:

**Table 3-6: Example Host Names and Addresses**

Hostname	Host IP Address
serverA	192.168.0.101
exec01	192.168.0.102
client001	192.168.0.103
client002	192.168.0.104

Here's what `etc\hosts` should look like at each host:

```
192.168.0.101 server
192.168.0.102 mom
192.168.0.103 client001
192.168.0.104 client002
```

Here's what `hosts.equiv` should look like at each host:

```
server
mom
client001
client002
```

### 3.7.7 Create Job Submission Accounts

Set up any user accounts that will be used to run PBS jobs. All job submission accounts must be part of the same domain as any Windows hosts. The accounts should not be Administrator-type accounts, that is, not a member of the "Domain Administrators" or local "Administrators" group, so that basic authentication using `hosts.equiv` can be used.

Once the accounts have been set up, list all PBS hosts (server, execution, client, file storage) in the `hosts.equiv` or job submitters' `.rhosts` files. Do this on all the hosts, to allow accounts on these hosts to access PBS services such as job submission and remote file copying.

The `hosts.equiv` file can usually be found in the following location:

```
C:\windows\system32\drivers\etc\hosts.equiv
```

### 3.7.8 Create Installation and Service Accounts

Before you install PBS, you must create the accounts that PBS requires.

On Windows, the PBS data service management account is the same as the PBS Windows service account. You do not need to create a separate data service account.

You need to create the installation and service accounts. We give instructions below.

You do not need to create the following accounts:

#### *PBS data service management account*

On Windows, the PBS data service management account is the same as the PBS Windows service account. You do not need to create a separate PBS data service management account.

#### *Data service account*

Account that is internal to the data service, and has its own data service password. On installation, PBS creates the internal data service account, and maps it to the PBS service account. The data service account name must be the same as the PBS service account.

You do need to create the installation and service accounts, and we give instructions below.

### 3.7.8.1 Creating Installation Account in Domained Environment

The installation account is the account from which PBS is installed. The installation account must be the only account that will be used for all steps of PBS installation including modifying configuration files, setting up failover, and so on. If any of the PBS configuration files are modified by an account that is not the installation account, permissions/ownerships of the files could be reset, rendering them inaccessible to PBS. For domained environments, the installation account must be a local account that is a member of the local Administrators group on the local computer.

### 3.7.8.2 Creating PBS Service Account in Domained Environment

The PBS service account is the account under which the PBS service (`pbs_mom`) will run.

- This account can have any name.
- The name of the account defaults to *pbsadmin*.
- This account must exist while any PBS services are running.
- The password for this account should not be changed while PBS is running.
- Create the PBS service account before installing PBS.
- For domained environments, the PBS service account must:
  - a. be a domain account
  - b. be a member of the "Domain Users" group, and **only** this group
  - c. have "domain read" privilege to all users and groups.
- For a domained environment, delegate "read access to all users and groups information" to the PBS service account. See [section 3.7.8.2.i, "Delegating Read Access to PBS Service Account in Domained Environment", on page 41](#).
- If the PBS service account is set up with no explicit domain read privilege, MoM may hang. The hang happens when users submit jobs from a network mapped drive without the `-o/-e` option for redirecting files. When this happens, bring up Task manager, look for a "cmd" process by the user who owned the job, and kill it. After the first cmd process is killed, you may have to look for a second one (the first one copies the output file, the second one does the error file). This should un-hang the MoM.
- The PBS service account must be a member of the local Administrators group. Add the PBS service account to the local Administrators group:
 

```
net localgroup Administrators <domain name>\<service account name> /add
```
- Do not put a CPU time limit on the service account . If you do, the datastore will die and kill the server.

---

### 3.7.8.2.i Delegating Read Access to PBS Service Account in Domained Environment

- To delegate "read access to users and groups information" to the PBS service account:
  - a. On the domain controller host, bring up Active Directory Users and Computers.
  - b. Select <domain name>, right mouse click, and choose "Delegate Control". This will bring up the "Delegation of Control Wizard".
  - c. When it asks for a user or group to which to delegate control, select the name of the PBS service account.
  - d. When it asks for a task to delegate, specify "Create a custom task to delegate".
  - e. For active directory object type, select the "this folder, existing objects in this folder, and creation of objects in this folder" button.
  - f. For permissions, select "Read" and "Read All Properties".
  - g. Exit out of Active Directory.

### 3.7.8.2.ii Service Account Caveats

If you change the name of the PBS service account:

- You must restart the daemons on that host
- On Windows, you must re-register the MoM service

## 3.7.9 Installation Notes for Domained Environment

### 3.7.9.1 Installation Path

- The destination/installation path of PBS must be NTFS. All PBS configuration files must reside on an NTFS filesystem.

### 3.7.9.2 Notes on Installation

- The installation account must be used in all future invocations of the install program when setting up a complex of PBS hosts.
- The install program requires the installer to supply the password for the PBS service account. This same password must be supplied to future invocations of the install program on other hosts.
- The install program will enable the following rights to the PBS service account: "Create Token Object", "Replace Process Level Token", "Log On As a Service", and "Act As Part of the Operating System".
- The install program will enable Full Control permission to local "Administrators" group on the install host for all PBS-related files.
- The install program will give you a specific error if the PBS service account is not a member of the local Administrators group on the local computer. It will quit at this point, and you must go back:
  - a. Make the PBS service account be a member of the local Administrators group on the local computer:

```
net localgroup Administrators <name of PBS service account> /add
```
  - b. Re-run the install program.

## 3.7.10 Steps to Install PBS on Windows

1. On each execution and client host, do the following:
  - a. Log in with the installation account.
  - b. Install the KB2999226 update for Windows on all Windows Server 2012 execution and client machines.
  - c. Download the MSI installer (the .msi file).
  - d. Double-click the MSI installer; the splash screen is displayed.
  - e. Click the *Next* button to move to the license page. Accept the license.
  - f. Click the *Next* button and choose the path where you will install the PBS executable. By default this path points to "C:\Program Files (x86)\PBS\".
  - g. Using "Run As Administrator", open a Command prompt.
2. Install the server/scheduler package on a Linux host. See [section 3.5.2.4.ii, "Install PBS on Server Host", on page 27](#).

## 3.7.11 Post-installation Steps

### 3.7.11.1 Configuring MoMs

On each execution host, manually execute the `win_postinstall.py` script as shown below. 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 `<domain>\<username>`, pass only *username* as an argument:

```
<PBS_EXEC>\python\python.exe <PBS_EXEC>\etc\win_postinstall.py -u <PBS service account> -p <PBS
service account password> -s <server name> -t execution -c <path to scp.exe>
```

### 3.7.11.2 Configuring Client Hosts

On each client host, manually execute the `win_postinstall.py` script as shown below. 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 `<domain>\<username>`, pass only *username* as an argument:

```
<PBS_EXEC>\python\python.exe <PBS_EXEC>\etc\win_postinstall.py -u <PBS service account> -p <PBS
service account password> -s <server name> -t client -c <path to scp.exe>
```

### 3.7.11.3 Defining Vnodes

Using the `qmgr` command, define the vnodes that the server will manage. See ["Creating Vnodes" on page 40 in the PBS Professional Administrator's Guide](#).

### 3.7.11.4 Configuring Remote File Copy

If you will use `scp` for your remote file copy mechanism, configure passwordless `ssh`. If you will use `$usecp` to specify your remote file copy mechanism, you do not need to configure passwordless `ssh`. See ["Configuring MoM for Remote Copy" on page 564 in the PBS Professional Administrator's Guide](#).

## 3.7.12 Post-installation Considerations on Windows

### 3.7.12.1 File Creation

The installation process automatically creates the following file:

```
[PBS Destination folder]\pbs.conf
```

containing at least the following entries:

```
PBS_EXEC=[PBS Destination Folder]\exec
PBS_HOME=[PBS Destination Folder]\home
PBS_SERVER=<server name>
PBS_START_SERVER=<value>
PBS_START_SCHED=<value>
PBS_START_MOM=<value>
PBS_START_COMM=<value>
PBS_AUTH_METHOD=pwd
```

where `PBS_EXEC` will contain subdirectories where the executable and scripts reside, `PBS_HOME` will house the log files, job files, and other processing files, and `server-name` will reference the system running the PBS server. The `pbs.conf` file can be edited by calling the PBS program "`pbs-config-add`". For example:

```
\Program Files (x86)\PBS\exec\bin\pbs-config-add "PBS_SCP=C:\Windows\System32\OpenSSH\scp.exe"
```

Don't edit `pbs.conf` directly as the permission on the file could get reset causing other users to have a problem running PBS.

### 3.7.12.2 File Access on Windows

Upon installation, some PBS directories have restricted access. The following directories have files that are readable by the `\Everyone` group but writable only by Administrators-type accounts:

```
PBS_HOME/mom_logs/
PBS_HOME/spool/
```

The following directories have files that are only accessible to Administrators-type accounts:

```
PBS_HOME/mom_priv/
```

## 3.7.13 Startup on Windows

- The auto-startup of the MoM service is controlled by the PBS `pbs.conf` file as well as the *Services* dialog. You invoke this via *Settings->Control Panel->Administrative Tools->Services*. If the service fails to start up with the message, "incorrect environment", it means that the `PBS_START_MOM` `pbs.conf` variable is set to `0` (*False*).
- On Windows, sometimes PBS may fail to start automatically after the boot. We recommend that you change the startup mode from "*[Startup type: Automatic]*" to "*[Startup type: Automatic (Delayed Start)]*", which means "shortly after boot".

Open `regedit` to change the registry keys. These are, in some versions of Windows: `HKLM\SYSTEM\CurrentControlSet\Services\<PBS Professional>\DelayedAutostart`.

When startup is delayed, this has the value `1`. When not delayed, its value is `0`.

---

### 3.7.13.1 Setting Up User Accounts and Directories

You should review the recommended steps for setting up user accounts and home directories, as documented in [section 2.3.6, "User Authorization Under Windows"](#), on page 15.

### 3.7.14 Uninstalling PBS Professional on Windows

For uninstalling versions 5.4.2 through 8.0, use a domain admin account. For later versions, use an Administrator account. Note that as of 19.4.1, the only PBS service on Windows is PBS\_MOM.

1. Use the Task Manager to stop/kill the services: PBS\_SERVER, PBS\_SCHED, PBS\_COMM, PBS\_MOM, and PBS\_RSHD.

2. Manually de-register the PBS services:

```
pbs_account --unreg pbs_server
pbs_account --unreg pbs_sched
pbs_account --unreg pbs_comm
pbs_account --unreg pbs_mom
pbs_account --unreg pbs_rshd
```

3. Use the MSI installer to uninstall the PBS package. At the second double click, you get the "*Remove*" option.
4. Manually delete the PBS directory at "C:\program Files (x86)\PBS"

# Communication

## 4.1 Communication Within a PBS Complex

There are two primary communication methods in PBS: TCP, where a client sends a request to a server using a non-permanent TCP connection, and TPP, in which daemons establish permanent TCP connections to one or more `pbs_comm` daemons and use these permanent connections to reach other daemons. TPP stands for "TCP-based Packet Protocol".

A PBS complex using TPP can handle much greater throughput than in previous versions of PBS, and the scheduler can start jobs much faster. A PBS complex using TPP does not need as many reserved ports as previous versions.

## 4.2 Terminology

### Endpoint

A PBS server, scheduler, or MoM daemon.

### Communication daemon, `comm`

The daemon which handles communication between the server, scheduler, and MoMs. Executable is `pbs_comm`.

### Leaf

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

### TPP

TCP-based Packet Protocol. Protocol used by `pbs_comm`.

## 4.3 Prerequisites

Each hostname must resolve to at least one non-loopback IP address.

## 4.4 Communication Parameters

### 4.4.1 Location of Communication Daemon for Endpoint

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

#### `PBS_LEAF_ROUTERS`

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

Format: `PBS_LEAF_ROUTERS=<host>[:<port>][,<host>[:>port>]]`

## 4.4.2 Location of Other Communication Daemons

When you add a communication daemon, you must tell it about the other `pbs_comms` 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.

### PBS\_COMM\_ROUTERS

Parameter in `/etc/pbs.conf`. Tells a `pbs_comm` where to find its fellow communication daemons.

Format: `PBS_COMM_ROUTERS=<host>[:<port>][,<host>[:<port>]]`

## 4.4.3 Number of Threads for Communication Daemon

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.

### PBS\_COMM\_THREADS

Parameter in `/etc/pbs.conf`. Tells `pbs_comm` how many threads to start.

Maximum allowed value: `100`

Format: *Integer*

Example:

```
PBS_COMM_THREADS=8
```

## 4.4.4 Daemon Log Mask

By default, `pbs_comm` produces few log messages. You can choose more logging, usually for troubleshooting. See [section 4.5.10, "Logging and Errors with TPP", on page 54](#) for logging details. The daemon rereads this parameter when HUPed.

### PBS\_COMM\_LOG\_EVENTS

Parameter in `/etc/pbs.conf`. Tells `pbs_comm` which log mask to use.

Format: *Integer*

Default: `511`

Example:

```
PBS_COMM_LOG_EVENTS=<log level>
```

## 4.4.5 Name of Endpoint Host

By default, the name of the endpoint's host is the hostname of the machine. You can set the name that the endpoint uses for its host. 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.



## 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.

Canonicalized value of this becomes the value of `resources_available.host`.

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.

The server only queries for the canonicalized address of the MoM host, unless you let it know via the `Mom` attribute; if you have set `PBS_LEAF_NAME` in `/etc/pbs.conf` to something else, make sure you set the `Mom` attribute at vnode creation.

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
```

## 4.4.6 Whether Host Runs Communication Daemon

Just as with the other PBS daemons, you can specify whether each host should start `pbs_comm`.

### 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.

Format: *Boolean*

Default: `0`

Example:

```
PBS_START_COMM=1
```

## 4.4.7 Scheduler Throughput Mode

You can tell the scheduler to run asynchronously, so it doesn't wait for each job to be accepted by MoM, which means it also doesn't wait for an `execjob_begin` hook to finish. Especially for short jobs, this can give you better scheduling performance. You can run the scheduler asynchronously only when the complex is using TPP mode.

### throughput\_mode

Scheduler attribute. When set to `True`, the scheduler runs asynchronously and can start jobs faster. Only available when complex is in TPP mode.

Format: *Boolean*

Default: `True`

Example:

```
qmgr -c "set sched throughput_mode=<Boolean value>"
```

Trying to set the value to a non-Boolean value generates the following error message:

```
qmgr obj= svr=default: Illegal attribute or resource value
qmgr: Error (15014) returned from server
```

## 4.4.8 Managing Communication Behavior

### rpp\_highwater

Server attribute.

This is the maximum number of messages per stream (meaning the maximum number of messages between each pair of endpoints).

Format: *Integer*

Valid values: *Greater than or equal to one*

Default: *1024*

Python type: *int*

### rpp\_max\_pkt\_check

Server attribute.

Maximum number of TPP messages processed by the main server thread per iteration.

Format: *Integer*

Default: *64*

Python type: *int*

### rpp\_retry

Server attribute.

In a fault-tolerant setup (multiple `pbs_comms`), when the first `pbs_comm` fails partway through a message, this is number of times TPP tries to use any other remaining `pbs_comms` to send the message.

Format: *Integer*

Valid values: *Greater than or equal to zero*

Default: *10*

Python type: *int*

## 4.5 Inter-daemon Communication Using TPP

The PBS server, scheduler, and MoM daemons communicate with each other using TPP through the communication daemon `pbs_comm`, except for scheduler-server and server-server communication, which uses TCP. The server, scheduler, and MoMs are communication endpoints, connected by one or more `pbs_comm` daemons. The following figure illustrates communication within a PBS complex using TPP.

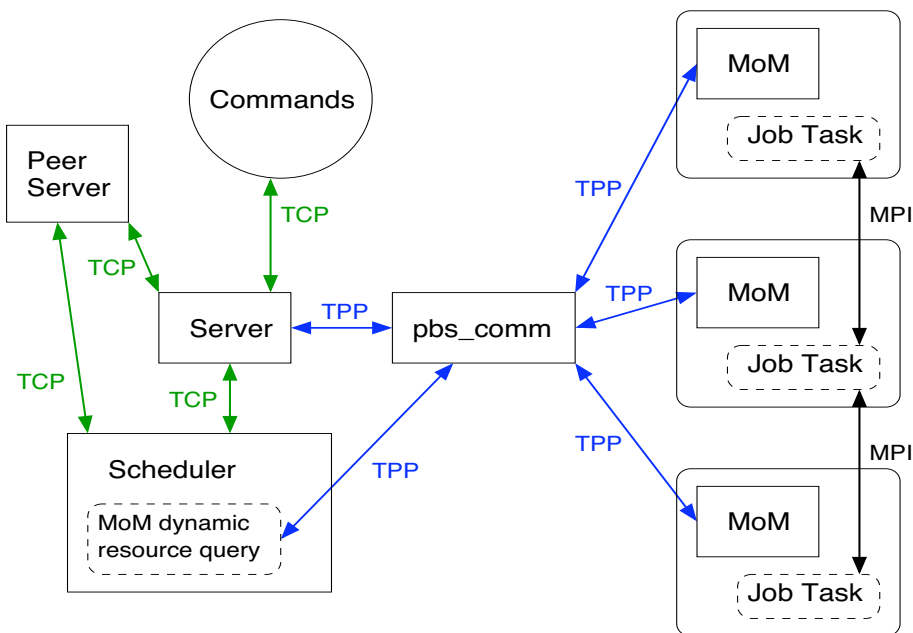


Figure 4-1: Communication Within PBS Complex Using TPP

Communication daemons are connected to each other. If there are multiple `pbs_comms`, and two endpoints on different `pbs_comms` transmit data, communication between endpoints goes from the first endpoint, to the endpoint's configured `pbs_comm` daemon, to the `pbs_comm` configured for the receiving endpoint, to the receiving endpoint.

### 4.5.1 Inter-daemon Connection Behavior Using TPP

When each endpoint starts up, it automatically attempts to connect to the configured or default `pbs_comm` daemon. If the `pbs_comm` daemon is available, the connection attempt succeeds; if not, the endpoint continues to attempt to connect to the `pbs_comm` daemon using a background thread. The order in which endpoints and `pbs_comms` are started is not important. Connections are completed when the `pbs_comm` daemon becomes available. If you have configured multiple `pbs_comms`, each endpoint continues to periodically attempt to connect to each one until all connections succeed.

If the connection from an endpoint to a `pbs_comm` daemon fails, the endpoint attempts to find another already-connected `pbs_comm` daemon to send data via that connection. When the original failed connection is reestablished (via automatic periodic background attempts to connect to the failed daemon) data exchange switches over to the original connection.

When a `pbs_comm` daemon is configured to talk to other `pbs_comms`, it behaves exactly the same way as an endpoint.

Just after you start a MoM, it may not appear to be up, because there is a delay between endpoint connection attempts. The MoM may need up to 30 seconds to show up.

If you have only one communication daemon installed (failover is not configured), and that communication daemon is killed, vnodes become unreachable.

### 4.5.1.1 Sending and Receiving

Endpoints have a built-in retry mechanism to re-send information that has not been acknowledged by the receiver. The receiving endpoint can determine whether it has received duplicate data packets.

### 4.5.1.2 Data Compression

Some jobs cause the server and MoMs to exchange a very large amount of data. The communication daemon automatically compresses the data before communication. In communications, there is usually benefit from compression, because communication is usually CPU-bound, not I/O-bound.

## 4.5.2 Communication Daemon Syntax

### 4.5.2.1 Usage on Linux

On Linux, the `pbs_comm` executable takes the following options:

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

`-r`

Used to specify the 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:

```
<host>[:<port>][,<host>[:<port>]]
```

`-t`

Used to specify the 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:

```
Integer
```

`-N`

The communication daemon runs in standalone mode.

## 4.5.3 Adding Communication Daemons

### 4.5.3.1 Installation Location of Communication Daemons

The `pbs_comm` daemon can be installed on any host that is connected to the PBS complex. By default, a `pbs_comm` is installed on the server host(s), and all endpoints will connect to it (them) by default.

### 4.5.3.2 Configuring Communication Daemons

Make sure that when you configure additional communication daemons, you only point one of each pair of `pbs_comms` to the other; do not point both at 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.

Steps to configure additional `pbs_comms`:

1. Tell each endpoint that goes with the new `pbs_comm` where to find the new `pbs_comm`. Edit the `pbs.conf` file on the endpoint's host, and add:

```
PBS_LEAF_ROUTERS=<host>[:<port>][,<host>[:>port>]]
```

2. For each new `pbs_comm`, tell each new `pbs_comm` about previous `pbs_comms`. Do not tell existing `pbs_comms` about new `pbs_comms`. So if you have an existing `pbs_comm` C1 and add a new `pbs_comm` C2, only point C2 to C1. In `pbs.conf` on C2's host, add:

```
PBS_COMM_ROUTERS=<C1 host>[:<C1 port>]
```

If you add C3, point C3 to both C1 and C2. On C3's host, add:

```
PBS_COMM_ROUTERS=<C1 host>[:<C1 port>],<C2 host>[:<C2 port>]
```

3. Optionally, set the number of threads the new `pbs_comm` will use. The default is 4. We recommend not specifying more threads than processors on the host. In `pbs.conf`, add:

```
PBS_COMM_THREADS=<number of threads>
```

4. Optionally, set the desired log level for the new `pbs_comm`. In `pbs.conf`, add:

```
PBS_COMM_LOG_EVENTS=<log level>
```

5. On the new `pbs_comm` host, tell the init script to start `pbs_comm`. In `pbs.conf`, add:

```
PBS_START_COMM=1
```

1. If you are running a PBS complex that contains both Linux and Windows execution hosts, on any hosts running comms, configure `sssd` so that the users of the Windows domain can log in to the Linux host on which `pbs_server` and `sssd` run. See "[Mixed Linux-Windows Operation](#)" on page 667 in the [PBS Professional Administrator's Guide](#).

For an example, see [section 8.4.5, "Configuring SSSD", on page 382](#). For information on configuring `sssd`, see [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html-single/windows\\_integration\\_guide/index#sssd-ad-proc](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/windows_integration_guide/index#sssd-ad-proc) and <https://access.redhat.com/articles/3023951>.

If you want the Linux host to automatically create a home directory for an Active Directory user if that home directory does not exist at login, you may have to set SELinux to permissive mode. This is optional.

#### 4.5.3.2.i Caveats for Configuring Communication Daemons

When you HUP the communication daemon, it reads only `PBS_COMM_LOG_EVENTS` from `pbs.conf`. If you change any of its other parameters, you must restart the communication daemon:

```
<path to start/stop script>/pbs restart
```

## 4.5.4 Recommendations for Maximizing Communication Performance

You can partition your endpoints so that each group of endpoints has its own `pbs_comm(s)`. Keeping the workload for each `pbs_comm` below the level that degrades performance will speed up your complex. Your site's characteristics determine how many `pbs_comms` you need. Here are some rules of thumb for adding `pbs_comms`:

- One `pbs_comm` per 2000 MoMs, where communication is light
- One `pbs_comm` per rack of ~150 - 200 MoMs, where communication is heavier
- If server start time doubles, add a `pbs_comm`
- If the CPU usage for a `pbs_comm` is above 10 or 15 percent, add a `pbs_comm`
- If performance drops, consider adding a `pbs_comm`

## 4.5.5 Robust Communication with TPP

### 4.5.5.1 Failover and Communication Daemons

When failover is configured, endpoints automatically connect to the `pbs_comm` daemons running on either the primary or secondary PBS server hosts, allowing for communication failover. If both `pbs_comms` are available, communication goes through the `pbs_comm` on the primary server host. If the primary server host fails, communication automatically goes through the `pbs_comm` on the secondary server host. When the primary server host comes back up, communication is automatically resumed by the `pbs_comm` on the primary server host. If failover is configured and the only `pbs_comms` are on the primary and secondary server hosts, and both of those hosts fail, communication between endpoints is unavailable.

### 4.5.5.2 Fault Tolerance

By default, endpoints automatically connect to the `pbs_comm` daemon running at the server host.

You can configure each endpoint to connect to multiple communication daemons. If one of the communication daemons fails, the endpoint can still communicate with the rest of the complex using the alternate communication daemons. When a failed `pbs_comm` comes back online, it automatically resumes handling communications.

If you have configured failover, you have communication fault tolerance to the extent of one of the `pbs_comms` on the primary or secondary server host failing. If you want fault tolerance beyond or instead of failover, you must explicitly install and configure additional `pbs_comm` daemons.

## 4.5.6 Extending Your Complex

To add a new rack to a PBS complex using TCP, take the following steps:

1. Install MoMs as usual on the new execution hosts.
2. Optionally, edit the configuration file on the new MoM hosts to include failover settings.
3. You can configure new MoMs to communicate via existing `pbs_comms`. However, if you are adding many MoMs, we recommend deploying additional `pbs_comms`. Follow the steps in ["Adding Communication Daemons" on page 50](#).
4. Start the daemons in the new rack, and tell the server about the new vnodes:

```
qmgr -c "create node <vnode name>"
```

If you use the IP address for the name of the vnode:

- a. Add `PBS_MOM_NODE_NAME=<IP address>` to `pbs.conf` on the execution host
- b. Restart MoM

## 4.5.7 Changing IP Address of `pbs_comm` Host

To change the IP address of a `pbs_comm` host:

1. Change the IP address of the host
2. Update the DNS
3. Restart `pbs_comm` on that host

Each endpoint or `pbs_comm` periodically retries the connection to each `pbs_comm` that it knows about. When a `pbs_comm` becomes unavailable, all connections to it are automatically retried until they succeed. Endpoint and `pbs_comm` IP addresses to target `pbs_comms` are internally cached for a short time, so if you change the IP address of a target, they will not be able to connect for this time. When this time runs out, endpoints and `pbs_comms` refresh their IP addresses, and connections are reestablished.

## 4.5.8 Configuring Communication for Internal and External Networks

PBS complexes often use an internal network and an external network. PBS clients such as `qsub` and `qstat` communicate to the server over the external network. The daemons communicate with each other over the internal network. In this case, the server host is configured with multiple network interfaces, one for each of the different networks.

The default value of the endpoint's name is the hostname. The TPP network resolves the endpoint's name to the IP address of the machine, and could end up using the external IP address of the host. When this endpoint, for example the server, sends a message to another endpoint, say the MoM, it would embed this external IP address in the message. The MoM detects that this message has arrived from an external IP address and could reject the message, since the MoM is typically configured to use only the internal network and is unaware of the external IP address.

Instead of letting the endpoint use the machine hostname as the endpoint's name (which is the default), set the endpoint's name to a variable that resolves to only the internal network address(es) of the server host. To do that, set the `PBS_LEAF_NAME` `pbs.conf` variable to the internal network name of the host.

The server only queries for the canonicalized address of the MoM host, unless you let it know via the `Mom` attribute; if you have set `PBS_LEAF_NAME` in `/etc/pbs.conf` to something else, make sure you set the `Mom` attribute at `vnode` creation.

## 4.5.9 Troubleshooting Communication with TPP

### New connections are being dropped at a `pbs_comm`

Check whether the `pbs_comm` log has messages saying that the process has exceeded the configured `nfiles` (the open file limit). If so, increase the allowed max open files limit, and restart the `pbs_comm` daemon.

### Message saying `NOROUTE` to destination `xxx:nnn`

The "noroute" message shows the destination address and the `pbs_comm` daemon or endpoint which generated the error. Example:

```
Received noroute to dest ::1:15003, msg:pbs_comm:::1:17001: Dest not found
```

The above message means that the `pbs_comm` running at address `:::1:17001` has responded that the destination address (MoM, in this case) `:::1:15003` is not known to it. This means the MoM at `localhost:15003` was not started (it is down) and/or did not register its address with this `pbs_comm`. Check the MoM logs for that MoM, and see whether it was started, and if so, what addresses it registered and to which `pbs_comm` daemon. These log lines from the `pbs_mom` logs may be useful:

```
Registering address :::1:15003 to pbs_comm
Registering address 192.168.184.156:15003 to pbs_comm
...
...
Connecting to pbs_comm hostname:port
```

The above messages list the actual `pbs_comm` daemon that the MoM or any endpoint is connected to, and when it connected. After connection, it registered the endpoint with the addresses as listed in the "Registering address" messages, before the connect message.

Corresponding to the above messages in the endpoint log, (in this case, MoM ), there should be messages in the associated `pbs_comm` daemon's logs, as follows:

```
tfd=14: Leaf registered address ::1:15003
tfd=14: Leaf registered address 192.168.184.156:15003
```

The above messages mean that a connection from socket file descriptor 14 at the `pbs_comm` daemon received data to register the endpoint with addresses `::1:15003` and `192.168.184.156:15003`.

The above messages from the endpoint and the associated `pbs_comm` daemon tell us whether there are address mismatches, or the endpoints never connected, or connected to the wrong MoMs, or the endpoints are not configured to use TCP.

### MoM down/stale on `pbsnodes -av` output

- Check whether the respective MoM is actually up.
- Check that the MoM that is showing as down is actually pointing to the correct `pbs_comm` daemon, by checking whether it is the default or `PBS_LEAF_ROUTERS` is set.
- Check that the `pbs_comms` that are handling the `pbs_server` and the MoM in question are running, and that none of them have a system error in their logs such as no files etc.
- Check the connection settings between this pair of `pbs_comms` is as intended. Check each of the `pbs_comm`'s `PBS_COMM_ROUTERS` settings.
- Follow a "noroute" message to trace where the "noroute" is originating, and troubleshoot why the route is not being found .

## 4.5.10 Logging and Errors with TPP

### 4.5.10.1 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 shared as part of the shared `PBS_HOME` 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 `yyyymmdd` (the same as for other PBS daemons).

Whenever a new log file is opened, the communication daemon logs `PBS_LEAF_NAME`, `PBS_MOM_NODE_NAME`, and the hostname. The daemon also logs all network interfaces, listing each interface and all of the hostnames associated with that interface. In addition, it logs the PBS version and the build information.

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-time;event\_code;daemon\_name(thread number);object\_type;object\_name;message*

An example is as follows:

```
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
```



---

### 4.5.10.2 Messages from Endpoints

Connected to pbs\_comm %s

Endpoint was able to connect to the named pbs\_comm daemon.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

Connection to pbs\_comm %s down

The endpoint's connection to the specified pbs\_comm daemon is down.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

Connection to pbs\_comm %s failed

The endpoint failed to connect to the specified pbs\_comm daemon. A system/socket error message may accompany this message.

Log level: *PBSEVENT\_ERROR*

Registering address %s to pbs\_comm

The endpoint logs the list of IP addresses it is registering with the pbs\_comm daemon.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

sd %d, Received noroute to dest %s, msg:%s

Specified stream sd (stream descriptor) has received a "noroute" message from the pbs\_comm daemon indicating that the destination is not known to the pbs\_comm daemon. An additional message from pbs\_comm is also printed.

Log level: *PBSEVENT\_ERROR*

Single pbs\_comm configured, TPP Fault tolerant mode disabled

Only one pbs\_comm daemon was configured, so fault tolerant mode is disabled.

Log level: *PBSEVENT\_SYSTEM* | *PBSEVENT\_ADMIN*

---

### 4.5.10.3 Messages from Communication Daemons

tfd=%d: endpoint registered address %s

Endpoint registered this address.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

Connection from leaf %s, tfd=%d down

The connection from an endpoint just went down.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

pbs\_comm %s connected

Another pbs\_comm daemon connected to this pbs\_comm daemon.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

pbs\_comm %s accepted connection

Specified pbs\_comm daemon accepted connection from this pbs\_comm.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

pbs\_comms should have at least 2 threads

Number of threads configured for the daemon is too few. There should be a minimum of two threads. The daemon will abort.

Log level: *PBSEVENT\_SYSTEM* | *PBSEVENT\_ADMIN* | *PBSEVENT\_FORCE*

Received TPP\_CTL\_NOROUTE for message, %s(sd=%d) -> %s: %s

The pbs\_comm daemon received a "noroute" message from a destination endpoint. This means that the destination stream was not found in that endpoint.

Log level: *PBSEVENT\_ERROR*

Connection from non-reserved port, rejected

The pbs\_comm received a connection request from an endpoint or another pbs\_comm or an endpoint, but since the connection originated from a non-reserved port, it was not accepted.

Log level: *PBSEVENT\_ERROR*

Failed initiating connection to pbs\_comm %s

This pbs\_comm daemon failed to initiate a connection to another pbs\_comm.

Log level: *PBSEVENT\_ERROR*

---

### 4.5.10.4 Important Messages from Communication or Other Daemons

#### Compression failed

Compression routine failed. Usually due to memory constraints.

Log level: PBSEVENT\_SYSTEM | PBSEVENT\_ADMIN | PBSEVENT\_FORCE

#### Decompression failed

Decompression routine failed due to bad input data. Usually a transmission/network error.

Log level: PBSEVENT\_SYSTEM | PBSEVENT\_ADMIN | PBSEVENT\_FORCE

#### Error %d resolving %s

There was an error in name resolution of a hostname.

Log level: PBSEVENT\_SYSTEM | PBSEVENT\_ADMIN | PBSEVENT\_FORCE

#### Error %d while binding to port %d

There was an error in binding to the specified port. Usually this means the address is already in use.

Log level: PBSEVENT\_SYSTEM | PBSEVENT\_ADMIN | PBSEVENT\_FORCE

#### No reserved ports available

No more reserved ports are available. Cannot initiate connection to a `pbs_comm` daemon. Not applicable on Windows.

Log level: PBSEVENT\_ERROR

#### Out of memory <in an operation>

An out-of-memory condition occurred.

Log level: PBSEVENT\_SYSTEM | PBSEVENT\_ADMIN | PBSEVENT\_FORCE

### 4.5.10.5 Informational Messages from Communication or Other Daemons

Initializing TPP transport Layer

Starting the initialization of the TPP layer: starting threads etc.; creating internal data structures.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

TPP initialization done

Initialization completed successfully; system ready to transmit data.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

Shutting down TPP transport Layer

TPP was asked to shut down.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

Max files allowed = %ld

Logs the *nfiles* currently configured.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

Max files too low - you may want to increase it

If *nfiles* is <1024, the *pbs\_comm* daemon emits the message. If *nfiles* configured is <100, the startup aborts. Usually *nfiles* must be configured to allow the number of connections (usually the number of MoMs) the *pbs\_comm* process is going to handle.

Log level: *PBSEVENT\_SYSTEM* | *PBSEVENT\_ADMIN*

Thread exiting, had %d connections

Each thread in the TPP layer logs the number of connections it was handling. For *pbs\_comm*, this is usually the number of MoMs that were handled by each thread. This gives you information useful for deciding when to increase the threads in order to distribute the load.

Log level: *PBSEVENT\_DEBUG* | *PBSEVENT\_DEBUG2*

## 4.6 Ports Used by PBS

PBS daemons listen for inbound connections at specific network ports. These ports have defaults, but can be configured if necessary. PBS daemons use any ports numbered less than 1024 for outbound communication. For PBS daemon-to-daemon communication over TCP, the originating daemon will request a privileged port for its end of the communication.

PBS makes use of fully-qualified host names for identifying jobs and their locations. A PBS installation is known by the host name on which the server is running. The canonical host name is used to authenticate messages, and is taken from the primary name field, *h\_name*, in the structure returned by the library call `gethostbyaddr()`. According to the IETF RFCs, this name must be fully qualified and consistent for any IP address assigned to that host.

Port numbers can be set via `/etc/services`, the command line, or in `pbs.conf`. If not set by any of these means, they will be set to the default values. The PBS components and the commands will attempt to use the system `services` file to identify the standard port numbers to use for communication. If the port number for a PBS daemon can't be found in the system file, a default value for that daemon will be used. The server, scheduler, and MoM daemons have startup options for setting port numbers. In the PBS Professional Reference Guide, see "[pbs\\_mom](#)" on page 72, "[pbs\\_sched](#)" on page 106, and "[pbs\\_server](#)" on page 109.

For port settings in `pbs.conf`, see "[Contents of Configuration File](#)" on page 371 in the [PBS Professional Installation & Upgrade Guide](#).

The scheduler uses any privileged port (less than 1024) as the outgoing port to talk to the server.

Under Linux, the `services` file is named `/etc/services`.

Under Windows, it is named `%WINDIR%\system32\drivers\etc\services`.

The port numbers listed are the default numbers used by PBS. If you change them, be careful to use the same numbers on all systems. The port number for `pbs_resmon` must be one higher than for `pbs_mom`.

## 4.6.1 Ports Used by PBS in TPP Mode

The table below lists the default port numbers for PBS daemons in TPP mode:

**Table 4-1: Ports Used by PBS Daemons in TPP Mode**

Daemon Listening at Port	Port Number	Protocol	Type of Communication
<code>pbs_server</code>	<i>15001</i>	TPP (TCP)	All communication to server
<code>pbs_mom</code>	<i>15002</i>	TPP (TCP)	All communication to MoM
<code>pbs_resmon</code>	<i>15003</i>	TPP (TCP)	Scheduler-MoM resource requests ( <code>pbs_resmon</code> listens on this port)
<code>pbs_sched</code>	<i>15004</i>	TPP (TCP)	All communication to scheduler
<code>pbs_datastore</code>	<i>15007</i>	proprietary	PBS information storage and retrieval
License server	<i>6200</i>	proprietary	All communication to license server
<code>pbs_comm</code>	<i>17001</i>	TPP (TCP)	All communication to <code>pbs_comm</code>

## 4.6.2 Port Settings in `pbs.conf`

You can set the following in `pbs.conf`:

**Table 4-2: Port Parameters in `pbs.conf`**

Parameter	Description
<code>PBS_BATCH_SERVICE_PORT</code>	Port server listens on
<code>PBS_BATCH_SERVICE_PORT_DIS</code>	DIS port server listens on
<code>PBS_DATA_SERVICE_PORT</code>	Used to specify non-default port for connecting to data service. Default is 15007.
<code>PBS_MANAGER_SERVICE_PORT</code>	Port MoM listens on
<code>PBS_MOM_SERVICE_PORT</code>	Port MoM listens on
<code>PBS_SCHEDULER_SERVICE_PORT</code>	Port scheduler listens on

## 4.7 PBS with Multihomed Systems

PBS expects the network to function according to IETF standards. Please make sure that your addresses resolve correctly. You can set host name parameters in `pbs.conf` to disambiguate addresses for contacting the server, sending mail, delivering output and error files, and establishing outgoing connections.

When setting these parameters, use fully qualified host names where you could have host name collisions, for example `master.foo.example.com` and `master.bar.example.com`. See the following sections for details.

Before tackling this section, make sure that you have taken care of everything listed in [section 2.1.3, "Name Resolution and Network Configuration", on page 8](#).

PBS uses only IPv4, so all names must resolve to IPv4 addresses.

### 4.7.1 Contacting the Server

Use the `PBS_SERVER_HOST_NAME` parameter in `pbs.conf` on each host in the complex to specify the FQDN of the server, under these circumstances:

- The host on which the PBS server runs has multiple interfaces and some of these interfaces are limited to a private network that might not be addressable outside of the immediate complex
- The server name to be used in Job IDs needs to be different from the `PBS_SERVER` parameter. It might become impossible for a client to contact the server where this option is not used or is misconfigured. Take extreme care when using `PBS_SERVER_HOST_NAME` for this reason.

You can specify the server name with the following order of precedence, highest first:

- Specifying server name at the client
  - Specifying server name at the command line, e.g. `pbsnodes -s <server name>`
  - Setting the `PBS_PRIMARY` and `PBS_SECONDARY` environment variables
  - Setting the `PBS_SERVER_HOST_NAME` environment variable
  - Setting the `PBS_SERVER` environment variable
- Setting `PBS_PRIMARY` and `PBS_SECONDARY` in `pbs.conf`
- Setting `PBS_SERVER_HOST_NAME` in `pbs.conf`
- Setting `PBS_SERVER` in `pbs.conf`

### 4.7.2 Delivering Output and Error Files

You can specify the host name portions of paths for standard output and standard error for jobs. To specify the host where the job's standard output and error files are delivered, use the `PBS_OUTPUT_HOST_NAME` parameter in `pbs.conf` on the server host. It is useful when submission and execution hosts are not visible to each other.

- If the job submitter specifies an output or error path with both file path and host name, PBS uses that path.
- If the job submitter specifies an output or error path containing only a file path:
  - If `PBS_OUTPUT_HOST_NAME` is set, PBS uses that as the host name portion of the path
  - If `PBS_OUTPUT_HOST_NAME` is not set, PBS follows the rules in ["Default Behavior For Output and Error Files", on page 40 of the PBS Professional User's Guide](#).
- If the job submitter does not specify an output or error path, PBS uses the current working directory of `qsub`, following the naming rules in ["Default Paths for Output and Error Files", on page 42 of the PBS Professional User's Guide](#), and appends an at sign ("`@`") and the value of `PBS_OUTPUT_HOST_NAME`.

---

### 4.7.3 When Installing and Upgrading

During installation or upgrade:

1. When asked whether you want to start the new version of PBS, reply "no"
2. Edit `pbs.conf` to set the desired network address parameters
3. Start the new version of PBS:

```
systemctl start pbs
```

or

```
<path to script>/pbs start
```

You may see differences in new job IDs. For example, if the prior value of `PBS_SERVER` was set to the fully qualified host name, the existing jobs will have IDs containing the full hostname, for example `123.server.example.com`. If the current value of `PBS_SERVER` is a short name, then new jobs will have IDs with the short form of the host name, in this case, `123.server`.

With version 13.0, PBS supports host names up to 255 characters. The format of the job files written by `pbs_mom` has changed due to this. If there are existing job files during an overlay upgrade, PBS prints a summary message showing the number of job files successfully upgraded and the total number of job files. For each job file that was not successfully upgraded, PBS prints a message that the job file was not successfully upgraded and gives the full path to that job file.

## 4.7.4 Hostname Parameters in pbs.conf

The following table describes the hostname parameters in the `pbs.conf` configuration file:

**Table 4-3: Hostname Parameters in pbs.conf**

Parameter	Description
PBS_LEAF_NAME	<p>Tells endpoint what hostname to use for network.</p> <p>The value does not include a port, since that is usually set by the daemon.</p> <p>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.</p> <p>The server only queries for the canonicalized address of the MoM host, unless you let it know via the <code>Mom</code> attribute; if you have set <code>PBS_LEAF_NAME</code> in <code>/etc/pbs.conf</code> to something else, make sure you set the <code>Mom</code> attribute at vnode creation.</p> <p>TPP internally resolves the name to a set of IP addresses, so you do not affect how <code>pbs_comm</code> works.</p>
PBS_MAIL_HOST_NAME	<p>Optional. Used in addressing mail regarding jobs and reservations that is sent to users specified in a job or reservation's <code>Mail_Users</code> attribute. See <a href="#">"Specifying Mail Delivery Domain" on page 22 in the PBS Professional Administrator's Guide</a>.</p> <p>Should be a fully qualified domain name. Cannot contain a colon (":").</p>
PBS_MOM_NODE_NAME	<p>Name that MoM should use for parent vnode, and if they exist, child vnodes. If this is not set, MoM defaults to using the non-canonicalized hostname returned by <code>gethostname()</code>.</p> <p>If you use the IP address for a vnode name, set <code>PBS_MOM_NODE_NAME=&lt;IP address&gt;</code> in <code>pbs.conf</code> on the execution host.</p> <p>This parameter cannot contain dots unless it is for an IP address.</p>
PBS_OUTPUT_HOST_NAME	<p>Optional. Host to which all job standard output and standard error are delivered. See <a href="#">section 4.7.2, "Delivering Output and Error Files", on page 60</a>.</p> <p>Should be a fully qualified domain name. Cannot contain a colon (":").</p>
PBS_PRIMARY	<p>Hostname of primary server. Overrides <code>PBS_SERVER_HOST_NAME</code>.</p>
PBS_SECONDARY	<p>Hostname of secondary server. Overrides <code>PBS_SERVER_HOST_NAME</code>.</p>
PBS_SERVER	<p>Hostname of host running the server. Cannot be longer than 255 characters. If the short name of the server host resolves to the correct IP address, you can use the short name for the value of the <code>PBS_SERVER</code> entry in <code>pbs.conf</code>. If only the FQDN of the server host resolves to the correct IP address, you must use the FQDN for the value of <code>PBS_SERVER</code>.</p> <p>Overridden by <code>PBS_SERVER_HOST_NAME</code> and <code>PBS_PRIMARY</code>.</p>
PBS_SERVER_HOST_NAME	<p>Optional. The FQDN of the server host. Used by clients to contact server. See <a href="#">section 4.7.1, "Contacting the Server", on page 60</a>.</p> <p>Should be a fully qualified domain name. Cannot contain a colon (":").</p>



# Initial Configuration

After you have installed PBS Professional, perform the following steps:

## 5.1 Validate the Installation

- Check files and directories: To validate the installation of PBS Professional, at any time, run the `pbs_probe` command. It will review the installation (installed files, directory and file permissions, etc) and report any problems found. For details, see ["pbs\\_probe" on page 82 of the PBS Professional Reference Guide](#).

The `pbs_probe` command is not available under Windows.

- Check PBS version. Use the `qstat` command to find out what version of PBS Professional you have:  
`qstat -fB`
- Check hostname resolution:
  - At the server, use the `pbs_hostn` command with the name of each host in the complex. This should complain if hostname resolution is not working correctly. See ["pbs\\_hostn" on page 64 of the PBS Professional Reference Guide](#).
  - Make sure that `rcp` and/or `scp` work correctly. They must work outside of PBS before PBS can use them. Run `rcp` and/or `scp` between machines in the complex to make sure they work. If there are problems, see [section 2.1.3, "Name Resolution and Network Configuration", on page 8](#).
- Windows: turn firewall off for execution hosts: see ["Windows Firewall" on page 397 in the PBS Professional Administrator's Guide](#)

## 5.2 Support PBS Features

- Configure PBS inter-daemon communication. See [Chapter 4, "Communication", on page 45](#).
- Define PATHs for users: set paths for all users to include PBS commands and man pages. For paths, see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Administrator commands are in `PBS_EXEC/sbin`, and user commands are in `PBS_EXEC/bin`. Man pages are `PBS_HOME/man`.
- Support X forwarding:
  - Edit each MoM's `PATH` variable to include the directory containing the `xauth` utility.
  - Add the path to the `xauth` binary to each MoM's `pbs_environment` file. For example, if you start with this path:  
`/bin:/user/bin`  
and the `xauth` utility is here:  
`/usr/bin/X11/xauth`  
The entry in the `pbs_environment` file would be the following:  
`PATH=/bin:/usr/bin:/usr/bin/X11`
  - In the `ssh_config` file for each machine that will use X forwarding, put this line:  
`ForwardX11Trusted yes`

---

X forwarding is not available under Windows.

- Allow interactive jobs. For interactive jobs, MoMs establish a connection back to the submission host:
  - Make sure that the ephemeral port range in your firewall is open (make sure that MoMs can connect to an ephemeral port on submission hosts). Check your OS documentation for the correct range.
  - Allow interactive jobs under Windows: see ["Allowing Interactive Jobs on Windows" on page 535 in the PBS Professional Administrator's Guide](#)
- Create and configure vnodes: see ["About Vnodes: Virtual Nodes" on page 39 in the PBS Professional Administrator's Guide](#)
- Create and configure queues: see ["Queues" on page 23 in the PBS Professional Administrator's Guide](#)
- Manage cgroups and cpusets: see ["Configuring and Using PBS with Cgroups" on page 573 in the PBS Professional Administrator's Guide](#)
- Configure resources: see ["Using PBS Resources" on page 231 in the PBS Professional Administrator's Guide](#)
- Set up resource limits: see ["Managing Resource Usage" on page 290 in the PBS Professional Administrator's Guide](#)
- Define scheduling policy: see ["Scheduling" on page 55 in the PBS Professional Administrator's Guide](#)
- Create hooks: see the PBS Professional Hooks Guide.
- Integrate with an MPI: see ["Using MPI with PBS" on page 453 in the PBS Professional Administrator's Guide](#)
- Use containers: see ["Configuring PBS for Containers" on page 623 in the PBS Professional Administrator's Guide](#)
- Use provisioning: see ["Provisioning" on page 329 in the PBS Professional Administrator's Guide](#)
- Set up failover: see ["Failover" on page 401 in the PBS Professional Administrator's Guide](#)
- Set up checkpointing: see ["Checkpoint and Restart" on page 420 in the PBS Professional Administrator's Guide](#)
- Minimize communication problems: see ["Preventing Communication and Timing Problems" on page 443 in the PBS Professional Administrator's Guide](#)
- Manage security features, including authentication and encryption: see ["Security" on page 361 in the PBS Professional Administrator's Guide](#)
  - Required on Windows: set up encryption via TLS. See ["Encrypting PBS Communication" on page 389 in the PBS Professional Administrator's Guide](#).
- Set up desired file transfer mechanism: see ["Setting File Transfer Mechanism" on page 561 in the PBS Professional Administrator's Guide](#)
- Configure where PBS components will put temporary files: see ["Temporary File Location for PBS Components" on page 569 in the PBS Professional Administrator's Guide](#)

# 6

# Upgrading

This chapter shows how to upgrade from a previous version of PBS Professional. If PBS Professional is not installed on your system, go instead to [Chapter 3, "Installation", on page 19](#).

## 6.1 Types of Upgrades

There are two types of upgrades available for PBS Professional:

### overlay upgrade

Installs the new PBS\_HOME and PBS\_EXEC on top of the old ones. Jobs cannot be running during an overlay upgrade.

### migration upgrade

You install the new PBS\_HOME and PBS\_EXEC in a separate location from the old PBS\_HOME and PBS\_EXEC. The new PBS\_HOME can be in the standard location if the old version has been moved. Jobs are moved from the old server to the new one, and cannot be running during the move.

### 6.1.1 Choosing Upgrade Type on Linux

Usually, you can do an overlay upgrade on Linux systems. However, the following require migration upgrades:

- When moving between hosts
- When upgrading from an open-source version of PBS Professional
- When certain European or Japanese characters are stored in the data store

For specific upgrade recommendations and updates, see the Release Notes.

### 6.1.2 Upgrading Existing All-Windows Complex

If your existing complex runs a PBS server on a Windows host, "upgrading" means doing a fresh install for the server/schedulers/comms, and upgrading your Windows MoMs. You cannot preserve any jobs in any state during the upgrade. See [Chapter 6, "Upgrading from an All-Windows Complex", on page 125](#).

### 6.1.3 Upgrading from Windows/Linux Combination to Windows/Linux Combination

Upgrading on Windows/Linux requires a migration upgrade; see [section 6.8, "Upgrading a Windows/Linux Complex", on page 109](#).

---

## 6.2 Differences from Previous Versions

### 6.2.1 New Way to Manage Vnode Attributes

As of version 2020.1, PBS can use the `cgroups` hook to manage `cpusets` and create child `vnodes` on multi-vnode machines.

If you use the `cgroups` hook on a host where you want to set the `sharing` attribute or define the placement sets, you can use an `exehost_startup` hook or a Version 2 configuration file for this, but make sure that you refer precisely to the `vnodes` that were created by the `cgroups` hook. Do not accidentally create new `vnodes` by defining them (that is, using a `vnode` name unknown to the `cgroups` hook).

### 6.2.2 New Scheduler Attributes

The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Schedulers now have a `log_events` attribute that replaces the `log_filter` scheduler parameter. You use `qmgr` to set these attributes.

### 6.2.3 Using RPM Instead of `INSTALL` (14.2)

You use RPM or another native package manager such as `yum` or `zypper` to install PBS, instead of the `INSTALL` script.

### 6.2.4 Using `systemd` Instead of Start/stop Script (14.2)

PBS uses `systemd` instead of the PBS start/stop script on Linux platforms that support `systemd`. On Linux platforms that do not support `systemd`, PBS still uses the start/stop script. You will see a choice of instructions for starting or stopping PBS.

### 6.2.5 Automatic Upgrade of Database (13.0)

The PBS installer automatically upgrades the database used by PBS for its data store.

### 6.2.6 Installing Communication Daemon (13.0)

As of 13.0, PBS uses a new daemon, `pbs_comm`, for communication. One communication daemon is automatically installed on each server host, and all daemons automatically connect to it. If you require additional communication daemons, you must install and configure them. See [section 4.5.3, "Adding Communication Daemons", on page 50](#).

### 6.2.7 Default Location of `PBS_EXEC` and `PBS_HOME`

`PBS_EXEC` is the directory that contains the PBS binaries. The default location for `PBS_EXEC` is `/opt/pbs`. `PBS_HOME` is the directory where PBS information is stored. The default location for `PBS_HOME` is `/var/spool/pbs`.

---

## 6.2.8 Use PBS Start Script or `systemd` During Overlay Upgrade

During an overlay upgrade, you must start the PBS server using `systemd` on platforms that support it, or the start/stop script where `systemd` is not supported, so that the server is initialized correctly. The instructions in this manual for overlay upgrading specify using `systemd` or the start script.

## 6.3 Caveats and Advice

### 6.3.1 Licensing

PBS starts faster if you install, configure, and start the Altair license server before starting PBS. We recommend that you follow the steps for installing and starting the license server before upgrading. See the *Altair License Management System Installation and Operations Guide*, available at [www.altair.com](http://www.altair.com). Do not attempt to use any license server other than the Altair license server.

### 6.3.2 Making Time to Upgrade

If you want to avoid having to work around running jobs when you perform an upgrade, you can set PBS up so that there are no running jobs when you want to do the upgrade. Follow these steps:

1. Figure out how much walltime the longest-running jobs are likely to need, e.g. two weeks
2. Pick a time further into the future than that, e.g. 3 weeks
3. On all PBS hosts, create dedicated time or a reservation for the amount of time you think the upgrade will require, e.g. a day
  - You can use a dedicated time slot, making it so that no jobs will be scheduled for that dedicated time. The system can be shut down all at once at the start of the dedicated time. See ["Dedicated Time" on page 127 in the PBS Professional Administrator's Guide](#).
  - You can create a reservation that reserves an entire host by using `-l place=exclhost`. The following reservation creates a reservation for the host `mars`, from 10am to 10pm:

```
pbs_rsub -R 1000 -D 12:00:00 -l select = host=mars -l place=exclhost
```

For more on creating reservations, see ["pbs\\_rsub" on page 97 of the PBS Professional Reference Guide](#).

### 6.3.3 Upgrading Database

PBS automatically upgrades the database used for its data store. If the process of upgrading the database fails, you must restore the database to its pre-upgrade state in order to upgrade PBS.

### 6.3.4 Data Service Account Must Be Same as When Installed

The data service account you use when upgrading PBS must be the same as when you installed the old version of PBS, otherwise the upgrade will fail. The workaround is to change the data service user ID to the ID used for installation of the old PBS data service, perform the upgrade, then change the ID back.

1. Identify the user who originally created the data store:

- a. Log in to the data store:

```
su - <data service account> -s /bin/sh -c "LD_LIBRARY_PATH=$PBS_EXEC/pgsql/lib
    $PBS_EXEC/pgsql/bin/psql -U <data service account> -p <data service port> -d
    pbs_datastore"
```

The default data service port is *15007*

The default data service account is *pbsdata*

- b. Run a query to get the list of users in the database:

```
pbs_datastore=# select pg_authid.oid, rolname from pg_authid;
```

```
oid | rolname
```

```
-----+-----
```

```
10 | pbsdata
```

```
16541 | <username>
```

```
(2 rows)
```

- c. Find the original user who created the database:

```
pbs_datastore=# select pg_authid.oid, rolname from pg_authid where pg_authid.oid=10;
```

```
oid | rolname
```

```
----+-----
```

```
10 | pbsdata
```

```
(1 row)
```

2. Exit the database
3. Create the original data service account in system if it is not available.
4. Update the current database user to the original data service account.

```
pbs_ds_password -C <original username>
```

5. Perform the overlay upgrade

6. Reset the current database user to desired username:

```
pbs_ds_password -C <later username>
```

See "[Setting Data Service Account Name and Password](#)" on page 560 in the *PBS Professional Administrator's Guide*.

### 6.3.5 Updating Hooks for New Python Version

As of version 19.4.1, PBS uses Python 3.6, so you need to make sure that your hooks and their configuration files are compatible with Python 3.6. To do this, you export each hook and configuration file in ASCII format, make sure it is compliant with Python 3.6, then import the 3.6-ready hook and configuration file in ASCII format. We include a link to a site with instructions for making your Python code compatible with version 3.6. We include all of these steps in the instructions.

### 6.3.6 New Server Requires New MoMs

As of version 12.0, you must **not** attempt to run a newer server with older MoMs. You must start the new server only when all MoMs have been upgraded. Follow the steps in this chapter.

### 6.3.7 Do Not Unset `default_chunk.ncpus`

Do not unset the value for the `default_chunk.ncpus` server attribute. It is set by the server to `1`. You can set it to another non-zero value, but a value of `0` will produce undefined behavior. When the PBS server initializes and the `default_chunk` server attribute has not been specified during a prior run, the server will internally set the following:

```
default_chunk.ncpus=1
```

This ensures that each "chunk" of a job's select specification requests at least one CPU.

If you explicitly set the `default_chunk` server attribute, that setting will be retained across server restarts.

### 6.3.8 Unset `PBS_EXEC` Environment Variable

Make sure that the `PBS_EXEC` environment variable is unset.

### 6.3.9 Saving and Re-creating Vnode Configuration

For an overlay upgrade, you do not need to save and re-create vnodes. For a migration upgrade, you can save your vnode configuration and re-create it using this sequence:

```
qmgr -c 'print node @default' > nodes.new
<clean up nodes.new>
qmgr < nodes.new
```

Why clean up `nodes.new` before reading it back in?

- PBS (the cgroups hook or MoM) should create all child vnodes (vnodes that are not parent vnodes). If you create these child vnodes using `qmgr`, you can end up with duplicate vnode objects.
- The `state` attribute and the `arch`, and `host`, and `vnode` resources are set automatically while creating vnodes. Do not set them explicitly. Doing so can get you into trouble especially if you are changing how hostname resolution works.
- The `qmgr` command overrides resource settings in Version 2 configuration files. If you use `qmgr` to set vnode resources, you can't set them later in Version 2 configuration files.
- MoM reports `mem`, `vmem` and `ncpus`. You can use `qmgr` to set these if they need to be explicitly set; otherwise, don't include these lines in `nodes.new`.
- Leave only the creation lines for parent vnodes and any resources you want managed on the server side through `qmgr`.

We include this step in the upgrading instructions; we explain why here.

### 6.3.10 Upgrading with Failover

If you are upgrading and using failover, do not start the new secondary server until the new primary has finished starting.

If your secondary server has a STONITH script, before you perform an upgrade, prevent the STONITH script from running by setting its permissions to `0644`. After the upgrade, you can set the permissions back to `0755`. We include these steps in the upgrade instructions.

## 6.4 Introduction to Upgrading Under Linux

When you get your new version of PBS, unpack it (unzip, untar) as a non-privileged user. When you follow the upgrading instructions below, all of the steps should be performed as root.

### 6.4.1 Directories

The location of `PBS_HOME` is specified in the file `/etc/pbs.conf`, but defaults to `/var/spool/pbs` if not specified. The default for `PBS_EXEC` is `/opt/pbs`. You can specify a non-default location for `PBS_EXEC` via the `--prefix` option to `rpm` when installing the new PBS.

### 6.4.2 Upgrading on Multiple Machines

Instead of running the installer by hand on each machine, you can use a command such as `pdsh`. The one-line format for a non-default install is:

```
PBS_SERVER=<server name> PBS_HOME=<new/home/location/pbs> rpm -i --prefix <new/exec/location/pbs>
pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 6.4.3 Upgrading on a Machine Running the Cpuset MoM

Machines running the cpuset MoM typically include HPE MC990X, HPE Superdome Flex, or HPE 8600.

When upgrading on a machine running the cpuset MoM, follow the instructions in [section 6.6, "Overlay Upgrade on One or More Machines Running Cpuset MoM"](#), on page 82.

## 6.5 Overlay Upgrade Under Linux

The steps in this section are for machines that are not running a cpuset MoM. Machines running the cpuset MoM typically include HPE MC990X, HPE Superdome Flex, or HPE 8600. When upgrading on a machine running the cpuset MoM, follow the instructions in [section 6.6, "Overlay Upgrade on One or More Machines Running Cpuset MoM"](#), on page 82.

The following commands must be run as root.

### 6.5.1 Prevent Jobs From Being Started

Prevent the scheduler(s) from starting jobs. Set scheduling to *false* for the default scheduler and each multisched:

```
qmgr -c "set sched <scheduler name> scheduling = false"
```

### 6.5.2 Allow Running Jobs to Finish, or Requeue Them

You cannot perform an upgrade while jobs are running. Either let running jobs finish, or requeue them. (You can also delete them.)



---

To requeue any running jobs:

1. List the jobs. This will list some jobs more than once. You only need to requeue each job once:

```
pbsnodes <hostname> | grep jobs
```

2. Requeue the jobs:

```
qrerun <job ID> <job ID> ...
```

To kill the jobs:

1. List the jobs. This will list some jobs more than once. You only need to kill each job once:

```
pbsnodes <hostname> | grep jobs
```

2. Use the `qdel` command to kill each job by job ID:

```
qdel <job ID> <job ID> ...
```

To drain the host, wait until any running jobs have finished.

Make sure that there are no old job files on any execution hosts. Remove any of the following:

```
$PBS_HOME/mom_priv/jobs/*.JB
```

### 6.5.3 Disable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, disable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

### 6.5.4 Disable STONITH Script

If your secondary server has a STONITH script, prevent the STONITH script from running by setting its permissions to 0644.

### 6.5.5 Unwrap Any Wrapped MPIs

If you used the `pbsrun_wrap` mechanism with your old version of PBS, you must first unwrap any MPIs that you wrapped. This includes MPICH-GM, MPICH-MX, MPICH2, etc. You can re-wrap your MPIs after upgrading PBS.

For example, you can unwrap an MPICH2 MPI:

```
# pbsrun_unwrap pbsrun.mpich2_64
```

See "[pbsrun\\_unwrap](#)" on page 51 of the [PBS Professional Reference Guide](#).

## 6.5.6 Save Execution Host Configuration Information

On each PBS execution host, copy the Version 1 and Version 2 configuration files:

1. Make a backup directory:
 

```
mkdir /tmp/pbs_mom_backup
```
2. Make a copy of the Version 1 configuration file:
 

```
cp $PBS_HOME/mom_priv/config /tmp/pbs_mom_backup/config.backup
```
3. Make a copy of the Version 2 configuration files:
 

```
mkdir /tmp/pbs_mom_backup/mom_configs
pbs_mom -s list | egrep -v '^PBS' | while read file
do
  pbs_mom -s show file > /tmp/pbs_mom_backup/mom_configs/$file
done
```

## 6.5.7 Save Hooks and Hook Configuration Files

Save your hooks and hook configuration files in ASCII format so you can check them and import them later. The new version of PBS includes a new `pbs_cgroups` hook with a new configuration file. If you use the `cgroups` hook, you must use the new hook and configuration file, but you may want to modify the configuration file, so if you have made any changes to your existing `pbs_cgroups` hook configuration file, you need to save it before you upgrade. Later, you can use the saved information to modify the new configuration file.

For **each** hook:

1. Save the hook. Export the hook:
 

```
# qmgr -c 'export hook <hook name> application/x-python default /tmp/<hook name>.old2.7'
```
2. Save your hook configuration file. Export the configuration file:
 

```
# qmgr -c 'export hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

## 6.5.8 Update Hooks and Hook Configuration Files for New Python

PBS 19.4.1 and later uses Python 3.6, so if you have not already, update all of your site-defined hooks (not the built-in hooks) to Python 3.6. For **each** hook except for the `pbs_cgroups` hook:

1. Update your hook to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your updated hook file differently; use something like `"/tmp/<hook name>.new3.6"`
2. Check that the contents of the configuration file are correct for Python 3.6

## 6.5.9 Shut Down Your Existing PBS

1. Shut down the server(s), default scheduler, and MoMs:
 

```
qterm -t immediate -m -s -f
```

If your server is not running in a failover environment, the "-f" option is not required.

2. Shut down any multischeds. On each multisched host:
  - a. Find the PID you want:
 

```
ps -ef | grep pbs_sched
```

For the default scheduler, you'll see "pbs\_sched", but for multischeds, you'll see "pbs\_sched -I <multisched name>".
  - b. Stop the scheduler or multisched:
 

```
kill <multisched PID>
```
3. On the server host and any other comm hosts, shut down the communication daemon:
 

```
systemctl stop pbs
```

or

```
<path to script>/pbs stop
```
4. Verify that PBS daemons are not running in the background:
 

```
ps -ef | grep pbs
```

If you see the pbs\_server, pbs\_sched, pbs\_mom, or pbs\_comm process running, manually terminate that process. If using failover, check both primary and secondary server hosts:

```
kill -9 <daemon PID>
```

## 6.5.10 Back Up Existing PBS Files

On each PBS host, make a tar file of the PBS\_HOME and PBS\_EXEC directories.

1. Make a backup directory:
 

```
mkdir /tmp/pbs_backup
```
2. Make a tar file of PBS\_HOME:
 

```
cd $PBS_HOME/..
tar -cvf /tmp/pbs_backup/PBS_HOME_tarbackup.tar $PBS_HOME
```
3. Make a tar file of PBS\_EXEC:
 

```
cd $PBS_EXEC/..
tar -cvf /tmp/pbs_backup/PBS_EXEC_tarbackup.tar $PBS_EXEC
```
4. Make a copy of your configuration file:
 

```
cp /etc/pbs.conf /tmp/pbs_backup/pbs.conf.backup
```
5. If this is a scheduler or multisched host, make a copy of the scheduler's directory to modify:
 

```
cp -r $PBS_HOME/sched_priv /tmp/pbs_backup/sched_priv.work
```

or

```
cp -r $PBS_HOME/sched_priv_<multisched name> /tmp/pbs_backup/sched_priv_<multisched name>.work
```

## 6.5.11 Install the New Version of PBS

For an overlay upgrade, you install the new PBS in the same location as the existing PBS. The default location for PBS\_HOME is /var/spool/pbs, and the default for PBS\_EXEC is /opt/pbs.

### 6.5.11.1 Install New PBS Server(s)

Install the new version of PBS without uninstalling the previous version. If you are using failover, do not upgrade the primary and secondary servers simultaneously. Upgrade the primary first, then once that is complete, upgrade the secondary.

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, PBS\_LICENSE\_INFO, PBS\_SERVER and PBS\_DATA\_SERVICE\_USER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#).

If you are using failover, pay special attention to your configuration parameters, including PBS\_HOME and PBS\_MOM\_HOME, when installing the server sub-package on the secondary server host. See "[Configuring the pbs.conf File for Failover](#)" on page 411 in the *PBS Professional Administrator's Guide*.

4. Install the server sub-package. The method you use depends on the version you are upgrading from.
  - When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/server sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
  - When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/server sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.5.11.2 Install New PBS MoMs

Install the new version of PBS on all execution hosts without uninstalling the previous version:

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
4. Install the execution sub-package. The method you use depends on the version you are upgrading from.
  - When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/execution sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
  - When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/execution sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.5.11.3 Install New PBS Client Commands

Install the new version of PBS on all hosts without uninstalling the previous version:

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#).
4. Install the client command sub-package. The method you use depends on the version you are upgrading from.
  - When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/client command sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
  - When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/client command sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 6.5.11.4 Install New PBS Communication Daemons

If you are installing a communication daemon on a communication-only host, install the server-scheduler-communication-MoM sub-package, and disable the server, scheduler, and MoM on that host. (MoM is disabled by default.) Install the new version of PBS without uninstalling the previous version.

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
4. Disable the server, scheduler, and MoM. In `pbs.conf`:
 

```
PBS_START_SERVER=0
PBS_START_SCHED=0
PBS_START_MOM=0
```
5. Install the server sub-package. The method you use depends on the version you are upgrading from.
  - When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/server sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
  - When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/server sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

## 6.5.12 Prepare Configuration File for New Scheduler(s)

If you were running one or more multischeds with your old version of PBS, make sure you update their configuration files along with that of the default scheduler. Note that the `preempt_order`, `preempt_prio`, `preempt_queue_prio`, `preempt_sort`, and `log_events` scheduler attributes are new; some were parameters in `sched_config` with the same names. In a later step (after the server is running), you will use `qmgr` to set the attributes. We explicitly list the step; don't worry.

For each scheduler:

1. Make a copy of the new `pbs_sched_config`, which is in `PBS_EXEC/etc/pbs_sched_config`.  

```
cp $PBS_EXEC/etc/pbs_sched_config $PBS_EXEC/etc/pbs_sched_config.new
```
2. Update `PBS_EXEC/etc/pbs_sched_config.new` with any modifications that were made to the old `%PBS_HOME/sched_priv/sched_config` or `%PBS_HOME/sched_priv_<multisched name>/sched_config`. This is saved in the backup directory `/tmp/pbs_backup/sched_priv.work`.
3. If you were using `vmem` at the queue or server level before the upgrade, then after upgrading you must add `vmem` to the `resource_unset_infinite sched_config` option. Otherwise jobs requesting `vmem` will not run.
4. Move `PBS_EXEC/etc/pbs_sched_config.new` to the correct name and location, i.e. `$PBS_HOME/sched_priv/sched_config` or `$PBS_HOME/sched_priv_<multisched name>/sched_config`:  

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv/sched_config
```

or

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv_<multisched name>/sched_config
```

### 6.5.13 Update Holidays File

Make sure your new holidays file is up to date.

### 6.5.14 Modify the New PBS Configuration File

Your new `pbs.conf` needs to reflect any changes that you made to the old file.

If you will use failover:

- Edit `pbs.conf` on the primary server host to include failover settings. See ["Configuring Failover For the Primary Server on Linux" on page 413 in the PBS Professional Administrator's Guide](#). Make any other changes to this file that you made to the old `pbs.conf`.
- Edit `pbs.conf` on the secondary server host to include failover settings. See ["Configuring Failover For the Secondary Server on Linux" on page 415 in the PBS Professional Administrator's Guide](#). Make any other changes to this file that you made to the old `pbs.conf`. You can use the following steps:
  - Copy `pbs.conf` from primary to secondary
  - Modify `pbs.conf` on secondary for failover (`PBS_START_SCHED = 0`)
- Edit `pbs.conf` on all execution and client hosts to include failover settings. See ["Configuring Failover For Execution and Client Hosts on Linux" on page 416 in the PBS Professional Administrator's Guide](#). Make any other changes to this file that you made to the old `pbs.conf`.

If you will not use failover, edit `pbs.conf` on each host to include changes that you made to the old `pbs.conf`.

### 6.5.15 Configure Communication Daemons

If you are using additional communication daemons (more than those automatically installed on server hosts), configure them. See [section 4.5.3.2, "Configuring Communication Daemons", on page 50](#).

---

## 6.5.16 Start Then Stop New PBS Servers (If Using Failover)

### 6.5.16.1 Start New Servers

If you are not using failover, skip this step. If you are using failover, this pair of start and stop steps really is necessary. Bear with us.

1. If you will run a MoM on each server host, disable MoM start in `pbs.conf`, so that it contains this:

```
PBS_START_MOM=0
```

2. Start PBS on the primary server host:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

3. Once the primary is finished starting, start PBS on the secondary server host:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

### 6.5.16.2 Stop the Servers

If you are not using failover, skip this step.

1. On the primary server host:

- a. Stop PBS:

```
systemctl stop pbs
```

or

```
<path to init.d>/init.d/pbs stop
```

- b. If a MoM is installed, enable it by setting `PBS_START_MOM=1` in `pbs.conf`

2. On the secondary server host:

- a. Stop PBS:

```
systemctl stop pbs
```

or

```
<path to init.d>/init.d/pbs stop
```

- b. If a MoM is installed, enable it by setting `PBS_START_MOM=1` in `pbs.conf`

## 6.5.17 Start New PBS MoMs, Schedulers, Servers, and Comms

### 6.5.17.1 Start PBS on Execution Hosts

On each execution host, first update `PBS_HOME` by running the start/stop script or `systemctl start`, then start the MoMs:

1. Prevent the script from starting MoMs by setting `PBS_START_MOM=0` in `pbs.conf`

2. Start PBS:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

3. Stop PBS:

```
systemctl stop pbs
```

or

```
<path to init.d>/init.d/pbs stop
```

4. Enable starting MoMs by setting `PBS_START_MOM=1` in `pbs.conf`

5. Start MoM:

```
$PBS_EXEC/sbin/pbs_mom
```

### 6.5.17.2 Start PBS on Server Hosts

If failover is configured, start PBS on the primary server host before the secondary.

1. Prevent the script from starting MoMs by setting `PBS_START_MOM=0` in `pbs.conf`

2. Start PBS on the primary server host:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

3. Once the primary is finished starting, start PBS on the secondary server host:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

4. If a MoM will run on the server host(s):

- a. Enable starting MoMs by setting `PBS_START_MOM=1` in `pbs.conf`

- b. Start MoM:

```
$PBS_EXEC/sbin/pbs_mom
```

### 6.5.17.3 Restart Multischeds

To start a multisched, call `pbs_sched` and specify the name and port you already gave it. For each multisched:

```
pbs_sched -I <name of multisched> -S <same value as sched_port for this multisched>
```



### 6.5.17.4 Start PBS on Communication-only Hosts

Start PBS on any communication-only hosts. On each communication-only host, type:

```
systemctl start pbs
or
<path to init.d>/init.d/pbs start
```

## 6.5.18 Import and Configure Hooks

Make sure you do not overwrite the new `pbs_cgroups` hook or its configuration file by importing the old ones. Instead, use the saved information from your old hook to modify the new hook and configuration file.

### 6.5.18.1 Import Old Hooks Except for Cgroups Hook

1. Do not import your old `pbs_cgroups` hook. Import your other hooks and their configuration files. For each hook **except** for `pbs_cgroups`:

```
# qmgr -c 'import hook <hook name> application/x-python default /tmp/<hook name>.new3.6'
# qmgr -c 'import hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

### 6.5.18.2 Modify Cgroups Hook Configuration File

If you will use the `cgroups` hook:

1. Export the `cgroups` hook configuration file to `pbs_cgroups.json`:
 

```
# qmgr -c 'export hook pbs_cgroups application/x-config default' > pbs_cgroups.json
```
2. If the `cgroups` memory subsystem is not mounted on the system, disable 'memory' in the `cgroups` hook configuration file:

- a. Check to see whether it is mounted:

```
# mount | grep cgroup | grep memory
```

If the `memory` subsystem is mounted, the command returns something like "cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)".

- b. If this returns empty, edit the `pbs_cgroups.json` file so that 'enabled' parameter for 'memory' under `cgroup` is *false*:

```
"cgroup": {
  ...
  "memory": {
    "enabled": false,
```

3. If you made changes to the old `cgroups` configuration file, you may want to make those changes in the new configuration file. Use the information saved in `/etc/pbs_cgroups.old2.7`
4. Import the modified configuration (make sure you use "x-config"):

```
# qmgr -c 'import hook pbs_cgroups application/x-config default pbs_cgroups.json'
```

### 6.5.18.3 Enable Cgroups Hook

If you will use the `cgroups` hook, enable the `pbs_cgroups` hook:

```
qmgr -c "set hook pbs_cgroups enabled=true"
```

### 6.5.18.4 Write and Deploy New Hooks

If you have written new hooks for the new version of PBS, deploy them now. See the *PBS Professional Hooks Guide*.

### 6.5.18.5 Restart MoMs

On each execution host, restart MoM :

```
ps -eaf | grep pbs_mom
kill <MoM PID>
/opt/pbs/sbin/pbs_mom
```

## 6.5.19 Set License Location Server Attribute

Set the `pbs_license_info` server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

## 6.5.20 Configure Sharing and Placement Sets

### 6.5.20.1 Configuration with Cgroups Hook

As of version 2020.1, the `cgroups` hook creates the child vnodes on a multi-vnode machine if you set `vnnode_per_numa_node` to `true`; in this case, it is important that any Version 2 configuration files refer only to these vnodes. Use Version 2 configuration files only to set the `sharing` attribute and optionally to set resources that will be used for placement sets. The default value for the `sharing` attribute of the vnodes is `"sharing=default_shared"`. You can change this, for example to `"sharing=default_excl"`.

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named `"vnodedefs"` that has MoM's list of vnodes; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#)

```
# pbsnodes -av | awk -F'|' '{printf "%s:\tsharing = default_excl\n", $2}' > vnodedefs
```

2. Edit the file to reflect what you want for the `sharing` attribute and placement sets. Use the information saved in `/tmp/pbs_mom_backup/mom_configs/` in step ["Save Execution Host Configuration Information" on page 72](#)

3. Create your new Version 2 configuration file and name it for example `"vnodedefs"`:

```
# pbs_mom -s insert vnodedefs vnodedefs
```

4. Restart `pbs_mom`:

```
# ps -eaf | grep pbs_mom
# kill <MoM PID>
# /opt/pbs/sbin/pbs_mom
```

### 6.5.20.2 Configuration without Cgroups Hook

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named "vnodedefs"; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#)
2. Create your new Version 2 configuration file and name it for example "vnodedefs":

```
# pbs_mom -s insert vnodedefs vnodedefs
```

3. Restart pbs\_mom:

```
# pbs -eaf | grep pbs_mom
# kill <MoM PID>
# /opt/pbs/sbin/pbs_mom
```

## 6.5.21 Set New Scheduler Attributes

For the default scheduler and all multischeds:

- The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).
- The scheduler's `log_filter` configuration parameter is **obsolete**. The scheduler's log filter now uses the same bitmask system as the other daemons. The new default value is `767`. Use `qmgr` to set the scheduler's `log_events` attribute to the value you want. See ["Specifying Scheduler Log Events" on page 550 in the PBS Professional Administrator's Guide](#).

## 6.5.22 Re-wrap Any MPIs

If you want any wrapped MPIs, wrap them. See ["Integration by Wrapping" on page 457 in the PBS Professional Administrator's Guide](#).

## 6.5.23 Enable STONITH Script

If your secondary server has a STONITH script, allow the STONITH script to run by setting its permissions to `0755`.

## 6.5.24 Enable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, enable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

## 6.5.25 Enable Scheduling

If you disabled scheduling earlier, enable it for the default scheduler and any multischeds:

```
qmgr -c 'set sched <scheduler name> scheduling = true'
```

## 6.5.26 Shut Down and Restart Servers

1. Shut down both servers:  
`qterm -f`
2. Restart PBS on the server hosts. On each server host, primary first:  
`systemctl start pbs`  
or  
`<path to init.d>/init.d/pbs start`

## 6.5.27 Removing Old PBS

If you decide to remove the old version of PBS after upgrading, **be sure to use the `--noscripts` option** when using `rpm -e`. Using `rpm -e` without this option, even on an older package than the one you are currently using, will cause any currently running PBS daemons to shut down, and will also remove the system V init and/or systemd service startup files. This will prevent PBS daemons from starting automatically at system boot time. If you wish to remove an older RPM without these effects, use `rpm -e --noscripts`.

# 6.6 Overlay Upgrade on One or More Machines Running Cpuset MoM

Machines running the cpuset MoM typically included HPE MC990X, HPE Superdome Flex, or HPE 8600, for versions of PBS before 2020.1.

As of 2020.1, we no longer provide `pbs_mom.cpuset`; instead, we use standard `pbs_mom`, and the `cgroups` hook manages the cpusets for jobs. We include the instructions on making the change from the cpuset MoM to the `cgroups` hook below.

You must run the following commands as root.

## 6.6.1 Prevent Jobs From Being Started

Prevent the scheduler(s) from starting jobs. Set `scheduling` to `false` for the default scheduler and each multisched:

```
qmgr -c 'set sched <scheduler name> scheduling = false'
```

## 6.6.2 Allow Running Jobs to Finish, or Requeue Them

You cannot perform an upgrade while jobs are running. Either let running jobs finish, or requeue them. (You can also delete them.)

To requeue any running jobs:

1. List the jobs. This will list some jobs more than once. You only need to requeue each job once:  
`pbsnodes <hostname> | grep jobs`
2. Requeue the jobs:  
`qrerun <job ID> <job ID> ...`

To kill the jobs:

1. List the jobs. This will list some jobs more than once. You only need to kill each job once:

```
pbsnodes <hostname> | grep jobs
```

2. Use the `qdel` command to kill each job by job ID:

```
qdel <job ID> <job ID> ...
```

To drain the host, wait until any running jobs have finished.

Make sure that there are no old job files on any execution hosts. Remove any of the following:

```
$PBS_HOME/mom_priv/jobs/*.JB
```

### 6.6.3 Disable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, disable cloud bursting. See the *Altair Control Administrator's Guide*.

### 6.6.4 Disable STONITH Script

If your secondary server has a STONITH script, prevent the STONITH script from running by setting its permissions to 0644.

### 6.6.5 Unwrap Any Wrapped MPIs

If you used the `pbsrun_wrap` mechanism with your old version of PBS, you must first unwrap any MPIs that you wrapped. This includes MPICH-GM, MPICH-MX, MPICH2, etc. You can re-wrap your MPIs after upgrading PBS.

For example, you can unwrap an MPICH2 MPI:

```
# pbsrun_unwrap pbsrun.mpich2_64
```

See "[pbsrun\\_unwrap](#)" on page 51 of the *PBS Professional Reference Guide*.

### 6.6.6 Save Execution Host Configuration Information

On each PBS execution host, copy the Version 1 and Version 2 configuration files:

1. Make a backup directory:

```
mkdir /tmp/pbs_mom_backup
```

2. Make a copy of the Version 1 configuration file:

```
cp $PBS_HOME/mom_priv/config /tmp/pbs_mom_backup/config.backup
```

3. Make a copy of the Version 2 configuration files:

```
mkdir /tmp/pbs_mom_backup/mom_configs
```

```
pbs_mom -s list | egrep -v '^PBS' | while read file
do
```

```
    pbs_mom -s show file > /tmp/pbs_mom_backup/mom_configs/$file
done
```

## 6.6.7 Save Hooks and Hook Configuration Files

Save your hooks and hook configuration files in ASCII format so you can check them and import them later. The new version of PBS includes a new `pbs_cgroups` hook with a new configuration file. You must use the new hook and configuration file, but you may want to modify the configuration file, so if you have made any changes to your existing `pbs_cgroups` hook configuration file, you need to save it before you upgrade. Later, you can use the saved information to modify the new configuration file.

For **each** hook:

1. Save the hook. Export the hook:  

```
# qmgr -c 'export hook <hook name> application/x-python default /tmp/<hook name>.old2.7'
```
2. Save your hook configuration file. Export the configuration file:  

```
# qmgr -c 'export hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

## 6.6.8 Update Hooks and Hook Configuration Files for New Python

PBS 19.4.1 and later uses Python 3.6, so if you have not already, update all of your site-defined hooks (not the built-in hooks) to Python 3.6. For **each** hook except for the `pbs_cgroups` hook:

1. Update your hook to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your updated hook file differently; use something like `"/tmp/<hook name>.new3.6"`
2. Check that the contents of the configuration file are correct for Python 3.6

## 6.6.9 Remove Old PBS Configuration and Resource Conflicts

1. Ensure that each cpuset MoM host has its values for `resources_available.(mem|vmem|ncpus)` unset:  

```
Qmgr: unset node <hostname> resources_available.mem
Qmgr: unset node <hostname> resources_available.ncpus
Qmgr: unset node <hostname> resources_available.vmem
```
2. Remove the old PBS reserved files. On each execution host:  

```
# rm /var/spool/pbs/mom_priv/config.d/PBSvnodedefs
```
3. Delete the old default vnodes. On the server host:  

```
# qmgr -c "delete node @default"
```

## 6.6.10 Shut Down Your Existing PBS

1. Shut down the server(s), default scheduler, and MoMs:  

```
qterm -t immediate -m -s -f
```

If your server is not running in a failover environment, the "-f" option is not required.

2. Shut down any multischeds. On each multisched host:
  - a. Find the PID you want:
 

```
ps -ef | grep pbs_sched
```

For the default scheduler, you'll see "pbs\_sched", but for multischeds, you'll see "pbs\_sched -I <multisched name>".
  - b. Stop the scheduler or multisched:
 

```
kill <multisched PID>
```
3. On the server host and any other comm hosts, shut down the communication daemon:
 

```
systemctl stop pbs
```

or

```
<path to script>/pbs stop
```
4. Verify that PBS daemons are not running in the background:
 

```
ps -ef | grep pbs
```

If you see the pbs\_server, pbs\_sched, pbs\_mom, or pbs\_comm process running, manually terminate that process. If using failover, check both primary and secondary server hosts:

```
kill -9 <daemon PID>
```

## 6.6.11 Back Up Existing PBS Files

On each PBS host, make a tar file of the PBS\_HOME and PBS\_EXEC directories. On the MC990X, make sure you copy your backups to the server host, because otherwise they will be lost during the upgrade.

1. Make a backup directory:
 

```
mkdir /tmp/pbs_backup
```
2. Make a tar file of PBS\_HOME:
 

```
cd $PBS_HOME/..
tar -cvf /tmp/pbs_backup/PBS_HOME_tarbackup.tar $PBS_HOME
```
3. Make a tar file of PBS\_EXEC:
 

```
cd $PBS_EXEC/..
tar -cvf /tmp/pbs_backup/PBS_EXEC_tarbackup.tar $PBS_EXEC
```
4. Make a copy of your configuration file:
 

```
cp /etc/pbs.conf /tmp/pbs_backup/pbs.conf.backup
```
5. If this is a scheduler host, make a copy of the scheduler's directory to modify:
 

```
cp -r $PBS_HOME/sched_priv /tmp/pbs_backup/sched_priv.work
```

or

```
cp -r $PBS_HOME/sched_priv_<multisched name> /tmp/pbs_backup/sched_priv_<multisched name>.work
```

## 6.6.12 Install the New Version of PBS

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, PBS\_LICENSE\_INFO, PBS\_SERVER and PBS\_DATA\_SERVICE\_USER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#).

If you are using failover, pay special attention to your configuration parameters, including PBS\_HOME and PBS\_MOM\_HOME, when installing the server sub-package on the secondary server host. See ["Configuring the pbs.conf File for Failover" on page 411 in the PBS Professional Administrator's Guide](#).

4. Install the server sub-package:
  - When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/server sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
  - When upgrading from 14.2 or a later version:
 

```
rpm -Uhv <path/to/server sub-package>pbspro-<daemon>-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 6.6.12.1 Installing MoM on non-HPE 8600

On execution-only hosts, install the MoM sub-package:

- When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/MoM sub-package>pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
- When upgrading from 14.2 or a later version:
 

```
rpm -Uhv <path/to/MoM sub-package>pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 6.6.12.2 Installing MoM on HPE 8600

You install and configure MoM once on the root file system, then you push the image to all of the compute nodes by propagating it to the rack leaders. Then you reboot each node with the new image.

1. Log on to the Admin node as root.
2. Determine which image file is being used on the compute nodes. To list the nodes on rack 1:

```
cimage --list-nodes r1
```

It will show output in the form "*node: image\_name kernel*" similar to

```
r1i0n0: compute-sles15sp1 2.6.26.46-0.12-smp
```

Thus node r1i0n0 is running the image "compute-sles15sp1" and the kernel version "2.6.26.46-0.12-smp". For the remaining steps, it is assumed that those are the images and kernel available.

3. List the available images:

```
cimage --list-images
```



which will list the images available for the compute nodes. Each image may have multiple kernels.

- Unless you are experienced in managing the image files, we suggest that you create a copy of the image in use and install PBS in that copy. To copy an image:

```
cinstallman --create-image --clone --source compute-sles15sp1 --image compute-sles15sp1pbs
```

- The image file lives in the directory `/opt/clmgr/image/images`, so change into the `tmp` directory found in the new image just cloned:

```
cd /opt/clmgr/image/images/compute-sles15sp1pbs/tmp
```

- Chroot to the new image file:

```
chroot /opt/clmgr/image/images/compute-sles15sp1pbs /bin/sh
```

The new root is in effect.

- Download, unzip and untar the PBS package
- Make sure that parameters for `PBS_HOME`, `PBS_EXEC` and `PBS_SERVER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#)
- Install the PBS execution sub-package in the normal execution directory, `/opt/pbs`, in this system image:

```
rpm -U <path/to/sub-package>pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hard-ware>.rpm
```

- Do not start PBS
- Exit from the chroot shell and return to root's normal home directory.
- Power down each rack of compute nodes:

```
for n in `cnodes --ice-compute` ; do
    cpower node off $n
done
```

- Publish the new system image to the compute nodes:

```
cimage --push-rack compute-sles15sp1pbs r\*
```

This instruction will take several minutes to finish.

- Set the new image and kernel to be booted. This need not be done if: (1) rather than cloning a new image, you have installed PBS into the image already running on the compute nodes; or (2) you are using an image that was already pushed to the nodes.

```
cimage --set compute-sles15sp1pbs 2.6.26.46-0.12-smp r\*i\*n\*
```

- Power up the compute nodes:

```
for n in `cnodes --ice-compute` ; do
    cpower node on $n
done
```

It will take several minutes for the compute nodes to reboot.

### 6.6.13 Prepare Configuration File for New Scheduler(s)

If you were running one or more multischeds with your old version of PBS, make sure you update their configuration files along with that of the default scheduler. Note that the `preempt_order`, `preempt_prio`, `preempt_queue_prio`, `preempt_sort`, and `log_events` scheduler attributes are new; some were parameters in `sched_config` with the same names. In a later step (after the server is running), you will use `qmgr` to set the attributes. We explicitly list the step; don't worry.

For each scheduler:

1. Make a copy of the new `sched_config`, which is in `PBS_EXEC/etc/pbs_sched_config`.  

```
cp $PBS_EXEC/etc/pbs_sched_config $PBS_EXEC/etc/pbs_sched_config.new
```
2. Update `PBS_EXEC/etc/pbs_sched_config.new` with any modifications that were made to the old `%PBS_HOME/sched_priv/sched_config` or `%PBS_HOME/sched_priv_<multisched name>/sched_config`. This is saved in the backup directory `/tmp/pbs_backup/sched_priv.work`.
3. If you were using `vmem` at the queue or server level before the upgrade, then after upgrading you must add `vmem` to the `resource_unset_infinite sched_config` option. Otherwise jobs requesting `vmem` will not run.
4. Move `PBS_EXEC/etc/pbs_sched_config.new` to the correct name and location, i.e. `$PBS_HOME/sched_priv/sched_config` or `$PBS_HOME/sched_priv_<multisched name>/sched_config`:  

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv/sched_config
```

or

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv_<multisched name>/sched_config
```

### 6.6.14 Update Holidays File

Make sure your new holidays file is up to date.

### 6.6.15 Modify the New PBS Configuration File

Your new `pbs.conf` needs to reflect any changes that you made to the old file.

If you will use failover:

- Edit `pbs.conf` on the primary server host to include failover settings. See ["Configuring Failover For the Primary Server on Linux" on page 413 in the PBS Professional Administrator's Guide](#). Make any other changes to this file that you made to the old `pbs.conf`.
- Edit `pbs.conf` on the secondary server host to include failover settings. See ["Configuring Failover For the Secondary Server on Linux" on page 415 in the PBS Professional Administrator's Guide](#). Make any other changes to this file that you made to the old `pbs.conf`. You can use the following steps:
  - Copy `pbs.conf` from primary to secondary
  - Modify `pbs.conf` on secondary for failover (`PBS_START_SCHED = 0`)
- Edit `pbs.conf` on all execution and client hosts to include failover settings. See ["Configuring Failover For Execution and Client Hosts on Linux" on page 416 in the PBS Professional Administrator's Guide](#). Make any other changes to this file that you made to the old `pbs.conf`.

If you will not use failover, edit `pbs.conf` on each host to include changes that you made to the old `pbs.conf`.

---

## 6.6.16 Configure Communication Daemons

If you are using additional communication daemons (more than those automatically installed on server hosts), configure them. See [section 4.5.3.2, "Configuring Communication Daemons", on page 50](#).

## 6.6.17 Start Then Stop New PBS Servers (If Using Failover)

### 6.6.17.1 Start New Servers

If you are not using failover, skip this step. If you are using failover, this pair of start and stop steps really is necessary. Bear with us.

Start PBS on the server host. The start/stop script is located here:

```
If /etc/init.d exists
```

```
    /etc/init.d/pbs
```

```
Else
```

```
    /etc/rc.d/init.d/pbs
```

1. If you will run a MoM on each server host, disable MoM start in `pbs.conf`, so that it contains this:

```
PBS_START_MOM=0
```

2. Start PBS on the primary server host and then the secondary server host:

```
systemctl start pbs
```

```
or
```

```
<path to init.d>/init.d/pbs start
```

### 6.6.17.2 Stop the Servers

If you are not using failover, skip this step.

1. On the primary server host:

- a. Stop PBS:

```
systemctl stop pbs
```

```
or
```

```
<path to init.d>/init.d/pbs stop
```

- b. If a MoM is to run, enable it by setting `PBS_START_MOM=1` in `pbs.conf`

2. On the secondary server host:

- a. Stop PBS:

```
systemctl stop pbs
```

```
or
```

```
<path to init.d>/init.d/pbs stop
```

- b. If a MoM is to run, enable it by setting `PBS_START_MOM=1` in `pbs.conf`

## 6.6.18 Start New PBS MoMs, Schedulers, Servers, and Comms

### 6.6.18.1 Start PBS on Execution Hosts

On each execution host, start MoM :

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

### 6.6.18.2 Start PBS on Server Hosts

If failover is configured, start the primary server host, wait until the primary is finished starting, then start the secondary:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

### 6.6.18.3 Restart Multischeds

To start a multisched, call `pbs_sched` and specify the name and port you already gave it. For each multisched:

```
pbs_sched -I <name of multisched> -S <same value as sched_port for this multisched>
```

### 6.6.18.4 Start PBS on Communication-only Hosts

Start PBS on any communication-only hosts. On each communication-only host, type:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

## 6.6.19 Import and Configure Hooks

Make sure you do not overwrite the new `pbs_cgroups` hook or its configuration file by importing the old ones. Instead, use the saved information from your old hook to modify the new hook and configuration file.

### 6.6.19.1 Import Old Hooks Except for Cgroups Hook

1. Do not import your old `pbs_cgroups` hook. Import your other hooks and their configuration files. For each hook **except** for `pbs_cgroups`:

```
# qmgr -c 'import hook <hook name> application/x-python default /tmp/<hook name>.new3.6'
```

```
# qmgr -c 'import hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

### 6.6.19.2 Modify Cgroups Hook Configuration File

1. Export the cgroups hook configuration file to `pbs_cgroups.json`:
 

```
# qmgr -c 'export hook pbs_cgroups application/x-config default' > pbs_cgroups.json
```
2. You can make the cgroups hook mimic the behavior of the cpuset MoM in previous versions:
  - a. Create one vnode for each NUMA node. Edit `pbs_cgroups.json` as follows (important):
 

```
"vnode_per_numa_node" : true,
```
  - b. Edit `pbs_cgroups.json` as follows (recommended):
 

```
"use_hyperthreads" : true,
```
  - c. Set the value of the `ncpus_are_cores` parameter; see ["Configuring Hyperthreading Support" on page 584 in the PBS Professional Administrator's Guide](#)
3. If the cgroups memory subsystem is not mounted on the system, disable 'memory' in the cgroups hook configuration file:
  - a. Check to see whether it is mounted:
 

```
# mount | grep cgroup | grep memory
```

 If the memory subsystem is mounted, the command returns something like `"cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)".`
  - b. If this returns empty, edit the `pbs_cgroups.json` file so that 'enabled' parameter for 'memory' under cgroup is `false`:
 

```
"cgroup": {
  ...
  "memory": {
    "enabled": false,
```
4. If you made changes to the old cgroups configuration file, you may want to make those changes in the new configuration file. Use the information saved in `/etc/pbs_cgroups.old2.7`
5. Import the modified configuration (make sure you use `"x-config"`):
 

```
# qmgr -c 'import hook pbs_cgroups application/x-config default pbs_cgroups.json'
```

### 6.6.19.3 Enable Cgroups Hook

6. Enable the `pbs_cgroups` hook:
 

```
qmgr -c "set hook pbs_cgroups enabled=true"
```

### 6.6.19.4 Write and Deploy New Hooks

If you have written new hooks for the new version of PBS, deploy them now. See the *PBS Professional Hooks Guide*.

### 6.6.19.5 Restart MoMs

On each execution host, restart MoM :

```
ps -eaf | grep pbs_mom
kill <MoM PID>
/opt/pbs/sbin/pbs_mom
```

## 6.6.20 Set License Location Server Attribute

Set the `pbs_license_info` server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

## 6.6.21 Configure Sharing and Placement Sets

As of version 2020.1, the `cgroups` hook creates the child `vnodes` on a multi-`vnode` machine; it is important that any Version 2 configuration files refer only to these `vnodes`. Use Version 2 configuration files only to set the `sharing` attribute and optionally to set resources that will be used for placement sets. The default value for the `sharing` attribute of the `vnodes` is `"sharing=default_shared"`. You can change this, for example to `"sharing=default_excl"`.

Make sure that a Version 2 configuration file matches your available `vnodes` every time MoM is started. If your machine reboots with a hardware change, your earlier placement set information will not make sense because child `vnode` names will not match the available hardware. You can use a script to regenerate this file each time the machine starts, and run the script before MoM is restarted.

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named `"vnodedefs"` that has MoM's list of `vnodes`; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#):

```
# pbsnodes -av | awk -F'|' '{printf "%s:\tsharing = default_excl\n", $2}' > vnodedefs
```

2. Edit the file to reflect what you want for the `sharing` attribute and placement sets. Use the information saved in `/tmp/pbs_mom_backup/mom_configs/` in step ["Save Execution Host Configuration Information" on page 83](#)
3. Create your new Version 2 configuration file and name it for example `"vnodedefs"`:

```
# pbs_mom -s insert vnodedefs vnodedefs
```

4. Restart `pbs_mom`:

```
# pbs -eaf | grep pbs_mom
# kill <MoM PID>
# /opt/pbs/sbin/pbs_mom
```

## 6.6.22 Re-Wrap Any MPIs

If you want any wrapped MPIs, wrap them. See ["Integration by Wrapping" on page 457 in the PBS Professional Administrator's Guide](#).

## 6.6.23 Shut Down and Restart Servers

1. Shut down both servers:

```
qterm -f
```

2. Restart PBS on the server hosts. On each server host, primary first:

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```

## 6.6.24 Set New Scheduler Attributes

The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See "[Scheduler Attributes](#)" on [page 300 of the PBS Professional Reference Guide](#).

The scheduler's `log_filter` configuration parameter is **obsolete**. The scheduler's log filter now uses the same bitmask system as the other daemons. The new default value is `767`. Use `qmgr` to set the scheduler's `log_events` attribute to the value you want. See "[Specifying Scheduler Log Events](#)" on [page 550 in the PBS Professional Administrator's Guide](#).

## 6.6.25 Enable STONITH Script

If your secondary server has a STONITH script, allow the STONITH script to run by setting its permissions to `0755`.

## 6.6.26 Enable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, enable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

## 6.6.27 Enable Scheduling

If you disabled scheduling earlier, enable it for the default scheduler and any multischeds:

```
qmgr -c 'set sched <scheduler name> scheduling = true'
```

## 6.6.28 Removing Old PBS

If you decide to remove the old version of PBS after upgrading, **be sure to use the `--noscripts` option** when using `rpm -e`. Using `rpm -e` without this option, even on an older package than the one you are currently using, will cause any currently running PBS daemons to shut down, and will also remove the system `V init` and/or `systemd` service startup files. This will prevent PBS daemons from starting automatically at system boot time. If you wish to remove an older RPM without these effects, use `rpm -e --noscripts`.

# 6.7 Migration Upgrade Under Linux

Use these instructions:

- When moving between hosts
- When upgrading from an open-source version of PBS Professional
- When certain European or Japanese characters are stored in the data store

For specific upgrade recommendations and updates, see the Release Notes.

For a migration upgrade, you kill or requeue all jobs, install the new PBS with `PBS_EXEC` and `PBS_HOME` in different locations from those of the old version of PBS, run both the old and new instances of PBS at the same time, and `qmove` the jobs from the old server to the new one.

During a migration upgrade, jobs cannot be running. You can let any jobs finish before the upgrade. You can checkpoint, terminate and requeue all possible jobs and requeue non-checkpointable but rerunnable jobs. Your options with non-rerunnable jobs are to either let them finish or kill them.

In the instructions below, file and directory pathnames are the PBS defaults. If you installed PBS in different locations, use your locations instead. `PBS_EXEC_OLD` refers to your existing, pre-upgrade location for `PBS_EXEC`.

The following commands must be run as root.

## 6.7.1 Set Paths for Old PBS

To use the following commands without having to substitute actual paths, on the server host, source your `/etc/pbs.conf` file.

We recommend using `/opt` as the location where you'll run your old PBS during the job transfer phase, rather than `/tmp`.

- Choose where you want to copy your old `PBS_EXEC`; set `PBS_EXEC_OLD` to this location, and export it
- Choose where you want to copy your old `PBS_HOME`; set `PBS_HOME_OLD` to this location, and export it

## 6.7.2 Prevent Jobs From Being Enqueued or Started

You must deactivate the scheduler(s) and queues. When the `scheduling` attribute is `false`, jobs are not started by the scheduler. When the queues' `enabled` attribute is `false`, jobs cannot be enqueued.

1. Prevent the scheduler(s) from starting jobs. Set `scheduling` to `false` for the default scheduler and each multisched:
 

```
qmgr -c "set sched <scheduler name> scheduling = false"
```
2. Print a list of all queues managed by the server. Save the list of queue names for the next step:
 

```
qstat -q
```
3. Disable queues to stop jobs from being enqueued. Do this for each queue in your list from the previous step:
 

```
qdisable <queue name>
```

## 6.7.3 Allow Running Jobs to Finish, or Requeue Them

You cannot perform a migration upgrade while jobs are running. Either let running jobs finish, or requeue them. (You can also delete them.)

To requeue any running jobs:

1. List the jobs. This will list some jobs more than once. You only need to requeue each job once:
 

```
pbsnodes <hostname> | grep jobs
```
2. Requeue the jobs:
 

```
qrerun <job ID> <job ID> ...
```

To kill the jobs:

1. List the jobs. This will list some jobs more than once. You only need to kill each job once:
 

```
pbsnodes <hostname> | grep jobs
```
2. Use the `qdel` command to kill each job by job ID:
 

```
qdel <job ID> <job ID> ...
```

To drain the host, wait until any running jobs have finished.

Make sure that there are no old job files on any execution hosts. Remove any of the following:

```
$PBS_HOME/mom_priv/jobs/*.JB
```



## 6.7.4 Disable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, disable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

## 6.7.5 Disable STONITH Script

If your secondary server has a STONITH script, prevent the STONITH script from running by setting its permissions to 0644.

## 6.7.6 Unwrap Any Wrapped MPIs

If you used the `pbsrun_wrap` mechanism with your old version of PBS, you must first unwrap any MPIs that you wrapped. This includes MPICH-GM, MPICH-MX, MPICH2, etc. You can re-wrap your MPIs after upgrading PBS.

For example, you can unwrap an MPICH2 MPI:

```
# pbsrun_unwrap pbsrun.mpich2_64
```

See "[pbsrun\\_unwrap](#)" on page 51 of the [PBS Professional Reference Guide](#).

## 6.7.7 Save Server Host Information To Be Used for New PBS

At the server:

1. Create a backup directory called `/tmp/pbs_backup`  

```
mkdir /tmp/pbs_backup
```
2. Make a copy of the server's configuration for the new PBS:  

```
qmgr -c "print server" > /tmp/pbs_backup/server.new
```
3. Make a copy of the vnode attributes for the new PBS:  

```
qmgr -c "print node @default" > /tmp/pbs_backup/nodes.new
```
4. Make a copy of all scheduler attributes for the new PBS (this prints all settable attributes for the default and multischeds):  

```
qmgr -c "print sched" > /tmp/pbs_backup/sched_attrs.new
```
5. Print reservation information to a file:  

```
pbs_rstat -f > /tmp/pbs_backup/reservations
```
6. Make a copy of `pbs.conf` for the new PBS. This command is all one line:  

```
cp /etc/pbs.conf /tmp/pbs_backup/pbs.conf.backup
```
7. Make a copy of each scheduler's directory for the new PBS. For the default scheduler and each multisched:  

```
cp -rp $PBS_HOME/sched_priv /tmp/pbs_backup/sched_priv.new
```

or

```
cp -rp $PBS_HOME/sched_priv_<multisched name> /tmp/pbs_backup/sched_priv_<multisched name>.new
```

## 6.7.8 Save Execution Host Configuration Files

On each PBS execution host, copy the Version 1 and Version 2 configuration files:

1. Make a backup directory:

```
mkdir /tmp/pbs_mom_backup
```

2. Make a copy of the Version 1 configuration file:

```
cp $PBS_HOME/mom_priv/config /tmp/pbs_mom_backup/config.backup
```

3. Make a copy of the Version 2 configuration files:

```
mkdir /tmp/pbs_mom_backup/mom_configs
$PBS_EXEC_OLD/sbin/pbs_mom -s list | egrep -v '^PBS' | while read file
do
    $PBS_EXEC_OLD/sbin/pbs_mom -s show file > /tmp/pbs_mom_backup/mom_configs/$file
done
```

## 6.7.9 Save Hooks and Hook Configuration Files

Save your hooks and hook configuration files in ASCII format so you can check them and import them later. The new version of PBS includes a new `pbs_cgroups` hook with a new configuration file. If you use the `cgroups` hook, you must use the new hook and configuration file, but you may want to modify the configuration file, so if you have made any changes to your existing `pbs_cgroups` hook configuration file, you need to save it before you upgrade. Later, you can use the saved information to modify the new configuration file.

For **each** hook:

1. Save the hook. Export the hook:

```
# qmgr -c 'export hook <hook name> application/x-python default /tmp/<hook name>.old2.7'
```

2. Save your hook configuration file. Export the configuration file:

```
# qmgr -c 'export hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

## 6.7.10 Update Hooks and Hook Configuration Files for New Python

PBS 19.4.1 and later uses Python 3.6, so if you have not already, update all of your site-defined hooks (not the built-in hooks) to Python 3.6. For **each** hook except for the `pbs_cgroups` hook:

1. Update your hook to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your updated hook file differently; use something like `"/tmp/<hook name>.new3.6"`
2. Check that the contents of the configuration file are correct for Python 3.6

## 6.7.11 Shut Down Your Existing PBS

Use the `-t immediate` option to `qterm` so that all possible running jobs will be requeued. If you are using failover, this will stop the secondary server as well:

1. Shut down the server, scheduler, and MoMs:

```
qterm -t immediate -m -s -f
```

If your server is not running in a failover environment, the "-f" option is not required.

2. Shut down any multischeds. On each multisched host:
  - a. Find the PID you want:
 

```
ps -ef | grep pbs_sched
```

For the default scheduler, you'll see "pbs\_sched", but for multischeds, you'll see "pbs\_sched -I <multisched name>".
  - b. Stop the scheduler or multisched:
 

```
kill <multisched PID>
```
3. On the server host and any other comm hosts, shut down the communication daemon:
 

```
systemctl stop pbs
```

or

```
<path to script>/pbs stop
```
4. Verify that PBS daemons are not running in the background:
 

```
ps -ef | grep pbs
```

If you see the pbs\_server, pbs\_sched, pbs\_mom, or pbs\_comm process running, manually terminate that process. If using failover, check both primary and secondary server hosts:

```
kill -9 <daemon PID>
```

## 6.7.12 Back Everything Up to Transfer Location

Later, you will run the old PBS server from the backup location while you are moving jobs to the new server. You must do a copy, not a move, because the installation software depends on the old version of PBS being available for it to remove. You'll be running commands from the backup directory, so we recommend a directory under /opt.

### 6.7.12.1 Back Up Server/scheduler/communication Host

On the server host, copy the existing PBS\_HOME and PBS\_EXEC hierarchies to the backup location.

1. Copy PBS\_HOME to the backup directory:
 

```
cp -rp $PBS_HOME $PBS_HOME_OLD
```
2. Copy PBS\_EXEC to the backup directory:
 

```
cp -rp $PBS_EXEC $PBS_EXEC_OLD
```

### 6.7.12.2 Back Up Execution Host Information

On each execution host, copy the existing PBS\_HOME and PBS\_EXEC hierarchies to the backup location. This is just for safekeeping.

1. Copy PBS\_HOME to the backup directory:
 

```
cp -rp $PBS_HOME /tmp/pbs_mom_backup/pbs_mom_home_backup
```
2. Copy PBS\_EXEC to the backup directory:
 

```
cp -rp PBS_EXEC /tmp/pbs_mom_backup/pbs_mom_exec_backup
```

## 6.7.13 Install the New Version of PBS

For a migration upgrade, use `rpm -i` so that the old version of PBS can still be used to move the jobs. You might think that you'd use `rpm -U`, but that removes the old PBS, and you still need it until the jobs are moved.

### 6.7.13.1 Install New PBS Server

On the server host, install the new version of PBS without uninstalling the previous version.

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for `PBS_HOME`, `PBS_EXEC`, `PBS_LICENSE_INFO`, `PBS_SERVER` and `PBS_DATA_SERVICE_USER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Make sure that `PBS_HOME` and `PBS_EXEC` are in locations that are different from your existing PBS.

If you are using failover, pay special attention to your configuration parameters, including `PBS_HOME` and `PBS_MOM_HOME`, when installing the server sub-package on the secondary server host. See [section 3.5.2.2, "Setting Installation Parameters", on page 25](#) and ["Configuring the pbs.conf File for Failover" on page 411 in the PBS Professional Administrator's Guide](#).

4. Install the server sub-package:

```
rpm -i --prefix=<new PBS_EXEC location> <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.7.13.2 Install New PBS MoMs

On each execution host, install the new version of PBS without uninstalling the previous version. You can install new MoMs in the same locations as the old MoMs.

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for `PBS_HOME`, `PBS_EXEC`, and `PBS_SERVER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#).
4. Install the execution sub-package. The method you use depends on the version you are upgrading from.

- When upgrading from 13.2 or an earlier version:

```
rpm -i <path/to/execution sub-package>/pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

- When upgrading from 14.2 or a later version:

```
rpm -U <path/to/execution sub-package>/pbspro-execution-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.7.13.3 Install New PBS Client Commands

On each client command host, install the new version of PBS without uninstalling the previous version:

1. Download the appropriate PBS package
2. Uncompress the package as an unprivileged user
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Make sure that PBS\_HOME and PBS\_EXEC point to the locations you're using for the new PBS.
4. Install the client command sub-package. The method you use depends on the version you are upgrading from.
  - When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/client command sub-package>/pbspro-client-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
  - When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/client command sub-package>/pbspro-client-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

### 6.7.13.4 Install New PBS Communication Daemons

If you are installing a communication daemon on a communication-only host, install the server-scheduler-communication-MoM sub-package, and disable the server, scheduler, and MoM on that host. (MoM is disabled by default.) Install the new version of PBS without uninstalling the previous version.

1. Download the appropriate PBS package
2. Uncompress the package
3. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Make sure that PBS\_HOME and PBS\_EXEC point to the locations you are using for the new PBS.
4. Disable the server, scheduler, and MoM. In pbs.conf:
 

```
PBS_START_SERVER=0
PBS_START_SCHED=0
PBS_START_MOM=0
```
5. Install the server sub-package. The method you use depends on the version you are upgrading from.
  - When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
  - When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

## 6.7.14 Switch To New PBS\_EXEC Path

Source your new /etc/pbs.conf file.

## 6.7.15 Create PBS\_HOME

Create the subdirectories under PBS\_HOME by running `pbs_habitat`. On the new PBS server host and on each execution host:

```
$PBS_EXEC/libexec/pbs_habitat
```

## 6.7.16 Start and Stop the New Server (If Using Failover)

If you are not using failover, skip this step. If you are using failover, this pair of start and stop steps really is necessary. Bear with us.

When the new server starts up it will have default queue "workq" and the server host already defined. You want to start the new server with empty configurations so that you can import your old settings.

1. Start the new server with empty queue and vnode configurations:

```
$PBS_EXEC/sbin/pbs_server -t create
```

A message will appear saying "Create mode and server database exists, do you wish to continue?"

Type "y" to continue.

Because of the new licensing scheme an additional message may appear:

```
"One or more PBS license keys are invalid, jobs may not run"
```

This message is expected. Continue to the next step in these instructions.

2. Shut down PBS:

```
qterm -t immediate -m -s -f
```

3. Verify that PBS daemons are not running in the background:

```
ps -ef | grep pbs
```

If you see the `pbs_server`, `pbs_sched`, `pbs_comm`, or `pbs_mom` process running, manually terminate that process. If using failover, check both primary and secondary server hosts:

```
kill -9 <daemon PID>
```

## 6.7.17 Start the New Server Without Defined Queues or Vnodes

When the new server starts up it will have default queue "workq" and the server host already defined. You want to start the new server with empty configurations so that you can import your old settings.

Start the new server with empty queue and vnode configurations:

```
$PBS_EXEC/sbin/pbs_server -t create
```

A message will appear saying "Create mode and server database exists, do you wish to continue?"

Type "y" to continue.

Because of the new licensing scheme an additional message may appear:

```
"One or more PBS license keys are invalid, jobs may not run"
```

This message is expected. Continue to the next step in these instructions.

## 6.7.18 Re-wrap Any MPIs

If you want any wrapped MPIs, wrap them. See ["Integration by Wrapping" on page 457 in the PBS Professional Administrator's Guide](#).

## 6.7.19 Set License Location Server Attribute

Set the `pbs_license_info` server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

## 6.7.20 Clean Up Configuration Information

### 6.7.20.1 Clean Up Scheduler Configuration Files

If you were running one or more multischeds with your old version of PBS, make sure you update their configuration files along with that of the default scheduler. Note that the `preempt_order`, `preempt_prio`, `preempt_queue_prio`, `preempt_sort`, and `log_events` scheduler attributes are new; some were parameters in `sched_config` with the same names. In a later step (after the server is running), you will use `qmgr` to set the attributes. For each scheduler:

1. Make a copy of the new `sched_config`, which is in `PBS_EXEC/etc/pbs_sched_config`.  

```
cp $PBS_EXEC/etc/pbs_sched_config $PBS_EXEC/etc/pbs_sched_config.new
```
2. Update `PBS_EXEC/etc/pbs_sched_config.new` with any modifications that were made to your old scheduler configuration file, saved in `%PBS_HOME/sched_priv/sched_config` or `%PBS_HOME/sched_priv_<multisched name>/sched_config`.
3. If you were using `vmem` at the queue or server level before the upgrade, then after upgrading you must add `vmem` to the `resource_unset_infinite` `sched_config` option. Otherwise jobs requesting `vmem` will not run.
4. Move `PBS_EXEC/etc/pbs_sched_config.new` to the correct name and location, i.e. `$PBS_HOME/sched_priv/sched_config` or `$PBS_HOME/sched_priv_<multisched name>/sched_config`:

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv/sched_config
```

or

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv_<multisched name>/sched_config
```

### 6.7.20.2 Clean Up Scheduler Attributes

For each scheduler, clean up the attributes saved in `/tmp/pbs_backup/<scheduler name>/sched_attrs.new`. When you read in multisched attributes, you'll re-create the multischeds, so make sure your new multischeds are what you want:

- Remove read-only attributes
- Remove lines containing the following:  

```
pbs_version
```

For the new default scheduler and all new multischeds:

- The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).
- The scheduler's `log_filter` configuration parameter is **obsolete**. The scheduler's log filter now uses the same bitmask system as the other daemons. The new default value is `767`. Use `qmgr` to set the scheduler's `log_events` attribute to the value you want. See ["Specifying Scheduler Log Events" on page 550 in the PBS Professional Administrator's Guide](#).

### 6.7.20.3 Clean Up Server Configuration

Remove read-only attributes from the server's configuration information in `server.new`. For example, remove lines containing the following:

```
license_count
pbs_version
```

Remove creation commands for any reservation queues. You will create reservations and their queues separately.

### 6.7.20.4 Copy User Credentials to New Server

PBS caches user credentials in `$PBS_HOME/server_priv/users`. PBS stores the credential for each user in a file named `<username>.CR`. Normally this directory is created by PBS when users log in. If you installed the new version of PBS in the same location as the old one, you do not need to copy user credentials.

However, if the new version of PBS is in a different location, you need to create the directory and copy the credential files, keeping the permissions the same:

1. Create the user credential directory:

```
mkdir -p $PBS_HOME/server_priv/users/
```

2. Copy the user credential files to the new directory:

```
cp -rpu $PBS_HOME_OLD/server_priv/users/* $PBS_HOME/server_priv/users/
```

### 6.7.20.5 Clean up Vnode Configuration

Here you prepare the vnode attribute input to the new `qmgr`.

If your system has multi-vnode hosts:

1. Copy your saved node configuration file `/tmp/pbs_backup/nodes.new` into two files:
  - `qmgr_parent_vnode.out`, which contains all the configuration information for parent vnodes
  - `qmgr_child_vnode.out`, which contains all the configuration information for vnodes that aren't parent vnodes
2. Continue by preparing configuration information for parent vnodes. You will prepare the configuration information for the other vnodes after they have been created, because the vnode names in your file must be precisely the same as the ones created by PBS.

If your system has only single-vnode hosts, follow the steps below for preparing configuration information for parent vnodes only.

#### 6.7.20.5.i Prepare Configuration Information for Parent Vnodes

Edit `qmgr_parent_vnode.out`:



Leave only the the following creation lines:

- Those for parent vnodes
- Any resources you want managed on the server side through `qmgr`
- Custom resources on the parent vnodes

Delete any lines for resources managed through Version 2 configuration files or that MoM reports from what the vnode's host OS is reporting. For example, delete:

- Child vnodes, that should be created by MoM (vnodes that are NOT parent vnodes)
- Lines that set the sharing attribute
- The `ncpus`, `mem`, and `vmem` resources, unless they should explicitly be set via `qmgr`

## 6.7.21 Create and Configure New Multischeds

Create the directories required for each new multisched, and configure each multisched. See ["Creating and Configuring a Multisched" on page 57 in the PBS Professional Administrator's Guide](#).

## 6.7.22 Start New Server and New Schedulers

1. Start the new server and new default scheduler. On the server host:

```
systemctl restart pbs
```

or

```
<path to init.d>/init.d/pbs restart
```

2. Start multischeds. To start a multisched, call `pbs_sched` and specify the name and port you already gave it. For each multisched:

```
pbs_sched -I <name of multisched> -S <same value as sched_port for this multisched>
```

## 6.7.23 Replicate Queue, Server, Scheduler, and Vnode Configurations

### 6.7.23.1 Replicate Server and Queue Attributes

1. Give the new server the old server's configuration, but modified for the new PBS:

```
$PBS_EXEC/bin/qmgr < /tmp/pbs_backup/server.new
```

2. Verify the configuration was read in properly:

```
$PBS_EXEC/bin/qmgr -c "print server"
```

### 6.7.23.2 Replicate Scheduler Attributes

1. Give the new default scheduler the old default scheduler's attributes, and re-create your multischeds:

```
$PBS_EXEC/bin/qmgr < /tmp/pbs_backup/<scheduler name>/sched_attrs.new
```

2. Verify the configurations were read in properly.

You can see all schedulers at once:

```
$PBS_EXEC/bin/qmgr -c "print sched"
```

Or for each scheduler:

```
$PBS_EXEC/bin/qmgr -c "print sched default"
or
$PBS_EXEC/bin/qmgr -c "print sched <multisched name>"
```

### 6.7.23.3 Replicate Vnode Attributes

Replicate vnode configuration, also modified for the new PBS:

1. Read in the parent vnode configuration file:

```
$PBS_EXEC/bin/qmgr < qmgr_parent_vnode.out
```

2. Wait until MoM or the cgroups hook creates any vnodes that are not parent vnodes. Check:

```
pbsnodes -av
```

3. Prepare configuration information for child vnodes:

Edit `qmgr_child_vnode.out`. Make sure that the vnode names in this file are exactly what MoM or the cgroups hook created. It's easiest to put all resource information into a Version 2 configuration file, rather than using `qmgr`.

Leave only the the following creation lines:

- Any resources you want managed on the server side through `qmgr`
- Custom resources on the other vnodes (but this may be easier in a Version 2 configuration file)

Delete any lines for resources managed through Version 2 configuration files or that MoM reports from what the vnode's host OS is reporting. For example, delete:

- Vnodes that should be created by the cgroups hook or MoM (vnodes that are NOT parent vnodes)
- Lines that set the `sharing` attribute
- The `ncpus`, `mem`, and `vmem` resources, unless they should explicitly be set via `qmgr`

4. Read in the configuration file for child vnodes (not parent vnodes):

```
$PBS_EXEC/bin/qmgr < qmgr_child_vnode.out
```

5. Verify the configurations were read in properly:

```
$PBS_EXEC/bin/pbsnodes -a
```

## 6.7.24 Import and Configure Hooks

Make sure you do not overwrite the new `pbs_cgroups` hook or its configuration file by importing the old ones. Instead, use the saved information from your old hook to modify the new hook and configuration file.

### 6.7.24.1 Import Old Hooks Except for Cgroups Hook

1. Do not import your old `pbs_cgroups` hook. Import your other hooks and their configuration files. For each hook **except** for `pbs_cgroups`:

```
# qmgr -c 'import hook <hook name> application/x-python default /tmp/<hook name>.new3.6'
# qmgr -c 'import hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

## 6.7.24.2 Modify Cgroups Hook Configuration File

If you will use the cgroups hook:

1. Export the cgroups hook configuration file to `pbs_cgroups.json`:
 

```
# qmgr -c 'export hook pbs_cgroups application/x-config default' > pbs_cgroups.json
```
2. If the cgroups memory subsystem is not mounted on the system, disable 'memory' in the cgroups hook configuration file:
  - a. Check to see whether it is mounted:
 

```
# mount | grep cgroup | grep memory
```

If the memory subsystem is mounted, the command returns something like "cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)".
  - b. If this returns empty, edit the `pbs_cgroups.json` file so that 'enabled' parameter for 'memory' under cgroup is *false*:
 

```
"cgroup": {
    ...
    "memory": {
      "enabled": false,
```
3. If you made changes to the old cgroups configuration file, you may want to make those changes in the new configuration file. Use the information saved in `/etc/pbs_cgroups.old2.7`
4. Import the modified configuration (make sure you use "x-config"):
 

```
# qmgr -c 'import hook pbs_cgroups application/x-config default pbs_cgroups.json'
```

## 6.7.24.3 Enable Cgroups Hook

If you will use the cgroups hook, enable the `pbs_cgroups` hook:

```
qmgr -c "set hook pbs_cgroups enabled=true"
```

## 6.7.24.4 Write and Deploy New Hooks

If you have written new hooks for the new version of PBS, deploy them now. See the *PBS Professional Hooks Guide*.

## 6.7.25 Start New MoMs

You can start the MoMs in any order.

- On each execution host:
 

```
systemctl start pbs
```

or

```
<path to init.d>/init.d/pbs start
```
- Optionally start a MoM on the new server host. If your old configuration had a MoM running on the server host, and you wish to replicate the configuration, you can start a MoM on that machine:
 

```
$PBS_EXEC/sbin/pbs_mom
```

## 6.7.26 Configure Sharing and Placement Sets

### 6.7.26.1 Configuration with Cgroups Hook

As of version 2020.1, the cgroups hook creates the child vnodes on a multi-vnode machine; if you will use the cgroups hook, it is important that any Version 2 configuration files refer only to these vnodes. Use Version 2 configuration files only to set the `sharing` attribute and optionally to set resources that will be used for placement sets. The default value for the `sharing` attribute of the vnodes is `"sharing=default_shared"`. You can change this, for example to `"sharing=default_excl"`.

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named `"vnodedefs"` that has MoM's list of vnodes; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#)

```
# pbsnodes -av | awk -F'=' '{printf "%s:\tsharing = default_excl\n", $2}' > vnodedefs
```

2. Edit the file to reflect what you want for the `sharing` attribute and placement sets. Use the information saved in `/tmp/pbs_mom_backup/mom_configs/` in step ["Save Execution Host Configuration Files" on page 96](#)

3. Create your new Version 2 configuration file and name it for example `"vnodedefs"`:

```
# pbs_mom -s insert vnodedefs vnodedefs
```

4. Restart `pbs_mom`:

```
# pbs -eaf | grep pbs_mom
# kill <MoM PID>
# /opt/pbs/sbin/pbs_mom
```

### 6.7.26.2 Configuration without Cgroups Hook

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named `"vnodedefs"`; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#)

2. Create your new Version 2 configuration file and name it for example `"vnodedefs"`:

```
# pbs_mom -s insert vnodedefs vnodedefs
```

3. Restart `pbs_mom`:

```
# pbs -eaf | grep pbs_mom
# kill <MoM PID>
# /opt/pbs/sbin/pbs_mom
```

## 6.7.27 Start New Communication Daemons

Start PBS on any communication-only hosts. On each communication-only host, type:

```
systemctl start pbs
or
<path to init.d>/init.d/pbs start
```

## 6.7.28 Verify Communication Between Server and MoMs

All new MoMs on all execution hosts should be running and communicating with the new server. Run `pbsnodes -a` on the new server host to see if it can communicate with the execution hosts in your complex. If a host is down, go to the problem host and restart the MoM:

```
# pbs -eaf | grep pbs_mom
# kill <MoM PID>
# /opt/pbs/sbin/pbs_mom
```

## 6.7.29 Re-create Reservations

You must re-create each reservation that was on the old server, using the `pbs_rsub` command. Each reservation is created as a new reservation. You can use all of the information about the old reservation except for its start time. Be sure to give each reservation a start time in the future. Use the information stored in `/tmp/pbs_backup/reservations`.

## 6.7.30 Change Ports and PBS\_EXEC Path in `pbs.conf` for Old PBS

You must edit the `pbs.conf` file of the old PBS so that all old services use ports that won't clash with those of the new PBS. Edit `/tmp/pbs_backup/pbs.conf.backup`.

You must change the port numbers for these PBS daemons: server and data service. You do not need to change the port number for the comm, MoM, or scheduler.

You must also make sure that the `PBS_EXEC` entry in the old `pbs.conf` points to the path for the old `PBS_EXEC`.

Edit `/tmp/pbs_backup/pbs.conf.backup` so that the entries look like those in the following table:

**Table 6-1: Entries in Old PBS Configuration File**

New Entry in <code>pbs.conf</code>	Description
<code>PBS_EXEC=&lt;path to PBS_EXEC_OLD&gt;</code>	Location where <code>PBS_EXEC</code> for your old PBS was copied
<code>PBS_HOME=&lt;path to PBS_HOME_OLD&gt;</code>	Location where <code>PBS_HOME</code> for your old PBS was copied
<code>PBS_START_SERVER=1</code>	Unchanged
<code>PBS_START_MOM=1</code>	Unchanged
<code>PBS_START_SCHED=1</code>	Unchanged
<code>PBS_SERVER=&lt;hostname&gt;</code>	Unchanged
<code>PBS_BATCH_SERVICE_PORT=13001</code>	This is the changed port number for the old server
<code>PBS_DATA_SERVICE_PORT=13007</code>	This is the changed port number for the old data service

## 6.7.31 Start the Old Server

You must start the old server in order to move jobs to the new server. The old server must be started on **alternate** ports. These are specified in `/tmp/pbs_backup/pbs.conf.backup`.

Start the old server daemon and point it to the old configuration file:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/sbin/pbs_server
```

## 6.7.32 Verify Old Server is Running on Alternate Ports

Verify that the old `pbs_server` is running on the alternate ports by running the following:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qstat @<old server host>:13001
```

## 6.7.33 Move Existing Jobs to the New Server

You must move existing jobs from the old server to the new server. To do this, you run the `qmove` commands from the old server, and give the new server's port number, 15001, in the destination. See "[qmove](#)" on page 173 of the [PBS Professional Reference Guide](#) or the `qmove (1B)` man page. When moving jobs from reservation queues, be sure to move them into the equivalent new reservation queues.

If your jobs have dependencies, move them according to the order in which they appear in the dependency chain. If job A depends on the outcome of job B, move job B first.

If your old server host also ran a MoM, you will need to delete that vnode from the old server.

Delete the vnode on the old server host:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qmgr -c "d n <old server host>"
<old server host>:13001
```

Move jobs from the old server to the new one:

1. Print the list of jobs on the old server:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qstat @<old server host>:13001
```

2. Move each job from each queue. Make sure that you move jobs in old reservation queues to their counterparts on the new server:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qmove <new queue name>@<new
server host>:15001 <job id>@<old server host>:13001
```

You can use `qselect` to select all the jobs in a queue instead of moving each job individually.

3. Move all jobs in a queue:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup
for jobname in $($PBS_EXEC_OLD/bin/qselect -q <queue name>@<old server host>:13001);
do
    $PBS_EXEC_OLD/bin/qmove <queue name>@<new server host>:15001 ${jobname}@<old server
host>:13001;
done
```

If you see the error message "Too many arguments...", there are too many jobs to fit in the shell's command line buffer. You can continue moving jobs one at a time until there are few enough.

## 6.7.34 Shut Down Old Server

Shut down the old server daemon:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qterm -t quick <old server
host>:13001
```

### 6.7.35 Enable STONITH Script

If your secondary server has a STONITH script, allow the STONITH script to run by setting its permissions to 0755.

### 6.7.36 Enable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, enable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

### 6.7.37 Enable Scheduling

If you disabled scheduling earlier, enable it for the default scheduler and any multischeds:

```
qmgr -c 'set sched <scheduler name> scheduling = true'
```

### 6.7.38 Removing Old PBS

If you decide to remove the old version of PBS after upgrading, **be sure to use the `--noscripts` option** when using `rpm -e`. Using `rpm -e` without this option, even on an older package than the one you are currently using, will cause any currently running PBS daemons to shut down, and will also remove the system V init and/or systemd service startup files. This will prevent PBS daemons from starting automatically at system boot time. If you wish to remove an older RPM without these effects, use `rpm -e --noscripts`.

## 6.8 Upgrading a Windows/Linux Complex

As of version 19.4.1, Windows MoMs and client commands run with a Linux server, scheduler(s), and comm(s). PBS servers, schedulers, and comms run on Linux only. These instructions are for upgrading from a Windows execution host/Linux server complex to a Windows execution host/Linux server complex. If your existing complex is all Windows, see [section 6.9, "Upgrading from an All-Windows Complex", on page 125](#).

You must use a migration upgrade with a Windows/Linux complex. During the migration upgrade, you can install the new version of PBS in the same place or in a new location, which can be the default location or a non-default location.

You will probably want to move jobs from the old system to the new. During a migration upgrade, jobs cannot be running. You can requeue rerunnable jobs. You can let non-rerunnable jobs finish, or you can kill them.

On the Windows hosts, the account from which you install PBS (the installation account) must be a local account that is a member of the local Administrators group on the local computer.

In the instructions below, file and directory pathnames are the PBS defaults. If you installed PBS in different locations, use your locations instead. Where you see `%WINDIR%`, it will be automatically replaced by the correct directory.

The name of the default server host is specified in `/etc/pbs.conf`.

The default installation location on Windows systems is `\Program Files (x86)\PBS\`.

You perform a migration upgrade by copying your old PBS to a temporary location and running it from that temporary location so that you can migrate jobs to the new PBS.

### 6.8.1 Set Paths for Old PBS

To use the following commands without having to substitute actual paths, on the server host, source your `/etc/pbs.conf` file.

We recommend using `/opt` as the location where you'll run your old PBS during the job transfer phase, rather than `/tmp`.

- Choose where you want to copy your old PBS\_EXEC; set PBS\_EXEC\_OLD to this location, and export it
- Choose where you want to copy your old PBS\_HOME; set PBS\_HOME\_OLD to this location, and export it

## 6.8.2 Prevent Jobs From Being Enqueued or Started

You must deactivate the scheduler(s) and queues. When the `scheduling` attribute is `false`, jobs are not started by the scheduler. When the queues' `enabled` attribute is `false`, jobs cannot be enqueued.

1. Prevent the scheduler(s) from starting jobs. Set `scheduling` to `false` for the default scheduler and each multisched:

```
qmgr -c "set sched <scheduler name> scheduling = false"
```

2. Print a list of all queues managed by the server. Save the list of queue names. You will need it in the next step and when moving jobs:

```
qstat -q
```

3. Disable queues to stop jobs from being enqueued. Do this for each queue in your list from the previous step:

```
qdisable <queue name>
```

## 6.8.3 Allow Running Jobs to Finish, or Requeue Them

You cannot perform a migration upgrade while jobs are running. Either let running jobs finish, or requeue them. (You can also delete them.)

To requeue any running jobs:

1. List the jobs. This will list some jobs more than once. You only need to requeue each job once:

```
pbsnodes <hostname> | findstr jobs
```

2. Requeue the jobs:

```
qrerun <job ID> <job ID> ...
```

To kill the jobs:

1. List the jobs. This will list some jobs more than once. You only need to kill each job once:

```
pbsnodes <hostname> | grep jobs
```

2. Use the `qdel` command to kill each job by job ID:

```
qdel <job ID> <job ID> ...
```

To drain the host, wait until any running jobs have finished.

Make sure that there are no old job files on any execution hosts. Remove any of the following:

```
C:\Program Files (x86)\PBS\home\mom_priv\jobs\*.JB
```

## 6.8.4 Disable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, disable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).



## 6.8.5 Disable STONITH Script

If your secondary server has a STONITH script, prevent the STONITH script from running by setting its permissions to 0644.

## 6.8.6 Save Server Host Information To Be Used for New PBS

At the server:

1. Create a backup directory called `/tmp/pbs_backup`  

```
mkdir /tmp/pbs_backup
```
2. Make a copy of the server's configuration for the new PBS:  

```
qmgr -c "print server" > /tmp/pbs_backup/server.new
```
3. Make a copy of the vnode attributes for the new PBS:  

```
qmgr -c "print node @default" > /tmp/pbs_backup/nodes.new
```
4. Make a copy of all scheduler attributes for the new PBS (this prints all settable attributes for the default and multi-scheds):  

```
qmgr -c "print sched" > /tmp/pbs_backup/sched_attrs.new
```
5. Print reservation information to a file:  

```
pbs_rstat -f > /tmp/pbs_backup/reservations
```
6. Make a copy of `pbs.conf` for the new PBS. This command is all one line:  

```
cp /etc/pbs.conf /tmp/pbs_backup/pbs.conf.backup
```
7. Make a copy of each scheduler's directory for the new PBS. For the default scheduler and each multisched:  

```
cp -rp $PBS_HOME/sched_priv /tmp/pbs_backup/sched_priv.new
```

or

```
cp -rp $PBS_HOME/sched_priv_<multisched name> /tmp/pbs_backup/sched_priv_<multisched name>.new
```

## 6.8.7 Save Execution Host Configuration Files

On each PBS execution host, copy the Version 1 and Version 2 configuration files:

1. Make a backup directory:  

```
mkdir "%WINDIR%\TEMP\PBS_MoM_Backup"
```
2. Make a copy of the Version 1 configuration file:  

```
copy "C:\Program Files (x86)\PBS\home\mom_priv\config" "%WINDIR%\TEMP\PBS_MoM_Backup\config.backup"
```
3. Make a copy of the Version 2 configuration files:  

```
mkdir "%WINDIR%\TEMP\PBS_MoM_Backup\mom_config"
for /f %a in (' "C:\Program Files (x86)\PBS\exec\sbin\pbs_mom.exe" -N -s list') do
  "C:\Program Files (x86)\PBS\exec\sbin\pbs_mom.exe" -N -s show %a >
  "%WINDIR%\TEMP\PBS_MoM_Backup\mom_config\%a"
```

## 6.8.8 Save Hooks and Hook Configuration Files

Save your hooks and hook configuration files in ASCII format so you can check them and import them later. The new version of PBS includes a new `pbs_cgroups` hook with a new configuration file. If you use the `cgroups` hook, you must use the new hook and configuration file, but you may want to modify the configuration file, so if you have made any changes to your existing `pbs_cgroups` hook configuration file, you need to save it before you upgrade. Later, you can use the saved information to modify the new configuration file.

For **each** hook:

1. Save the hook. Export the hook:
 

```
# qmgr -c 'export hook <hook name> application/x-python default /tmp/<hook name>.old2.7'
```
2. Save your hook configuration file. Export the configuration file:
 

```
# qmgr -c 'export hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

## 6.8.9 Update Hooks and Hook Configuration Files for New Python

PBS 19.4.1 and later uses Python 3.6, so if you have not already, update all of your site-defined hooks (not the built-in hooks) to Python 3.6. For **each** hook except for the `pbs_cgroups` hook:

1. Update your hook to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your updated hook file differently; use something like `"/tmp/<hook name>.new3.6"`
2. Check that the contents of the configuration file are correct for Python 3.6

## 6.8.10 Shut Down Your Existing PBS

Use the `-t immediate` option to `qterm` so that all possible running jobs will be requeued. If you are using failover, this will stop the secondary server as well:

1. Shut down the server, scheduler, and MoMs:
 

```
qterm -t immediate -m -s -f
```

If your server is not running in a failover environment, the `-f` option is not required.
2. Shut down any multischeds. On each multisched host:
  - a. Find the PID you want:
 

```
ps -ef | grep pbs_sched
```

For the default scheduler, you'll see `"pbs_sched"`, but for multischeds, you'll see `"pbs_sched -I <multisched name>"`.
  - b. Stop the scheduler or multisched:
 

```
kill <multisched PID>
```
3. On the server host and any other comm hosts, shut down the communication daemon:
 

```
systemctl stop pbs
```

or

```
<path to script>/pbs stop
```
4. Verify that PBS daemons are not running in the background:
 

```
ps -ef | grep pbs
```

If you see the `pbs_server`, `pbs_sched`, `pbs_mom`, or `pbs_comm` process running, manually terminate that process. If using failover, check both primary and secondary server hosts:

```
kill -9 <daemon PID>
```

or

```
net stop pbs_mom
```

## 6.8.11 Back Everything Up to Transfer Location

Later, you will run the old PBS server from the backup location while you are moving jobs to the new server. You must do a copy, not a move, because the installation software depends on the old version of PBS being available for it to remove. You'll be running commands from the backup directory, so we recommend a directory under `/opt`.

### 6.8.11.1 Back Up Server/scheduler/communication Host

On the server host, copy the existing `PBS_HOME` and `PBS_EXEC` hierarchies to the backup location.

1. Copy `PBS_HOME` to the backup directory:

```
cp -rp $PBS_HOME $PBS_HOME_OLD
```

2. Copy `PBS_EXEC` to the backup directory:

```
cp -rp $PBS_EXEC $PBS_EXEC_OLD
```

### 6.8.11.2 Back Up Execution Host Information

On each execution host, copy the existing `PBS_HOME` and `PBS_EXEC` hierarchies to the backup location. This is just for safekeeping.

1. Copy `PBS_HOME` to the backup directory:

```
xcopy /o /E /C "C:\Program Files (x86)\PBS\home" %WINDIR%\TEMP\PBS_MoM_Backup
```

2. Copy `PBS_EXEC` to the backup directory:

```
xcopy /o /E /C "C:\Program Files (x86)\PBS\exec" %WINDIR%\TEMP\PBS_MoM_Backup
```

## 6.8.12 Install the New Version of PBS

For a migration upgrade, use `rpm -i` so that the old version of PBS can still be used to move the jobs. You might think that you'd use `rpm -U`, but that removes the old PBS, and you still need it until the jobs are moved.

### 6.8.12.1 Install New PBS Server

On the server host, install the new version of PBS without uninstalling the previous version.

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package as an unprivileged user
4. Make sure that parameters for `PBS_HOME`, `PBS_EXEC`, `PBS_LICENSE_INFO`, `PBS_SERVER` and `PBS_DATA_SERVICE_USER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Make sure that `PBS_HOME` and `PBS_EXEC` are in locations that are different from your existing PBS.

If you are using failover, pay special attention to your configuration parameters, including `PBS_HOME` and `PBS_MOM_HOME`, when installing the server sub-package on the secondary server host. See [section 3.5.2.2, "Setting Installation Parameters", on page 25](#) and ["Configuring the pbs.conf File for Failover" on page 411 in the PBS Professional Administrator's Guide](#).

5. Install the server sub-package:

```
rpm -i --force --prefix=<new PBS_EXEC location> <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.8.12.2 Install New PBS Communication Daemons

If you are installing a communication daemon on a communication-only host, install the server-scheduler-communication-MoM sub-package, and disable the server, scheduler, and MoM on that host. (MoM is disabled by default.) Install the new version of PBS without uninstalling the previous version.

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package
4. Make sure that parameters for `PBS_HOME`, `PBS_EXEC`, and `PBS_SERVER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Make sure that `PBS_HOME` and `PBS_EXEC` point to the locations you are using for the new PBS.
5. Disable the server, scheduler, and MoM. In `pbs.conf`:

```
PBS_START_SERVER=0
PBS_START_SCHED=0
PBS_START_MOM=0
```

6. Install the server sub-package. The method you use depends on the version you are upgrading from.

- When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
- When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.8.12.3 Switch To New PBS\_EXEC Path

On the server host, source your new `/etc/pbs.conf` file.

### 6.8.12.4 Create PBS\_HOME

Create the subdirectories under `PBS_HOME` by running `pbs_habitat`. On the new PBS server host:

```
$PBS_EXEC/libexec/pbs_habitat
```

### 6.8.12.5 Install New PBS MoMs and Client Commands

On each execution and client host, do the following:

1. Log in with the installation account.
2. Install the KB2999226 update for Windows on all Windows Server 2012 execution and client machines.
3. Download the MSI installer (the .msi file).
4. Double-click the MSI installer; the splash screen is displayed.
5. Click the *Next* button to move to the license page. Accept the license.
6. Click the *Next* button and choose the path where you will install the PBS executable. By default this path points to "C:\Program Files (x86)\PBS\".
7. Using "Run As Administrator", open a Command prompt.

### 6.8.12.6 Configure New PBS MoMs and Client Hosts

On each execution and client host, manually execute the `win_postinstall.py` script as shown below. 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 `<domain>\<username>`, pass only *username* as an argument.

On each execution host:

- Delete the "home" folder inside "C:\Program Files (x86)\PBS\" if it exists
- Run `win_postinstall`:
 

```
<PBS_EXEC>\python\python.exe <PBS_EXEC>\etc\win_postinstall.py -u <PBS service account> -p
      <PBS service account password> -s <server name> -t execution -c <path to scp.exe>
```

On each client host:

```
<PBS_EXEC>\python\python.exe <PBS_EXEC>\etc\win_postinstall.py -u <PBS service account> -p <PBS
service account password> -s <server name> -t client -c <path to scp.exe>
```

### 6.8.13 Start and Stop the New Server (If Using Failover)

If you are not using failover, skip this step. If you are using failover, this pair of start and stop steps really is necessary. Bear with us.

When the new server starts up it will have default queue "workq" and the server host already defined. You want to start the new server with empty configurations so that you can import your old settings.

1. Start the new server with empty queue and vnode configurations:
 

```
$PBS_EXEC/sbin/pbs_server -t create
```

A message will appear saying "Create mode and server database exists, do you wish to continue?"

Type "y" to continue.

Because of the new licensing scheme an additional message may appear:

"One or more PBS license keys are invalid, jobs may not run"

---

This message is expected. Continue to the next step in these instructions.

2. Shut down PBS:

```
qterm -t immediate -m -s -f
```

3. Verify that PBS daemons are not running in the background:

```
ps -ef | grep pbs
```

If you see the `pbs_server`, `pbs_sched`, `pbs_comm`, or `pbs_mom` process running, manually terminate that process. If using failover, check both primary and secondary server hosts:

```
kill -9 <daemon PID>
```

## 6.8.14 Start the New Server Without Defined Queues or Vnodes

When the new server starts up it will have default queue "workq" and the server host already defined. You want to start the new server with empty configurations so that you can import your old settings.

Start the new server with empty queue and vnode configurations:

```
$PBS_EXEC/sbin/pbs_server -t create
```

A message will appear saying "Create mode and server database exists, do you wish to continue?"

Type "y" to continue.

Because of the new licensing scheme an additional message may appear:

```
"One or more PBS license keys are invalid, jobs may not run"
```

This message is expected. Continue to the next step in these instructions.

## 6.8.15 Set License Location Server Attribute

Set the `pbs_license_info` server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

## 6.8.16 Clean Up Configuration Information

### 6.8.16.1 Clean Up Scheduler Configuration Files

If you were running one or more multischeds with your old version of PBS, make sure you update their configuration files along with that of the default scheduler. Note that the `preempt_order`, `preempt_prio`, `preempt_queue_prio`, `preempt_sort`, and `log_events` scheduler attributes are new; some were parameters in `sched_config` with the same names. In a later step (after the server is running), you will use `qmgr` to set the attributes. For each scheduler:

1. Make a copy of the new `sched_config`, which is in `PBS_EXEC/etc/pbs_sched_config`.  

```
cp $PBS_EXEC/etc/pbs_sched_config $PBS_EXEC/etc/pbs_sched_config.new
```
2. Update `PBS_EXEC/etc/pbs_sched_config.new` with any modifications that were made to your old scheduler configuration file, saved in (Windows) `"%WINDIR%\TEMP\PBS_Backup\sched_priv.sched_config"` or `"%WINDIR%\TEMP\PBS_Backup\sched_priv_<multisched name>.sched_config"`, or in (Linux) `%PBS_HOME/sched_priv/sched_config` or `%PBS_HOME/sched_priv_<multisched name>/sched_config`.
3. If you were using `vmem` at the queue or server level before the upgrade, then after upgrading you must add `vmem` to the `resource_unset_infinite sched_config` option. Otherwise jobs requesting `vmem` will not run.
4. Move `PBS_EXEC/etc/pbs_sched_config.new` to the correct name and location, i.e. `$PBS_HOME/sched_priv/sched_config` or `$PBS_HOME/sched_priv_<multisched name>/sched_config`:  

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv/sched_config
```

or

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv_<multisched name>/sched_config
```

### 6.8.16.2 Clean Up Scheduler Attributes

For each scheduler, clean up the attributes saved in `/tmp/pbs_backup/<scheduler name>/sched_attrs.new`. When you read in multisched attributes, you'll re-create the multischeds, so make sure your new multischeds are what you want:

- Remove read-only attributes
- Remove lines containing the following:  

```
pbs_version
```

For the new default scheduler and all new multischeds:

- The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).
- The scheduler's `log_filter` configuration parameter is **obsolete**. The scheduler's log filter now uses the same bitmask system as the other daemons. The new default value is `767`. Use `qmgr` to set the scheduler's `log_events` attribute to the value you want. See ["Specifying Scheduler Log Events" on page 550 in the PBS Professional Administrator's Guide](#).

### 6.8.16.3 Clean Up Server Configuration

Remove read-only attributes from the server's configuration information in `server.new`. For example, remove lines containing the following:

```
license_count
pbs_version
```

Remove creation commands for any reservation queues. You will create reservations and their queues separately.

### 6.8.16.4 Copy User Credentials to New Server

PBS caches user credentials in `$PBS_HOME/server_priv/users`. PBS stores the credential for each user in a file named `<username>.CR`. Normally this directory is created by PBS when users log in. If you installed the new version of PBS in the same location as the old one, you do not need to copy user credentials.

However, if the new version of PBS is in a different location, you need to create the directory and copy the credential files, keeping the permissions the same:

1. Create the user credential directory:

```
mkdir -p $PBS_HOME/server_priv/users/
```

2. Copy the user credential files to the new directory:

```
cp -rpu $PBS_HOME_OLD/server_priv/users/* $PBS_HOME/server_priv/users/
```

### 6.8.16.5 Clean up Vnode Configuration

Here you prepare the vnode attribute input to the new `qmgr`.

If your system has multi-vnode hosts:

- Copy your saved node configuration file "`%WINDIR%\TEMP\PBS_Backup\nodes.new`" into two files:
  - `qmgr_parent_vnode.out`, which contains all the configuration information for parent vnodes
  - `qmgr_child_vnode.out`, which contains all the configuration information for vnodes that aren't parent vnodes
- Continue by preparing configuration information for parent vnodes. You will prepare the configuration information for the child vnodes after they have been created, because the vnode names in your file must be precisely the same as the ones created by PBS.

If your system has only single-vnode hosts, follow the steps below for preparing configuration information for parent vnodes only.

#### 6.8.16.5.i Prepare Configuration Information for Parent Vnodes

Edit `qmgr_parent_vnode.out`:

Leave only the the following creation lines:

- Those for parent vnodes
- Any resources you want managed on the server side through `qmgr`
- Custom resources on the parent vnodes

Delete any lines for resources managed through Version 2 configuration files or that MoM reports from what the vnode's host OS is reporting. For example, delete:

- Child vnodes, that should be created by MoM (vnodes that are NOT parent vnodes)
- Lines that set the sharing attribute
- The `ncpus`, `mem`, and `vmem` resources, unless they should explicitly be set via `qmgr`

### 6.8.17 Create and Configure New Multischeds

Create the directories required for each new multisched, and configure each multisched. See "[Creating and Configuring a Multisched](#)" on page 57 in the *PBS Professional Administrator's Guide*.



## 6.8.18 Start New Server and New Schedulers

1. Start the new server and new default scheduler. On the server host:

```
systemctl restart pbs
```

or

```
<path to init.d>/init.d/pbs restart
```

2. Start multischeds. To start a multisched, call `pbs_sched` and specify the name and port you already gave it. For each multisched:

```
pbs_sched -I <name of multisched> -S <same value as sched_port for this multisched>
```

## 6.8.19 Replicate Queue, Server, Scheduler, and Vnode Configurations

### 6.8.19.1 Replicate Server and Queue Attributes

1. Give the new server the old server's configuration, but modified for the new PBS:

```
$PBS_EXEC/bin/qmgr < /tmp/pbs_backup/server.new
```

2. Verify the configuration was read in properly:

```
$PBS_EXEC/bin/qmgr -c "print server"
```

### 6.8.19.2 Replicate Scheduler Attributes

1. Give the new default scheduler the old default scheduler's attributes, and re-create your multischeds:

```
$PBS_EXEC/bin/qmgr < /tmp/pbs_backup/<scheduler name>/sched_attrs.new
```

2. Verify the configurations were read in properly.

You can see all schedulers at once:

```
$PBS_EXEC/bin/qmgr -c "print sched"
```

Or for each scheduler:

```
$PBS_EXEC/bin/qmgr -c "print sched default"
```

or

```
$PBS_EXEC/bin/qmgr -c "print sched <multisched name>"
```

### 6.8.19.3 Replicate Vnode Attributes

Replicate vnode configuration, also modified for the new PBS:

1. Read in the parent vnode configuration file:

```
$PBS_EXEC/bin/qmgr < qmgr_natural_vnode.out
```

2. Wait until MoM or the cgroups hook creates any child vnodes. Check:

```
pbsnodes -av
```

3. Prepare configuration information for child vnodes:

Edit `qmgr_child_vnode.out`. Make sure that the vnode names in this file are exactly what MoM or the cgroups hook created. It's easiest to put all resource information into a Version 2 configuration file, rather than using `qmgr`.

Leave only the the following creation lines:

- Any resources you want managed on the server side through `qmgr`
- Custom resources on the child vnodes (but this may be easier in a Version 2 configuration file)

Delete any lines for resources managed through Version 2 configuration files or that MoM reports from what the vnode's host OS is reporting. For example, delete:

- Child vnodes, that should be created by the cgroups hook or MoM (vnodes that are NOT parent vnodes)
- Lines that set the `sharing` attribute
- The `ncpus`, `mem`, and `vmem` resources, unless they should explicitly be set via `qmgr`

4. Read in the configuration file for child vnodes (not parent vnodes):

```
$PBS_EXEC/bin/qmgr < qmgr_not_natural_vnode.out
```

5. Verify the configurations were read in properly:

```
$PBS_EXEC/bin/pbsnodes -a
```

## 6.8.20 Import and Configure Hooks

Make sure you do not overwrite the new `pbs_cgroups` hook or its configuration file by importing the old ones. Instead, use the saved information from your old hook to modify the new hook and configuration file.

### 6.8.20.1 Import Old Hooks Except for Cgroups Hook

1. Do not import your old `pbs_cgroups` hook. Import your other hooks and their configuration files. For each hook **except** for `pbs_cgroups`:

```
# qmgr -c 'import hook <hook name> application/x-python default /tmp/<hook name>.new3.6'
# qmgr -c 'import hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

## 6.8.20.2 Modify Cgroups Hook Configuration File

If you will use the cgroups hook:

1. Export the new cgroups hook configuration file to `pbs_cgroups.json`:
 

```
# qmgr -c 'export hook pbs_cgroups application/x-config default' > pbs_cgroups.json
```
2. If the cgroups memory subsystem is not mounted on the system, disable 'memory' in the cgroups hook configuration file:
  - a. Check to see whether it is mounted:
 

```
# mount | grep cgroup | grep memory
```

If the memory subsystem is mounted, the command returns something like "cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)".
  - b. If this returns empty, edit the `pbs_cgroups.json` file so that 'enabled' parameter for 'memory' under cgroup is *false*:
 

```
"cgroup": {
  ...
  "memory": {
    "enabled": false,
```
3. If you made changes to the old cgroups configuration file, you may want to make those changes in the new configuration file. Use the information saved in `/etc/pbs_cgroups.old2.7`
4. Import the modified configuration (make sure you use "x-config"):
 

```
# qmgr -c 'import hook pbs_cgroups application/x-config default pbs_cgroups.json'
```

## 6.8.20.3 Enable Cgroups Hook

If you will use the cgroups hook, enable the `pbs_cgroups` hook:

```
qmgr -c "set hook pbs_cgroups enabled=true"
```

## 6.8.20.4 Write and Deploy New Hooks

If you have written new hooks for the new version of PBS, deploy them now. See the *PBS Professional Hooks Guide*.

## 6.8.20.5 Start MoMs

On each execution host, start MoM :

```
net start pbs_mom
```

## 6.8.21 Configure Sharing and Placement Sets

### 6.8.21.1 Configuration with Cgroups Hook

As of version 2020.1, the cgroups hook creates the child vnodes on a multi-vnode machine; if you will use the cgroups hook, it is important that any Version 2 configuration files refer only to these vnodes. Use Version 2 configuration files only to set the `sharing` attribute and optionally to set resources that will be used for placement sets. The default value for the `sharing` attribute of the vnodes is "sharing=default\_shared". You can change this, for example to "sharing=default\_excl".

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named "vnodedefs" that has MoM's list of vnodes; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#)

```
# pbsnodes -av | awk -F'=' '{printf "%s:\tsharing = default_excl\n", $2}' > vnodedefs
```

2. Edit the file to reflect what you want for the `sharing` attribute and placement sets. Use the information saved in `"%WINDIR%\TEMP\PBS_MoM_Backup\mom_config"` in step ["Save Execution Host Configuration Files" on page 111](#)

3. Create your new Version 2 configuration file and name it for example "vnodedefs":

```
# pbs_mom -s insert vnodedefs vnodedefs
```

4. Restart `pbs_mom`:

```
net stop pbs_mom
net start pbs_mom
```

### 6.8.21.2 Configuration without Cgroups Hook

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named "vnodedefs"; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#)

2. Create your new Version 2 configuration file and name it for example "vnodedefs":

```
# pbs_mom -s insert vnodedefs vnodedefs
```

3. Restart `pbs_mom`:

```
net stop pbs_mom
net start pbs_mom
```

### 6.8.22 Start New Communication Daemons

Start PBS on any communication-only hosts. On each communication-only host, type:

```
systemctl start pbs
or
<path to init.d>/init.d/pbs start
```

### 6.8.23 Verify Communication Between Server and MoMs

All new MoMs on all execution hosts should be running and communicating with the new server. Run `pbsnodes -a` on the new server host to see if it can communicate with the execution hosts in your complex. If a host is down, go to the problem host and restart the MoM:

```
net stop pbs_mom
net start pbs_mom
```

## 6.8.24 Re-create Reservations

You must re-create each reservation that was on the old server, using the `pbs_rsub` command. Each reservation is created as a new reservation. You can use all of the information about the old reservation except for its start time. Be sure to give each reservation a start time in the future. Use the information stored in `/tmp/pbs_backup/reservations`.

## 6.8.25 Change Ports and PBS\_EXEC Path in `pbs.conf` for Old PBS

You must edit the `pbs.conf` file of the old PBS so that all old services use ports that won't clash with those of the new PBS. Edit `/tmp/pbs_backup/pbs.conf.backup`.

You must change the port numbers for the PBS server and data service. You do not need to change the port numbers for the comm, MoM, or scheduler.

You must also make sure that the `PBS_EXEC` entry in the old `pbs.conf` points to the path for the old `PBS_EXEC`.

Edit `/tmp/pbs_backup/pbs.conf.backup` so that the entries look like those in the following table:

**Table 6-2: Entries in Old PBS Configuration File**

New Entry in <code>pbs.conf</code>	Description
<code>PBS_EXEC=&lt;path to PBS_EXEC_OLD&gt;</code>	Location where <code>PBS_EXEC</code> for your old PBS was copied
<code>PBS_HOME=&lt;path to PBS_HOME_OLD&gt;</code>	Location where <code>PBS_HOME</code> for your old PBS was copied
<code>PBS_START_SERVER=1</code>	Unchanged
<code>PBS_START_MOM=1</code>	Unchanged
<code>PBS_START_SCHED=1</code>	Unchanged
<code>PBS_SERVER=&lt;hostname&gt;</code>	Unchanged
<code>PBS_BATCH_SERVICE_PORT=13001</code>	This is the changed port number for the old server
<code>PBS_DATA_SERVICE_PORT=13007</code>	This is the changed port number for the old data service

## 6.8.26 Start the Old Server

You must start the old server in order to move jobs to the new server. The old server must be started on **alternate** ports. These are specified in `/tmp/pbs_backup/pbs.conf.backup`.

Start the old server daemon and point it to the old configuration file:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/sbin/pbs_server
```

## 6.8.27 Verify Old Server is Running on Alternate Ports

Verify that the old `pbs_server` is running on the alternate ports by running the following:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qstat @<old server host>:13001
```

## 6.8.28 Move Existing Jobs to the New Server

You must move existing jobs from the old server to the new server. To do this, you run the `qmove` commands from the old server, and give the new server's port number, 15001, in the destination. See "[qmove](#)" on page 173 of the [PBS Professional Reference Guide](#) or the `qmove(1B)` man page. When moving jobs from reservation queues, be sure to move them into the equivalent new reservation queues.

If your jobs have dependencies, move them according to the order in which they appear in the dependency chain. If job A depends on the outcome of job B, move job B first.

If your old server host also ran a MoM, you will need to delete that vnode from the old server.

Delete the vnode on the old server host:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qmgr -c "d n <old server host>"
<old server host>:13001
```

Move jobs from the old server to the new one:

1. Print the list of jobs on the old server:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qstat @<old server host>:13001
```

2. Move each job from each queue. Make sure that you move jobs in old reservation queues to their counterparts on the new server:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qmove <new queue name>@<new
server host>:15001 <job id>@<old server host>:13001
```

You can use `qselect` to select all the jobs in a queue instead of moving each job individually.

3. Move all jobs in a queue:

```
export PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup
for jobname in $($PBS_EXEC_OLD/bin/qselect -q <queue name>@<old server host>:13001);
do
    $PBS_EXEC_OLD/bin/qmove <queue name>@<new server host>:15001 ${jobname}@<old server
host>:13001;
done
```

If you see the error message "Too many arguments...", there are too many jobs to fit in the shell's command line buffer. You can continue moving jobs one at a time until there are few enough.

## 6.8.29 Shut Down Old Server

Shut down the old server daemon:

```
PBS_CONF_FILE=/tmp/pbs_backup/pbs.conf.backup $PBS_EXEC_OLD/bin/qterm -t quick <old server
host>:13001
```

## 6.8.30 Enable STONITH Script

If your secondary server has a STONITH script, allow the STONITH script to run by setting its permissions to 0755.

## 6.8.31 Enable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, enable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

## 6.8.32 Enable Scheduling

If you disabled scheduling earlier, enable it for the default scheduler and any multischeds:

```
qmgr -c 'set sched <scheduler name> scheduling = true'
```

## 6.8.33 Removing Old PBS

If you decide to remove the old version of PBS after upgrading, **be sure to use the `--noscripts` option** when using `rpm -e`. Using `rpm -e` without this option, even on an older package than the one you are currently using, will cause any currently running PBS daemons to shut down, and will also remove the system V init and/or systemd service startup files. This will prevent PBS daemons from starting automatically at system boot time. If you wish to remove an older RPM without these effects, use `rpm -e --noscripts`.

## 6.9 Upgrading from an All-Windows Complex

As of version 19.4.1, Windows MoMs and client commands run with a Linux server, scheduler(s), and comm(s). PBS servers, schedulers, and comms run on Linux only. If you are already using a Linux server with Windows MoMs, see [section 6.8, "Upgrading a Windows/Linux Complex", on page 109](#).

These instructions are for upgrading from a Windows/Windows complex to a Windows/Linux complex.

If your existing complex runs a PBS server on a Windows host, "upgrading" means doing a fresh install for the server/schedulers/comms, and upgrading your Windows MoMs. You cannot preserve any jobs in any state during the upgrade. You can let jobs finish, or you can kill them.

On the Windows hosts, the account from which you install PBS (the installation account) must be a local account that is a member of the local Administrators group on the local computer.

In the instructions below, file and directory pathnames are the PBS defaults. If you installed PBS in different locations, use your locations instead. Where you see `%WINDIR%`, it will be automatically replaced by the correct directory.

The name of the old default server host is specified in `\Program Files (x86)\PBS\pbs.conf`.

On Windows systems, PBS is installed in `\Program Files (x86)\PBS\`.

### 6.9.1 Prevent Jobs From Being Enqueued or Started

You must deactivate the scheduler(s) and queues. When the `scheduling` attribute is `false`, jobs are not started by the scheduler. When the queues' `enabled` attribute is `false`, jobs cannot be enqueued.

1. Prevent the scheduler(s) from starting jobs. Set `scheduling` to `false` for the default scheduler and each multisched:

```
qmgr -c "set sched <scheduler name> scheduling = false"
```

2. Print a list of all queues managed by the server. Save the list of queue names. You will need it in the next step and when moving jobs:

```
qstat -q
```

3. Disable queues to stop jobs from being enqueued. Do this for each queue in your list from the previous step:

```
qdisable <queue name>
```

### 6.9.2 Allow Running Jobs to Finish, or Kill Them

You cannot perform this upgrade while jobs are running or queued. Either let running jobs finish, or kill them.

To drain the host, wait until any running jobs have finished. To kill the jobs:

1. List the jobs. This will list some jobs more than once. You only need to kill each job once:

```
pbsnodes <hostname> | findstr jobs
```

2. Use the qdel command to kill each job by job ID:

```
qdel <job ID> <job ID> ...
```

Make sure that there are no old job files on any execution hosts. Remove any of the following:

```
C:\Program Files (x86)\PBS\home\mom_priv\jobs\*.JB
```

### 6.9.3 Disable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, disable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

### 6.9.4 Disable STONITH Script

If your secondary server has a STONITH script, prevent the STONITH script from running.

### 6.9.5 Save Server Host Information To Be Used for New PBS

At the server:

1. Make a backup directory:

```
mkdir "%WINDIR%\TEMP\PBS_Backup"
```

2. Make a copy of the server's configuration for the new PBS:

```
qmgr -c "print server" > "%WINDIR%\TEMP\PBS_Backup\server.new"
```

3. Make a copy of the vnode attributes for the new PBS:

```
qmgr -c "print node @default" > "%WINDIR%\TEMP\PBS_Backup\nodes.new"
```

4. Make a copy of all scheduler configurations for the new PBS (this prints settable attributes for default and multischeds):

```
qmgr -c "print sched" > "%WINDIR%\TEMP\PBS_Backup\sched_attrs.new"
```

5. Print reservation information to a file:

```
pbs_rstat -f > "%WINDIR%\TEMP\PBS_Backup\reservations"
```

6. Make a copy of pbs.conf for the new PBS. This command is all one line:

```
copy "\Program Files (x86)\PBS\pbs.conf" "%WINDIR%\TEMP\PBS_Backup\pbs.conf.new"
```

7. Make a copy of each scheduler's directory for the new PBS. For the default scheduler and each multisched:

```
xcopy /o /E /C "C:\Program Files (x86)\PBS\home\sched_priv"
"%WINDIR%\TEMP\PBS_Backup\sched_priv.work"
```

or

```
xcopy /o /E /C "C:\Program Files (x86)\PBS\home\sched_priv_<multisched name>"
"%WINDIR%\TEMP\PBS_Backup\sched_priv_<multisched name>.work"
```



When you see this message:

Does C:\Windows\TEMP\PBS\_Backup\sched\_priv.work specify a file name or directory name on the target (F = file, D = directory)?

Type this:

D

## 6.9.6 Save Execution Host Configuration Files

On each PBS execution host, copy the Version 1 and Version 2 configuration files:

1. Make a backup directory:

```
mkdir "%WINDIR%\TEMP\PBS_MoM_Backup"
```

2. Make a copy of the Version 1 configuration file:

```
copy "C:\Program Files (x86)\PBS\home\mom_priv\config" "%WINDIR%\TEMP\PBS_MoM_Backup\config.backup"
```

3. Make a copy of the Version 2 configuration files:

```
mkdir "%WINDIR%\TEMP\PBS_MoM_Backup\mom_config"
```

```
for /f %a in (' "C:\Program Files (x86)\PBS\exec\sbin\pbs_mom.exe" -N -s list') do
  "C:\Program Files (x86)\PBS\exec\sbin\pbs_mom.exe" -N -s show %a >
  "%WINDIR%\TEMP\PBS_MoM_Backup\mom_config\%a"
```

## 6.9.7 Save Hooks and Hook Configuration Files

Save your hooks and hook configuration files in ASCII format so you can check them and import them later. The new version of PBS includes a new pbs\_cgroups hook with a new configuration file. If you use the cgroups hook, you must use the new hook and configuration file, but you may want to modify the configuration file, so if you have made any changes to your existing pbs\_cgroups hook configuration file, you need to save it before you upgrade. Later, you can use the saved information to modify the new configuration file.

For **each** hook:

1. Save the hook. Export the hook:

```
# qmgr -c 'export hook <hook name> application/x-python default %WINDIR%\TEMP\PBS_Backup\<hook name>.old2.7'
```

2. Save your hook configuration file. Export the configuration file:

```
# qmgr -c 'export hook <hook name> application/x-config default %WINDIR%\TEMP\PBS_Backup\<hook name>.configcheck'
```

3. Run dos2unix to convert the hooks and hook configuration files from DOS to UNIX format:

```
dos2unix /tmp/pbs_backup/<saved file>
```

## 6.9.8 Update Hooks and Hook Configuration Files for New Python

PBS 19.4.1 and later uses Python 3.6, so if you have not already, update all of your site-defined hooks (not the built-in hooks) to Python 3.6. For **each** hook except for the `pbs_cgroups` hook:

1. Update your hook to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your updated hook file differently; use something like "WINDIR%\TEMP\PBS\_Backup\- 2. Check that the contents of the configuration file are correct for Python 3.6

## 6.9.9 Shut Down Your Existing PBS

Use the `-t immediate` option to `qterm` so that all possible running jobs will be requeued. If you are using failover, this will stop the secondary server as well:

1. Shut down the server, scheduler, and MoMs:

```
qterm -t immediate -m -s -f
```

If your server is not running in a failover environment, the "-f" option is not required.

2. Shut down any multischeds. On each multisched host:

```
net stop pbs_sched
```

3. On the server host and any other comm hosts, shut down the communication daemon:

```
net stop pbs_comm
```

## 6.9.10 Install the New Version of PBS

### 6.9.10.1 Install New PBS Server

On the server host, install the new version of PBS without uninstalling the previous version.

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package as an unprivileged user
4. Make sure that parameters for `PBS_HOME`, `PBS_EXEC`, `PBS_LICENSE_INFO`, `PBS_SERVER` and `PBS_DATA_SERVICE_USER` are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Make sure that `PBS_HOME` and `PBS_EXEC` are in locations that are different from your existing PBS.

If you are using failover, pay special attention to your configuration parameters, including `PBS_HOME` and `PBS_MOM_HOME`, when installing the server sub-package on the secondary server host. See [section 3.5.2.2, "Setting Installation Parameters", on page 25](#) and ["Configuring the pbs.conf File for Failover" on page 411 in the PBS Professional Administrator's Guide](#).

5. Install the server sub-package:

```
rpm -i --prefix=<new PBS_EXEC location> <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.9.10.2 Install New PBS Communication Daemons

If you are installing a communication daemon on a communication-only host, install the server-scheduler-communication-MoM sub-package, and disable the server, scheduler, and MoM on that host. (MoM is disabled by default.) Install the new version of PBS without uninstalling the previous version.

1. Log in as root
2. Download the appropriate PBS package
3. Uncompress the package
4. Make sure that parameters for PBS\_HOME, PBS\_EXEC, and PBS\_SERVER are set correctly; see [section 3.5.2.2, "Setting Installation Parameters", on page 25](#). Make sure that PBS\_HOME and PBS\_EXEC point to the locations you are using for the new PBS.
5. Disable the server, scheduler, and MoM. In `pbs.conf`:

```
PBS_START_SERVER=0
PBS_START_SCHED=0
PBS_START_MOM=0
```

6. Install the server sub-package. The method you use depends on the version you are upgrading from.

- When upgrading from 13.2 or an earlier version:
 

```
rpm -i <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```
- When upgrading from 14.2 or a later version:
 

```
rpm -U <path/to/server sub-package>/pbspro-server-<version>-0.<platform-specific-dist-tag>.<hardware>.rpm
```

Do **not** start PBS now.

### 6.9.10.3 Create PBS\_HOME

Create the subdirectories under PBS\_HOME by running `pbs_habitat`. On the new PBS server host:

```
$PBS_EXEC/libexec/pbs_habitat
```

### 6.9.10.4 Install New PBS MoMs and Client Commands

On each execution and client host, do the following:

1. Log in with the installation account.
2. Install the KB2999226 update for Windows on all Windows Server 2012 execution and client machines.
3. Download the MSI installer (the `.msi` file).
4. Double-click the MSI installer; the splash screen is displayed.
5. Click the *Next* button to move to the license page. Accept the license.
6. Click the *Next* button and choose the path where you will install the PBS executable. By default this path points to "C:\Program Files (x86)\PBS\".
7. Using "Run As Administrator", open a Command prompt.

### 6.9.10.5 Configure New PBS MoMs and Client Hosts

On each execution and client host, manually execute the `win_postinstall.py` script as shown below. 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 `<domain>\<username>`, pass only `username` as an argument.

On each execution host:

- Delete the "home" folder inside "C:\Program Files (x86)\PBS\" if it exists
- Run `win_postinstall`:  

```
<PBS_EXEC>\python\python.exe <PBS_EXEC>\etc\win_postinstall.py -u <PBS service account> -p
<PBS service account password> -s <server name> -t execution -c <path to scp.exe>
```

On each client host:

```
<PBS_EXEC>\python\python.exe <PBS_EXEC>\etc\win_postinstall.py -u <PBS service account> -p <PBS
service account password> -s <server name> -t client -c <path to scp.exe>
```

### 6.9.11 Start the New Server Without Defined Queues or Vnodes

When the new server starts up it will have default queue "workq" and the server host already defined. You want to start the new server with empty configurations so that you can import your old settings.

Start the new server with empty queue and vnode configurations:

```
$PBS_EXEC/sbin/pbs_server -t create
```

A message will appear saying "Create mode and server database exists, do you wish to continue?"

Type "y" to continue.

Because of the new licensing scheme an additional message may appear:

```
"One or more PBS license keys are invalid, jobs may not run"
```

This message is expected. Continue to the next step in these instructions.

### 6.9.12 Set License Location Server Attribute

Set the `pbs_license_info` server attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

## 6.9.13 Clean Up Configuration Information

### 6.9.13.1 Clean Up Scheduler Configuration Files

If you were running one or more multischeds with your old version of PBS, make sure you update their configuration files along with that of the default scheduler. Note that the `preempt_order`, `preempt_prio`, `preempt_queue_prio`, `preempt_sort`, and `log_events` scheduler attributes are new; some were parameters in `sched_config` with the same names. In a later step (after the server is running), you will use `qmgr` to set the attributes. For each scheduler:

1. Make a copy of the new `sched_config`, which is in `PBS_EXEC/etc/pbs_sched_config`.  

```
cp $PBS_EXEC/etc/pbs_sched_config $PBS_EXEC/etc/pbs_sched_config.new
```
2. Update `PBS_EXEC/etc/pbs_sched_config.new` with any modifications that were made to your old scheduler configuration file, saved in (Windows) `"%WINDIR%\TEMP\PBS_Backup\sched_priv.sched_config"` or `"%WINDIR%\TEMP\PBS_Backup\sched_priv_<multisched name>.sched_config"`, or in (Linux) `%PBS_HOME/sched_priv/sched_config` or `%PBS_HOME/sched_priv_<multisched name>/sched_config`.
3. If you were using `vmem` at the queue or server level before the upgrade, then after upgrading you must add `vmem` to the `resource_unset_infinite sched_config` option. Otherwise jobs requesting `vmem` will not run.
4. Move `PBS_EXEC/etc/pbs_sched_config.new` to the correct name and location, i.e. `$PBS_HOME/sched_priv/sched_config` or `$PBS_HOME/sched_priv_<multisched name>/sched_config`:  

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv/sched_config
```

or

```
mv $PBS_EXEC/etc/pbs_sched_config.new $PBS_HOME/sched_priv_<multisched name>/sched_config
```

### 6.9.13.2 Clean Up Scheduler Attributes

For each scheduler, clean up the attributes saved in `"%WINDIR%\TEMP\PBS_Backup\sched_attrs.new"`. When you read in multisched attributes, you'll re-create the multischeds, so make sure your new multischeds are what you want:

- Remove read-only attributes
- Remove lines containing the following:  

```
pbs_version
```

For the new default scheduler and all new multischeds:

- The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).
- The scheduler's `log_filter` configuration parameter is **obsolete**. The scheduler's log filter now uses the same bitmask system as the other daemons. The new default value is `767`. Use `qmgr` to set the scheduler's `log_events` attribute to the value you want. See ["Specifying Scheduler Log Events" on page 550 in the PBS Professional Administrator's Guide](#).

### 6.9.13.3 Clean Up Server Configuration

Remove read-only attributes from the server's configuration information in `server.new`. For example, remove lines containing the following:

```
license_count
pbs_version
```

Remove creation commands for any reservation queues. You will create reservations and their queues separately.

### 6.9.13.4 Clean up Vnode Configuration

Here you prepare the vnode attribute input to the new `qmgr`.

If your system has multi-vnode hosts:

- Copy your saved node configuration file "`%WINDIR%\TEMP\PBS_Backup\nodes.new`" into two files:
  - `qmgr_parent_vnode.out`, which contains all the configuration information for parent vnodes
  - `qmgr_child_vnode.out`, which contains all the configuration information for vnodes that aren't parent vnodes
- Continue by preparing configuration information for parent vnodes. You will prepare the configuration information for the child vnodes after they have been created, because the vnode names in your file must be precisely the same as the ones created by PBS.

If your system has only single-vnode hosts, follow the steps below for preparing configuration information for parent vnodes only.

#### 6.9.13.4.i Prepare Configuration Information for Parent Vnodes

Edit `qmgr_natural_vnode.out`:

Leave only the the following creation lines:

- Those for parent vnodes
- Any resources you want managed on the server side through `qmgr`
- Custom resources on the parent vnodes

Delete any lines for resources managed through Version 2 configuration files or that MoM reports from what the vnode's host OS is reporting. For example, delete:

- Child vnodes, that should be created by MoM (vnodes that are NOT parent vnodes)
- Lines that set the sharing attribute
- The `ncpus`, `mem`, and `vmem` resources, unless they should explicitly be set via `qmgr`

## 6.9.14 Create and Configure New Multischeds

Create the directories required for each new multisched, and configure each multisched. See "[Creating and Configuring a Multisched](#)" on page 57 in the *PBS Professional Administrator's Guide*.

## 6.9.15 Start New Server and New Schedulers

1. Start the new server and new default scheduler. On the server host:

```
systemctl restart pbs
```

or

```
<path to init.d>/init.d/pbs restart
```

2. Start multischeds. To start a multisched, call `pbs_sched` and specify the name and port you already gave it. For each multisched:

```
pbs_sched -I <name of multisched> -S <same value as sched_port for this multisched>
```

## 6.9.16 Replicate Queue, Server, Scheduler, and Vnode Configurations

### 6.9.16.1 Replicate Server and Queue Attributes

1. Give the new server the old server's configuration, but modified for the new PBS:

```
$PBS_EXEC/bin/qmgr < /tmp/pbs_backup/server.new
```

2. Verify the configuration was read in properly:

```
$PBS_EXEC/bin/qmgr -c "print server"
```

### 6.9.16.2 Replicate Scheduler Attributes

1. Give the new default scheduler the old default scheduler's attributes, and re-create your multischeds:

```
$PBS_EXEC/bin/qmgr < /tmp/pbs_backup/<scheduler name>/sched_attr.new
```

2. Verify the configurations were read in properly.

You can see all schedulers at once:

```
$PBS_EXEC/bin/qmgr -c "print sched"
```

Or for each scheduler:

```
$PBS_EXEC/bin/qmgr -c "print sched default"
```

or

```
$PBS_EXEC/bin/qmgr -c "print sched <multisched name>"
```

### 6.9.16.3 Replicate Vnode Attributes

Replicate vnode configuration, also modified for the new PBS:

1. Read in the parent vnode configuration file:

```
$PBS_EXEC/bin/qmgr < qmgr_natural_vnode.out
```

2. Wait until MoM or the cgroups hook creates any child vnodes. Check:

```
pbsnodes -av
```

3. Prepare configuration information for child vnodes:

Edit `qmgr_not_natural_vnode.out`. Make sure that the vnode names in this file are exactly what MoM or the cgroups hook created. It's easiest to put all resource information into a Version 2 configuration file, rather than using `qmgr`.

Leave only the the following creation lines:

- Any resources you want managed on the server side through `qmgr`
- Custom resources on the child vnodes (but this may be easier in a Version 2 configuration file)

Delete any lines for resources managed through Version 2 configuration files or that MoM reports from what the vnode's host OS is reporting. For example, delete:

- Child vnodes, that should be created by the cgroups hook or MoM (vnodes that are NOT parent vnodes)
- Lines that set the sharing attribute
- The ncpus, mem, and vmem resources, unless they should explicitly be set via qmgr

4. Read in the configuration file for child vnodes (not parent vnodes):

```
$PBS_EXEC/bin/qmgr < qmgr_not_natural_vnode.out
```

5. Verify the configurations were read in properly:

```
$PBS_EXEC/bin/pbsnodes -a
```

## 6.9.17 Import and Configure Hooks

Make sure you do not overwrite the new pbs\_cgroups hook or its configuration file by importing the old ones. Instead, use the saved information from your old hook to modify the new hook and configuration file.

### 6.9.17.1 Import Old Hooks Except for Cgroups Hook

1. Do not import your old pbs\_cgroups hook. Import your other hooks and their configuration files. For each hook **except** for pbs\_cgroups:

```
# qmgr -c 'import hook <hook name> application/x-python default /tmp/<hook name>.new3.6'
# qmgr -c 'import hook <hook name> application/x-config default /tmp/<hook name>.configcheck'
```

### 6.9.17.2 Modify Cgroups Hook Configuration File

If you will use the cgroups hook:

1. Export the new cgroups hook configuration file to pbs\_cgroups.json:
 

```
# qmgr -c 'export hook pbs_cgroups application/x-config default' > pbs_cgroups.json
```
2. If the cgroups memory subsystem is not mounted on the system, disable 'memory' in the cgroups hook configuration file:

- a. Check to see whether it is mounted:

```
# mount | grep cgroup | grep memory
```

If the memory subsystem is mounted, the command returns something like "cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)".

- b. If this returns empty, edit the pbs\_cgroups.json file so that 'enabled' parameter for 'memory' under cgroup is *false*:

```
"cgroup": {
  ...
  "memory": {
    "enabled": false,
```

3. If you made changes to the old cgroups configuration file, you may want to make those changes in the new configuration file. Use the information saved in /etc/pbs\_cgroups.old2.7
4. Import the modified configuration (make sure you use "x-config"):

```
# qmgr -c 'import hook pbs_cgroups application/x-config default pbs_cgroups.json'
```



### 6.9.17.3 Enable Cgroups Hook

If you will use the cgroups hook, enable the pbs\_cgroups hook:

```
qmgr -c "set hook pbs_cgroups enabled=true"
```

### 6.9.17.4 Write and Deploy New Hooks

If you have written new hooks for the new version of PBS, deploy them now. See the *PBS Professional Hooks Guide*.

### 6.9.17.5 Start MoMs

On each execution host, start MoM :

```
net start pbs_mom
```

## 6.9.18 Configure Sharing and Placement Sets

### 6.9.18.1 Configuration with Cgroups Hook

As of version 2020.1, the cgroups hook creates the child vnodes on a multi-vnode machine; if you will use the cgroups hook, it is important that any Version 2 configuration files refer only to these vnodes. Use Version 2 configuration files only to set the `sharing` attribute and optionally to set resources that will be used for placement sets. The default value for the `sharing` attribute of the vnodes is `"sharing=default_shared"`. You can change this, for example to `"sharing=default_excl"`.

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named "vnodedefs" that has MoM's list of vnodes; see ["Version 2 Vnode Configuration Files" on page 44 in the PBS Professional Administrator's Guide](#)
2. Edit the file to reflect what you want for the `sharing` attribute and placement sets. Use the information saved in `"%WINDIR%\TEMP\PBS_MoM_Backup\mom_config"` in step ["Save Execution Host Configuration Files" on page 127](#)
3. Create your new Version 2 configuration file and name it for example "vnodedefs":

```
# pbs_mom -s insert vnodedefs vnodedefs
```

4. Restart pbs\_mom:

```
net stop pbs_mom
net start pbs_mom
```

### 6.9.18.2 Configuration without Cgroups Hook

Do **not** set `resources_available.mem`, `resources_available.ncpus`, or `resources_available.vmem` in the Version 2 configuration file.

On each execution host:

1. Create a file named "vnodedefs"; see "[Version 2 Vnode Configuration Files](#)" on page 44 in the *PBS Professional Administrator's Guide*

2. Insert your new Version 2 configuration file and name it for example "vnodedefs":

```
# pbs_mom -s insert vnodedefs vnodedefs
```

3. Restart pbs\_mom:

```
net stop pbs_mom
net start pbs_mom
```

## 6.9.19 Start New Communication Daemons

Start PBS on any communication-only hosts. On each communication-only host, type:

```
systemctl start pbs
or
<path to init.d>/init.d/pbs start
```

## 6.9.20 Verify Communication Between Server and MoMs

All new MoMs on all execution hosts should be running and communicating with the new server. Run `pbsnodes -a` on the new server host to see if it can communicate with the execution hosts in your complex. If a host is down, go to the problem host and restart the MoM:

```
net stop pbs_mom
net start pbs_mom
```

## 6.9.21 Re-create Reservations

You must re-create each reservation that was on the old server, using the `pbs_rsub` command. Each reservation is created as a new reservation. You can use all of the information about the old reservation except for its start time. Be sure to give each reservation a start time in the future. Use the information stored in `/tmp/pbs_backup/reservations`.

## 6.9.22 Enable STONITH Script

If your secondary server has a STONITH script, allow the STONITH script to run by setting its permissions to 0755.

## 6.9.23 Enable Cloud Bursting

If you are using Altair Control for cloud bursting with PBS, enable cloud bursting. See the *Altair Control Administrator's Guide*, at [www.pbsworks.com](http://www.pbsworks.com).

## 6.9.24 Enable Scheduling

If you disabled scheduling earlier, enable it for the default scheduler and any multischeds:

```
qmgr -c 'set sched <scheduler name> scheduling = true'
```

## 6.10 After Upgrading

### 6.10.1 Making Upgrade Transparent for Users

You may wish to make the upgrade transparent for users, if the installation program hasn't done that already. See [section 3.5.5, "Making User Paths Work", on page 36](#).



# Installing and Upgrading on Cray

## 7.1 Installing PBS with Shasta

To install the PBS server and client packages on Shasta, follow the instructions supplied by Cray.

To install the PBS MoM and comm packages on Shasta, follow the standard Linux instructions. You can install comms on compute nodes. See [Chapter 3, "Installation", on page 19](#).

### 7.1.1 Prerequisites for PBS on Shasta

If you want to be able to use `pbs_snapshot`, install the `file` command.

## 7.2 Prerequisites for Using Power Profiles with Cray XC

- The Cray XC `capmc` package must be running on XC30 or XC40 hardware with SMW software release 7.2.UP02 or later.
- Make sure that the `capmc` utility is available and working correctly in your PBS complex.

## 7.3 Support for IMPS and CLE 6 and 7

With CLE 6 and 7, Cray XC introduces a new mechanism to manage system images, because Cray XC is no longer using `xtopview` on a shared root. The IMPS (Image Management and Provisioning System) interface includes new commands to install and boot on a Cray XC. PBS Professional uses this new Cray XC installation mechanism to include PBS in the images.

This version of PBS allows a PBS SLES12 package to be added to a Cray XC image recipe, as well as building and installing an image using Cray XC's new IMPS command-line interfaces.

### 7.3.1 Prerequisites for IMPS

This version of PBS relies on the following Cray XC interfaces:

`recipe`

`image`

`cfgset`

image customization via `postbuild_copy` and `postbuild_chroot`

`cnode`

`xtbootsys`

---

## 7.3.2 Where to Run PBS Daemons for IMPS

MoMs may be run on any nodes where ALPS is configured, and the PBS server may be run on any nodes that have access to the license server.

## 7.4 Caveats and Advice for Installing and Upgrading on the Cray XC

- If you install the server/scheduler on a non-CLE machine, any automatic installation behavior on that machine is not Cray XC-specific. Therefore you need to make sure that all of your settings on that host are appropriate. See [section 7.5.2.1, "Installing PBS on a Cray XC esLogin Node or Repurposed Compute Node"](#), on page 142.
- When upgrading PBS on a Cray XC, you can use an overlay upgrade.
- All MoMs on X\* series Cray machines must be the same version.
- For a list of supported operating systems, see the release notes.
- On CLE 5.2UP03, if you are using preemption via suspend/resume, you may see that after a SWITCH request, ALPS continues to start new applications, so that the system is never ready for PBS to suspend job processes. The system can go from RUNNING to EMPTY to RUNNING.
- If you start PBS with `PBS_START_MOM = 0` for the first time, PBS does not create MoM directories on that host. If you want to use the machine as an execution host (running a MoM on it), set `PBS_START_MOM = 1`, run `pbs_habitat`, then start MoM.
- Beware of overflow: PBS stores fairshare allocations in a signed integer (32-bit on Linux x86\_64 platforms), and fairshare usage in a long (64-bit on Linux x86\_64 platforms).
- If you are upgrading and using failover, do not upgrade the primary and secondary servers simultaneously. Upgrade the primary first, then once that is complete, upgrade the secondary.

### 7.4.1 Where to Run PBS Server, Scheduler, Comm Daemons

The PBS Professional server, scheduler, and communication daemons can be run on an external host; they do not have to run on your Cray system. One instance of the `pbs_mom` daemon is run on each login node of the Cray XC system, allowing PBS Professional to select the login node to which a job is assigned.

The PBS Professional scheduler, server, and communication daemons can be run on any UNIX/Linux machine or node that fits the requirements. We include instructions for all options.

- Make sure you choose the correct PBS package for the server/scheduler/communication host or node. If you are installing PBS components on a Cray XC service node, install from the Cray XC package. If you are installing on a non-Cray XC host, choose the package for that host's OS.
- Make sure that the server/scheduler/communication host or node can reach the license server host.
- Make sure that the server/scheduler/communication host or node can communicate with the login nodes where PBS MoMs are running.
- Make sure that the server/scheduler/communication host meets the minimum requirements to run PBS. See [section 2.1.2, "Resources Required by PBS"](#), on page 7.

## 7.4.2 Licensing PBS

If you intend to use socket licenses for your Cray XC, you must have enough socket licenses for all of the sockets on the entire machine. Each PBS complex is licensed using either PBSProNodes or PBSProSockets licenses, but not a mix.

Install and configure the Altair license server before you install PBS. See the *Altair License Management System Installation and Operations Guide*, available at [www.altair.com](http://www.altair.com) for information on installing and configuring the Altair license server(s).

To license a PBS complex, set the server `pbs_license_info` attribute to the location of the license server:

```
# qmgr -c 'set server pbs_license_info=<port>@<license server hostname>'
```

## 7.4.3 Hardware and Software Requirements

Confirm the following before beginning installation:

- You must have root permission.
- Make sure that all configuration files have been updated appropriately for your site.
- On the login nodes and on the host or node on which the PBS server and Scheduler will run, make sure that the `/var` directory is owned by root.
- Make sure your ALM license server is up and running and has enough licenses for your site.
- Compute nodes must be in the "batch mode" before they can be used by jobs under the control of PBS Professional. To set all of the compute nodes in the batch mode:

On the boot node execute:

```
xtprocadmin -k m batch
```

## 7.4.4 Caveats When Upgrading to PBS 2021.1.2

- If you have an existing queue that has a dot (".") in its name, PBS 2021.1.2 will not accept the queue name. The queue name specification in previous versions of PBS did not include a dot, but this was not enforced, and names with a dot could slide by. However, PBS 2021.1.2 enforces the specification and does not accept dots in queue names. The format for the name of a queue is a "PBS NAME"; for a description of PBS NAME, see the "Formats" chapter in the PBS Professional Reference Guide.
- PBS vnodes representing some types of Xeon Phi nodes do not have values for the `PBSscrayorder` and `PBSscraylabel` resources. Vnodes representing other kinds of nodes do have values for those resources.
- Unlike prior Cray CLE versions (Pearl and earlier), with CLE 6 and 7 `/var/spool` is not persistent by default. When you reboot a node, any `PBS_HOME/*_priv` configuration changes you may have made will be gone unless you use Cray's `cfgset` command to make particular directories (such as `PBS_HOME`) persistent. Please refer to Cray's `cfgset` man page for more information.
- This version of PBS supports IMPS but only so far as is necessary to support CLE 6 and 7.
- The provisioning hook called `PBS_xeon_phi_provision` that is shipped with this version of PBS is disabled by default. You must enable it according to the instructions in the PBS Professional Administrator's Guide:

```
Qmgr: set pbshook PBS_xeon_phi_provision enabled=true
```

- If you installed PBS on a Cray XC or a non-Cray headnode, make sure that the inventory hook, `PBS_alps_inventory_check`, is enabled. List it:

```
qmgr -c "list pbshook PBS_alps_inventory_check"
```

If the hook is not enabled:

```
qmgr -c "set pbshook PBS_alps_inventory_check enabled=true"
```

## 7.5 Installing PBS on the Cray XC

### 7.5.1 Changes for Cray XC Installation

- The default installation directory is now `/opt/pbs`. When PBS is installed on systems using the `modules` package (e.g. `module load pbs`), `PATH` is updated to include `/opt/pbs/bin` and `MANPATH` is updated to include `/opt/pbs/man`. The PBS module file is in `PBS_EXEC/etc/modulefile`.
- The order in which spec file scripts are run during an upgrade is described in <https://www.ibm.com/developer-works/library/l-rpm2/>. Because the spec file scripts were removing the PBS init script during an upgrade, the spec file scripts have been updated to avoid deleting this file and disabling PBS via `chkconfig` during an upgrade. When uninstalling (e.g. `rpm -e`), these instructions are called and PBS is disabled.
- On CLE 5.2, all PBS services are disabled by default in `/etc/pbs.conf`. On non-Cray XC systems, PBS services are enabled based on the type of installation. On Cray CLE 5.2 systems, additional configuration steps are required to specialize `/etc/pbs.conf` on various nodes where the services will run. As a result, `PBS_START_SERVER`, `PBS_START_SCHED`, `PBS_START_COMM`, and `PBS_START_MOM` are all set to zero by default on Cray CLE 5.2 systems only.
- The `$alps_client` entry in the MoM config file is automatically added for Cray X\* series systems. The lines `"$vnodedef_additive 0"` and `"$alps_client /opt/cray/alps/default/bin/apbasil"` are added automatically on Cray XC systems only.

### 7.5.2 Installation Notes

#### 7.5.2.1 Installing PBS on a Cray XC esLogin Node or Repurposed Compute Node

To install the MoM on a Cray XC esLogin node or a repurposed compute node, you should follow the appropriate install procedure for that operating system. You may use either:

- The "PBS for Cray XC" binaries package
  - You should not set `$alps_client` (because there is no ALPS client to connect to)
  - Because this binary expects to be able to talk to ALPS, you will see warnings and/or errors in the MoM logs because PBS complains in the logs when it can't talk to ALPS
- The appropriate Linux package

If the scheduler is installed on an esLogin node or repurposed compute node:

- Modify the `<sched_priv directory>/sched_config` file `resources:` line to add `"vntype"`
- Set `restrict_res_to_release_on_suspend` to `"ncpus"`

### 7.5.3 Installing PBS on CLE 6 and 7

#### 7.5.3.1 Caveats for Installing PBS on CLE 6 and 7

When you install PBS on CLE 6 and 7, some directories and files are not created or configured until PBS is started for the first time. These include `PBS_HOME` and `/etc/pbs.conf`.

PBS will report only the compute nodes that are up and in batch mode.



### 7.5.3.2 Installation Notes for CLE 6 and 7

These instructions rely on Cray's Image Management and Provisioning System (IMPS) features, and are based on the information in the section titled "Install Third-Party Software with a Custom Image" of the Cray document S-2559. See section "Miscellaneous Installation and Configuration Procedures", sub-section "Install Third-Party Software with a Custom Image Recipe". We list this and other useful Cray documents in [section 7.9, "Cray XC References", on page 158](#). Please use the newest versions of the Cray documents.

### 7.5.3.3 Installation Steps for CLE 6 and 7

1. Create a repository. Follow Cray's instructions in the section titled "Install Third-Party Software with a Custom Image" in the *Software Installation and Configuration Guide* in Cray document S-2559.
2. Create a package collection. Follow Cray's instructions in Cray document S-2559. Use the RPM named "pbspro\_server-\*".
3. Create a new recipe and update it with the PBS Professional RPM. Follow Cray's instructions in Cray document S-2559.

Each recipe must be based on the type of node where the image will run. For example, a service node is in the service image as a base sub-recipe named "service\_cle\_6.0up01\_sles\_12\_x86-64\_ari".

If you will run the PBS Professional server/scheduler/comm on the sdb node and the MoM on the login nodes, you will need at least two different recipes, one for the server/scheduler/comm and one for the MoMs.

- a. Add the other RPMs required by PBS Professional to the recipe. Syntax:
    1. Provide `/bin/sh`. Add the repo named "sle-server\_<version>\_x86-64"
 

```
smw# recipe update -r sle-server_<version>_x86-64 <recipe name>
```

 For example:
 

```
smw# recipe update -r sle-server_12sp3_x86-64 <recipe name>
```
    2. Cray XC requires adding the base and base\_updates repository pairs; add "sle-server\_<version>\_x86-64\_updates". Syntax:
 

```
smw# recipe update -r sle-server_<server>_x86-64_updates <recipe name>
```

 For example:
 

```
smw# recipe update -r sle-server_12sp3_x86-64_updates <recipe name>
```
  - b. Validate the image recipe. Follow Cray's instructions in Cray document S-2559.
  - c. Repeat step 3 including parts a and b for each different type of node.
4. Build and push/package the image; follow Cray's instructions in Cray document S-2559.
  5. If you are upgrading PBS, make sure that `PBS_EXEC = /opt/pbs`, and that `PBS_HOME` is set to the same path as before the upgrade.
  6. Make `/etc/pbs.conf` persistent by creating any desired configuration sets. See Cray's documentation on configuration sets and the configurator.

On each host where a PBS daemon or command will run, edit the Cray XC Ansible play named "simple sync" to include the information for each `/etc/pbs.conf`.

- a. It may be necessary to create different persistent `/etc/pbs.conf` files depending on how the system will be configured. When making the `/etc/pbs.conf` file persistent, include the following (at a minimum):
 

```
PBS_EXEC=/opt/pbs
PBS_SERVER=<hostname of the machine hosting the server>
PBS_START_SERVER=<set this to 1 if server is to run on this host, 0 otherwise>
PBS_START_SCHED=<set this to 1 if scheduler is to run on this host, 0 otherwise>
PBS_START_COMM=<set this to 1 if a communication daemon is to run on this host, 0 otherwise>
PBS_START_MOM=<set this to 1 if MoM is to run on this host, 0 otherwise>
PBS_HOME=/var/spool/pbs
PBS_CORE_LIMIT=unlimited
PBS_SCP=/usr/bin/scp
```
  - b. Add any other configuration changes to `/etc/pbs.conf` that need to persist across reboots.
7. Assign the new boot image to the nodes where PBS Professional is to be installed. This includes all nodes running

PBS Professional daemons (e.g. `pbs_server`, `pbs_mom`, etc.) and those requiring access to PBS commands (e.g. `qsub`, `qstat`, etc.). Include the configuration set from step 6 that contains the information that needs to be persistent. Follow Cray's instructions in Cray document S-2559. General syntax:

```
cnode update -i /var/opt/cray/imps/boot_images/<image name>.cpio -c <configuration set name>
<cname-of-node-to-update or group-to-update>
```

8. Reboot the system. Follow Cray's instructions in Cray document S-2559.
9. If you are upgrading PBS and `PBS_HOME` is already persistent, you can skip this step. Log on to each node that hosts a PBS Professional daemon:
  - a. Start PBS Professional on that node. For example, on the server node:
 

```
# PBS_DATA_SERVICE_USER=<non-root user for the database> /etc/init.d/pbs start
```
  - b. Stop PBS Professional (in order to make the `PBS_HOME` directory persistent in the next step)
10. Make `PBS_HOME` (e.g. `/var/spool/pbs`) persistent:
 

See Cray's documentation on configuration sets and the configurator. Edit the Cray XC Ansible play named "Persistent Dirs". Modify the same configuration set that you created earlier in step 6.

Unlike CLE 5.2 and earlier, with CLE 6 and 7 `/var/spool` is **not persistent by default**. Any configuration changes are lost when nodes are rebooted, unless you use Cray's `cfgset` command to make particular directories persistent. You may want to make `/var/spool/pbs`, `/etc/pbs.conf`, and others persistent.
11. Start PBS Professional. At this point PBS Professional will be up and running, connected to ALPS, and ready to be configured. You can add MoMs to the server, etc. See ["Configuring PBS for Cray" on page 473 in the PBS Professional Administrator's Guide](#).
12. Optional step: PBS is shipped with a module file in `PBS_EXEC/etc/modulefile`. If your system uses modules, you can copy this module file to the appropriate location for your system configuration.

## 7.5.4 Installing PBS on CLE 5.2 and Older

### 7.5.4.1 Installation and Configuration Overview for CLE 5.2

1. If you will run the server, scheduler, and communication daemons on a non-Cray XC host, install the server there. Follow the steps in [section 7.5.4.3, "Installing PBS Server, Scheduler, and Communication Daemons on Non-Cray XC Host", on page 148](#)

The new version of the PBS Professional software is now installed, and you are ready to proceed with configuration.
2. If you will run the server, scheduler, and communication daemons on a service node, configure the service node to run the server, scheduler, and communication daemons. Follow the steps in [section 7.5.4.4, "Configuring Service Node to Run Server, Scheduler, and Communication", on page 149](#)
3. Configure the PBS server, scheduler, and communication daemons. Follow the steps in [section 7.5.4.5, "Configuring PBS Server, Scheduler, and Communication Daemons", on page 150](#)
4. Configure the PBS MoMs. Follow the steps in [section 7.5.4.6, "Configuring MoMs on the Cray XC", on page 151](#)
5. This step is optional. Configure startup for PBS Professional only if you need to change your startup configuration. Follow the steps in [section 7.5.4.7, "Configuring PBS Professional Startup and Shutdown", on page 152](#)

## 7.5.4.2 Installation Steps on CLE 5.2

1. Log in to the boot node.
2. If the `/rr/current/software/pbspro` directory is not present, create it.
3. Copy the PBS server RPM to the `/rr/current/software/pbspro` directory on the boot node. Make sure it is the only RPM file present in the directory.

4. Set the compute nodes to *batch*:

```
boot# xtprocadmin -k m batch
```

5. Install the PBS RPM by running the following command on the boot node. Adjust the value of `PBS_SERVER` to the name of the node where the scheduler/server/comm services will run. For a fresh install run the following:

```
boot# xtopview -d /rr/current/software/pbspro -m "Installing PBS Pro" -e "PBS_SERVER=sdb rpm -i /mnt/pbspro-server-*.rpm"
```

6. Check to make sure `PBS_SERVER` is set correctly in `/etc/pbs.conf` and edit if necessary. All PBS services should be disabled:

```
boot# xtopview -e "cat /etc/pbs.conf"
PBS_EXEC=/opt/pbs
PBS_SERVER=sdb
PBS_START_SERVER=0
PBS_START_SCHED=0
PBS_START_COMM=0
PBS_START_MOM=0
PBS_HOME=/var/spool/pbs
PBS_CORE_LIMIT=unlimited
PBS_SCP=/usr/bin/scp
```

7. Update `/etc/pbs.conf` for the login nodes: set the value of `PBS_START_MOM` to `1`:

```
boot# xtopview -c login -e "xtspec /etc/pbs.conf"
boot# xtopview -c login -e "vi /etc/pbs.conf"
***File /etc/pbs.conf was MODIFIED
boot# xtopview -c login -e "cat /etc/pbs.conf"
PBS_EXEC=/opt/pbs
PBS_SERVER=sdb
PBS_START_SERVER=0
PBS_START_SCHED=0
PBS_START_COMM=0
PBS_START_MOM=1
PBS_HOME=/var/spool/pbs
PBS_CORE_LIMIT=unlimited
PBS_SCP=/usr/bin/scp
```

8. Determine the NID of the node that will run the PBS server and scheduler. In this example, the SDB node will be used:

```
boot# ssh sdb cat /proc/cray_xt/nid
5
```

9. Use the value returned (in this example, the value is `5`) as the argument to the `-n` parameter and update the

`/etc/pbs.conf` settings for the PBS server. The values of `PBS_START_SERVER`, `PBS_START_SCHED`, and `PBS_START_COMM` should all be set to `1`:

```
boot# xtopview -n 5 -e "xtspec /etc/pbs.conf"
boot# xtopview -n 5 -e "vi /etc/pbs.conf"
***File /etc/pbs.conf was MODIFIED
boot# xtopview -n 5 -e "cat /etc/pbs.conf"
PBS_EXEC=/opt/pbs
PBS_SERVER=sdb
PBS_START_SERVER=1
PBS_START_SCHED=1
PBS_START_COMM=1
PBS_START_MOM=0
PBS_HOME=/var/spool/pbs
PBS_CORE_LIMIT=unlimited
PBS_SCP=/usr/bin/scp
```

- The PBS data service runs on the PBS server node. The processes must be owned by an account other than root (e.g. `pbsdata`, `postgres`, etc.). If you are performing these steps while upgrading a version of PBS prior to 17, `PBS_HOME` already exists on the PBS server node and the file `PBS_HOME/server_priv/db_user` is already populated with the name of the account. If this is a clean install, there are two options.

- Create a Linux user account named `pbsdata` on the PBS server host. That is the default account name used by the PBS data service:

```
boot# xtopview -n 5 -e "useradd -c 'PBS Pro Dataservice' -d /home/users/pbsdata -m pbsdata"
boot# ssh sdb
== Welcome to sdb ==
sdb# /etc/init.d/pbs start
```

OR

- Specify the data service account name when starting PBS for the first time. In this example, the SDB node will be running the PBS data service as the user `postgres`. The `postgres` account must already exist prior to starting PBS for the first time:

```
boot# ssh sdb
== Welcome to sdb ==
sdb# PBS_DATA_SERVICE_USER=postgres /etc/init.d/pbs start
```

- Start the PBS service on each execution host:

```
sdb# ssh nid00030 /etc/init.d/pbs start
```

- Enable `flatuid` on the server:

```
sdb# qmgr -c "set server flatuid = true"
```

- Tell the PBS server where to find the license server. Set the `pbs_license_info` server attribute via `qmgr`. For example:

```
sdb# qmgr -c "set server pbs_license_info = 6200@licenseserver"
```

- With licensing now configured, restart PBS on the SDB node:

```
sdb# /etc/init.d/pbs restart
```

- Working on the server node, configure the execution hosts. Create a node in PBS for each login/service node that will be running `pbs_mom`:

```
sdb# qmgr -c "create node nid00030"
```

The installation of PBS should now be complete. The PBS services should be up and running, and the system should be ready for jobs to be submitted.

### 7.5.4.3 Installing PBS Server, Scheduler, and Communication Daemons on Non-Cray XC Host

If you will run the PBS server, scheduler, and communication daemons on the Cray XC, skip this section. If you will run the PBS server, scheduler, and communication daemons on a non-Cray XC host, do the following:

1. Log on to the server host as root.
2. Unzip and untar the install package.
3. Change to the directory created when the package was untarred.
4. Run the INSTALL script: accept the default locations, and select the number *1* (server, scheduler, communication, execution and commands) option.
5. When asked if you wish to start PBS, answer *"no"*.
6. Make sure that MoM does not run on the server/scheduler/communication host. Modify `/etc/pbs.conf` by changing the line `"PBS_START_MOM=1"` to `"PBS_START_MOM=0"`.
7. Ensure that the `PBS_SERVER` entry in `pbs.conf` names the machine on which you plan to run the server, scheduler, and communication daemons and that `PBS_EXEC` is set to `/opt/pbs`.
8. Log in as root to the host on which the PBS server `pbs_server` will be running.

```
# ssh root@<server host>
```

9. At this point, you **must** start the PBS server with the `init.d/pbs` script. Do not start the server by any other means.

```
<host name> # /etc/init.d/pbs start
```

This will create or update the PBS working directories, `PBS_HOME`, in `/var/spool`.

---

### 7.5.4.4 Configuring Service Node to Run Server, Scheduler, and Communication

If you will run the PBS server, scheduler, and communication daemons on a non-Cray XC host, skip this section. If you will run the PBS server, scheduler, and communication daemons on a service node, configure the service node to run the PBS server, scheduler, and communication daemons:

1. Log into the boot node.
2. Determine the node ID, or NID, of the node on which you plan to run the PBS server, scheduler, and communication daemons.
3. Tailor the install to run the PBS server, scheduler, and communication daemons on the service node:

Use the `xtopview` command:

```
# xtopview -m "pbs.conf for Server" -n <NID>
```

Run the `xtspec` command to specialize the change to `/etc/pbs.conf` for the service node:

```
<nodeID>: # xtspec /etc/pbs.conf
```

4. Enable the server, scheduler, and communication daemons on the service node. Edit the `/etc/pbs.conf` file or verify that it contains the following lines:

```
PBS_START_SERVER=1
PBS_START_SCHED=1
PBS_START_COMM=1
```

5. Verify that the MoM start line in `/etc/pbs.conf` is:

```
PBS_START_MOM=0
```

6. Ensure that the `PBS_SERVER` entry in `pbs.conf` names the node on which you plan to run the server, scheduler, and communication daemons and that `PBS_EXEC` is set to `/opt/pbs`.
7. Exit from `xtopview`.

---

### 7.5.4.5 Configuring PBS Server, Scheduler, and Communication Daemons

1. If you are on a Cray XC node, load the pbs module:

```
<service node>: # module load pbs
```

2. Using the qmgr command, create the PBS execution node or nodes, which are the login nodes.

When creating a vnode to represent a login node, use the short name returned by the `hostname` command on the login node. For example, if `hostname` returns *HostA*, do the following:

```
Qmgr: create node HostA
```

If you create a vnode with a different name from the short name returned by `hostname`, the following happens:

- MoM creates a vnode whose name is the short name returned by `hostname`
- The vnode you created is not recognized by MoM, and is marked *stale*

3. The execution queue *workq* is automatically created during the normal PBS post-install procedure.
4. Make sure that `flatuid` is set to *True* in the PBS server configuration:

```
<server host>: # qmgr -c "list server"
```

If `flatuid` is not set to true, set it to true:

```
<server host>: # qmgr -c "set server flatuid=true"
```

5. To ensure that scheduling of jobs is initiated when PBS is restarted, set the server's `scheduling` attribute to true:

```
<server host>: # qmgr -c "set server scheduling=true"
```

6. Shut down the PBS daemons using the following command:

```
/etc/init.d/pbs stop
```



### 7.5.4.6 Configuring MoMs on the Cray XC

Configure each login node to run a PBS MoM:

1. As root, log in to the boot node:

```
# ssh root@boot
```

2. Tailor the install to run MoMs on the login nodes:

Use the `xtopview` command:

```
xtopview -m "pbs.conf for login" -c login
```

Run the `xtspec` command to specialize the change to `/etc/pbs.conf` for the login nodes:

```
class/login/: # xtspec /etc/pbs.conf
```

3. Enable MoM. Edit `/etc/pbs.conf` ensuring the following lines are present:

```
PBS_START_MOM=1
```

4. Make sure that the server, scheduler, and communication daemons do not start:

```
PBS_START_SERVER=0
```

```
PBS_START_SCHED=0
```

```
PBS_START_COMM=0
```

5. If you want each login node to run a `pbs_comm`, configure the `pbs_comms` and MoMs.

- a. Enable the communication daemon. Edit `pbs.conf` to set this variable:

```
PBS_START_COMM=1
```

- b. Tell each MoM about its `pbs_comm` by adding this line to its `pbs.conf`:

```
PBS_LEAF_ROUTERS=<host>[:<port>][,<host>[:>port]]
```

- c. Tell each `pbs_comm` where to find any already-configured `pbs_comm`, by adding this to its `pbs.conf`:

```
PBS_COMM_ROUTERS=<host>[,<host>]
```

For details, see [section 4.5.3, "Adding Communication Daemons", on page 50](#).

6. Ensure that the `PBS_SERVER` entry names the host or node on which you plan to run the server, scheduler, and communication daemons.
7. Ensure that `PBS_EXEC` is set to `/opt/pbs`.
8. Ensure that `/opt/pbs` is a symbolic link to the location you specified for the binaries.
9. Exit from `xtopview`.

On each login node, log in and perform the following steps:

1. Start the PBS MoM daemon with the `init.d/pbs` script. Do not use any other means to start `pbs_mom`. This will create or update the PBS working directories, `PBS_HOME`, in `/var/spool`.

```
Login:~ # /etc/init.d/pbs start
```

2. Stop the MoM daemon by running the `init.d/pbs` script:

```
Login:~ # /etc/init.d/pbs stop
```

3. Add the following directives in the `pbs_mom` configuration file, `PBS_HOME/mom_priv/config`:

For CLE:

```
$clienthost <server host or node>
$alps_client /usr/bin/apbasil (CLE 4.1 and earlier)
$alps_client /opt/cray/alps/default/bin/apbasil (CLE 5.0 and later)
```

The sample configuration is now complete.

4. HUP the MoM.

### 7.5.4.7 Configuring PBS Professional Startup and Shutdown

The PBS start/stop script is located in `/etc/init.d/pbs`. To use the `init.d` script and the `chkconfig` command to start up PBS Professional, execute the following steps:

1. If you will run the PBS server, scheduler, and communication daemons on a Cray XC service node, find the node ID (NID) of the node on which the PBS server, scheduler, and communication daemons will run. For example, on the `sdb` node:

```
grep sdb /etc/hosts
```

2. If you will run the PBS server, scheduler, and communication daemons on a Cray XC service node, enable PBS startup for the server, scheduler, and communication daemons on the service node. This example uses an NID of 3.

```
boot001: # xtopview -n 3
node/3:/ # chkconfig pbs on
node/3:/ # exit
```

3. Enable PBS startup for the MoM on each login node:

```
# xtopview -c login
class/login/: # chkconfig pbs on
class/login/: # exit
```

## 7.6 After Installing on the Cray XC

- We recommend that you set `node_fail_requeue` to 0 (zero) on any complex that manages Cray X\* series machines. See ["Prevent Jobs from Being Requeued on Cray XC" on page 483 in the PBS Professional Administrator's Guide](#).
- If you installed PBS on a Cray XC or a non-Cray XC headnode, make sure that the inventory hook, `PBS_alps_inventory_check`, is enabled. List it:

```
qmgr -c "list pbshook PBS_alps_inventory_check"
```

If the hook is not enabled:

```
qmgr -c "set pbshook PBS_alps_inventory_check enabled=true"
```

---

## 7.7 Upgrading on the Cray XC

### 7.7.1 Upgrading on CLE 6 and 7

#### 7.7.1.1 Overlay when Existing PBS Version is 13.0.40x or Lower

Prior to version 18, PBS used a script named `INSTALL` to install and configure the RPMs. The `INSTALL` script provided support for installing multiple versions simultaneously, using a symbolic link (`/opt/pbs/default`) to select the "active" version on the system. As of version 18, the `INSTALL` script is no longer supported. The new package allows you to install and upgrade PBS as you would any other RPM-based package.

1. Make sure that `PBS_HOME` is persistent.
2. Drain the system of running jobs. Retain queued jobs.
3. Remove existing vnodes. Prior to 18.x, PBS created one vnode per NUMA node, but from 18.x forward, PBS creates one vnode per compute node. We remove all older vnodes because they will be marked as stale and may be difficult to remove later.

```
# qmgr -c "delete node @default"
```

4. Shut down PBS on all the nodes where a PBS daemon is running:

```
# /etc/init.d/pbs stop
```

5. Install PBS: use the same configuration set that was used for the previous installation of PBS. Use the installation steps in [section 7.5.3.3, "Installation Steps for CLE 6 and 7", on page 144](#).
6. The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).
7. PBS 19.4.1 and later uses Python 3.6. Update all of your site-defined hooks (not the built-in hooks) to Python 3.6.
  - a. Export your hooks in ASCII format. For **each** hook:

```
# qmgr -c 'export hook <hook name> application/x-python default <hook name>.old2.7'
```
  - b. Update your hooks to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your new hook file differently; use something like "`<hook name>.new3.6`".
  - c. Import your hooks. For **each** hook:

```
# qmgr -c 'import hook <hook name> application/x-python default <hook name>.new3.6'
```

### 7.7.1.2 Overlay When Existing PBS Version is 18.2 or Higher

For the steps below, we will use the original repository, package collection, and recipes that were used for the existing version of PBS (the one that we are about to replace.)

1. Make sure that `PBS_HOME` is already persistent.
2. Drain the system of running jobs. Retain queued jobs.
3. Shut down PBS on all the nodes where a PBS daemon is running:

```
# /etc/init.d/pbs stop
```

4. Update the original repository. Follow Cray's instructions for updating a repository; refer to the section titled "Install Third-Party Software with a Custom Image" in the *Software Installation and Configuration Guide* in Cray document S-2559. Use the same repository that was used for the existing installed image. General syntax:

```
repo update -a "<path to new PBS server package> <repository name>
```

For example, if the original repository was called `my_repo`, the command might look something like this:

```
smw# repo update -a "./pbspro-server-18.2.0.20171109010636-0.x86_64.rpm" my_repo
```

5. Update the original package collection. Follow Cray's instructions in Cray document S-2559.

- a. Use the `pkgcoll update` command. General syntax:

```
pkgcoll update -p <new PBS server package> <package collection name>
```

For example, if the original package collection was called `my_collection`, the command might look something like this:

```
smw# pkgcoll update -p pbspro-server-18.2.0.20171109010636-0.x86_64 my_collection
```

- b. See what's in the package collection. Use the `pkgcoll` command. General syntax:

```
pkgcoll show <package collection name>
```

Following our example:

```
smw# pkgcoll show my_collection
```

```
my_collection:
```

```
name: my_collection
```

```
packages:
```

```
pbspro-server-18.2.0.20171018010818-0.x86_64
```

```
pbspro-server-18.2.0.20171109010636-0.x86_64
```

- c. Remove the older version of PBS. Use the package collection `update` option. General syntax:

```
pkgcoll update -P <PBS package> <package collection name>
```

Following our example:

```
smw# pkgcoll update -P pbspro-server-18.2.0.20171018010818-0.x86_64 my_collection
```

- d. Check to see that there is only one PBS RPM version. Use the package collection `show` command. General syntax:

```
pkgcoll show <package collection name>
```

This command should now show only one PBS RPM version.

6. Validate the recipe; follow Cray's instructions in Cray document S-2559. Even though changes have been made to the repository and the package collection, the original recipe should still validate at this point.

7. Build and push/package each type of image that you need such as server, login, etc.

Build and push/package the image using the original recipe; follow Cray's instructions in Cray document S-2559. Use a new image name in order to keep the original image untouched. General syntax:

```
image create -r <original recipe> <new image name>
```

```
image export <new image name>
```

For example, if the original recipe name was *my\_recipe* and the original image created was *my\_recipe\_image*, then the new image could be *my\_new\_image* and the command might look something like this:

```
smw# image create -r my_recipe my_new_image
```

```
smw# image export my_new_image
```

8. Assign the new boot image to each node where PBS Professional should be installed; follow Cray's instructions in Cray document S-2559. This includes all nodes running PBS Professional services such as `pbs_server`, `pbs_mom`, etc., and those requiring access to PBS commands, e.g. `qsub`, `qstat`, etc. Include the original configuration set that contains the persistent information. General syntax:

```
smw# cnode update -i /var/opt/cray/imps/boot_images/<image name>.cpio -c <configuration set name>
      <cname-of-node-to-update or group-to-update>
```

9. Reboot the system.
10. Log on to each node hosting a PBS Professional daemon, and start PBS.
11. At this point PBS Professional should be up and running, and connected to ALPS. Prior queued jobs should still be queued.
12. Turn on scheduling so PBS can resume normal operation.
13. The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).
14. PBS 19.4.1 and later uses Python 3.6. Update all of your site-defined hooks (not the built-in hooks) to Python 3.6.

- a. Export your hooks in ASCII format. For **each** hook:

```
# qmgr -c 'export hook <hook name> application/x-python default <hook name>.old2.7'
```

- b. Update your hooks to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your new hook file differently; use something like "`<hook name>.new3.6`".

- c. Import your hooks. For **each** hook:

```
# qmgr -c 'import hook <hook name> application/x-python default <hook name>.new3.6'
```

## 7.7.2 Upgrading on CLE 5.2

### 7.7.2.1 Overlay when Existing PBS Version is 13.0.40x or Lower

Prior to version 18, PBS used a script named `INSTALL` to install and configure the RPMs. The `INSTALL` script provided support for installing multiple versions simultaneously, using a symbolic link (`/opt/pbs/default`) to select the "active" version on the system. As of version 18, the `INSTALL` script is no longer supported. The new package allows you to install and upgrade PBS as you would any other RPM based package.

Due to the significant packaging changes, we recommend that you uninstall the old version of PBS prior to installing the new version. Uninstalling the old version after installing the new version will prevent PBS from starting automatically at boot. The `/etc/pbs.conf` file and contents of the `PBS_HOME` directories will not be affected when PBS is uninstalled.

The following instructions assume that the PBS server is running on the SDB node and that the PBS MoMs are running on the login nodes. Please adjust the commands according to your local configuration.

1. Drain the system of running jobs. Retain queued jobs.
2. Remove all existing vnodes:
 

```
sdb# qmgr -c "delete node @default"
```
3. Shut down PBS on all login nodes:
 

```
login# /etc/init.d/pbs stop
```
4. Shut down PBS on the server:
 

```
/etc/init.d/pbs stop
```
5. Remove specialization of the existing `/etc/pbs.conf` file:
 

```
boot# xtopview -e "xtunspec -N /etc/pbs.conf"
boot# xtopview -e "xtunspec -C /etc/pbs.conf"
```
6. There may be more than one version of PBS installed. Get the list of all currently installed PBS RPMs:
 

```
boot# xtopview -e "rpm -qa" | grep pbs
```
7. Remove each installed version of PBS:
 

```
boot# xtopview -e "rpm -e pbspro-<version>"
```
8. Remove the `/opt/pbs` directory:
 

```
xtopview -e "rm -rf /opt/pbs"
```
9. PBS is now completely uninstalled, but the `/etc/pbs.conf` and `PBS_HOME` directories remain. Leave these files in place and follow the steps to perform a fresh install of PBS in [section 7.5.4, "Installing PBS on CLE 5.2 and Older", on page 145](#).
10. The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).
11. PBS 19.4.1 and later uses Python 3.6. Update all of your site-defined hooks (not the built-in hooks) to Python 3.6.
  - a. Export your hooks in ASCII format. For **each** hook:
 

```
# qmgr -c 'export hook <hook name> application/x-python default <hook name>.old2.7'
```
  - b. Update your hooks to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your new hook file differently; use something like "`<hook name>.new3.6`".

- c. Import your hooks. For **each** hook:

```
# qmgr -c 'import hook <hook name> application/x-python default <hook name>.new3.6'
```

### 7.7.2.2 Overlay When Existing PBS Version is 18.2 or Higher

The following instructions assume that the PBS server is running on the SDB node and that the PBS MoMs are running on the login nodes. Please adjust the commands according to your local configuration.

1. Log in to the boot node.
2. If the `/rr/current/software/pbspro` directory is not present, create it.
3. Copy the new PBS server RPM to `/rr/current/software/pbspro` on the boot node.
4. Drain the system of running jobs. Retain queued jobs.
5. Shut down PBS on all login nodes:

```
login# /etc/init.d/pbs stop
```

6. Shut down PBS on the server:

```
sdb# /etc/init.d/pbs stop
```

7. On the boot node, run this command:

```
boot# xtopview -d /rr/current/software/pbspro -m "Upgrading PBS Pro" -e "rpm -U  
/mnt/pbspro-server-*.rpm"
```

8. Start the PBS server:

```
sdb# /etc/init.d/pbs start
```

9. The `preempt_order`, `preempt_prio`, `preempt_queue_prio`, and `preempt_sort` preemption settings were scheduler parameters in `$PBS_HOME/sched_priv/sched_config` in older versions of PBS. They are now scheduler attributes with the same names and formats. Make sure that you use `qmgr` to set the attributes as desired. See ["Scheduler Attributes" on page 300 of the PBS Professional Reference Guide](#).

10. On each machine that will host a MoM, start the MoM:

```
login# /etc/init.d/pbs start
```

11. PBS 19.4.1 and later uses Python 3.6. Update all of your site-defined hooks (not the built-in hooks) to Python 3.6.

- a. Export your hooks in ASCII format. For **each** hook:

```
# qmgr -c 'export hook <hook name> application/x-python default <hook name>.old2.7'
```

- b. Update your hooks to Python 3.6. See <https://docs.python.org/3.6/howto/pyporting.html>. Name your new hook file differently; use something like "`<hook name>.new3.6`".

- c. Import your hooks. For **each** hook:

```
# qmgr -c 'import hook <hook name> application/x-python default <hook name>.new3.6'
```

## 7.8 After Upgrading on Cray XC

### 7.8.1 Check `qstat` Wrappers and Scripts

You must check any `qstat` wrappers or scripts. Either replace any `mpp*` resources, or modify the wrapper or script to use values in the job's `Submit_arguments` attribute. This attribute contains the original job submission line.

## 7.8.2 Configure PBS to Support Cray XC

PBS provides tight integration with Cray XC. To take advantage of the tight integration, go to "[Configuring PBS for Cray](#)" on page 473 in the *PBS Professional Administrator's Guide*.

## 7.8.3 Enable Inventory Hook

If you upgraded PBS on a Cray XC or a non-Cray XC headnode, make sure that the inventory hook, `PBS_alps_inventory_check`, is enabled.

## 7.9 Cray XC References

*XC Series Software Installation and Configuration Guide S-2559*

<https://pubs.cray.com/content/S-2559/CLE%206.0.UP04/xctm-series-software-installation-and-configuration-guide-cle-60up04-s-2559-rev-b/install-third-party-software-with-a-custom-image-recipe>

*XC Series Configurator User Guide S-2560*

<https://pubs.cray.com/content/S-2560/CLE%206.0.UP04/xctm-series-configurator-user-guide-cle-60up04-s-2560/about-variable-names-in-the-configurator-and-configuration-worksheets>

*XC Series DVS Administration Guide S-0005*

<https://pubs.cray.com/content/S-0005/CLE%206.0.UP05/xctm-series-dvs-administration-guide>



# Starting & Stopping PBS on Linux

## 8.1 Automatic Start on Bootup

On installation, PBS is configured to start automatically. Under Linux, PBS starts on bootup using `init` or `systemd`. PBS uses `systemd` for automatic startup on platforms that support `systemctl`; for platforms that support only `init`, PBS uses `init` for automatic startup.

You specify which PBS daemons start on each host on bootup in that host's `/etc/pbs.conf`. The table below lists the parameters that control startup of daemons:

**Table 8-1: Daemon Start Parameters in `pbs.conf`**

Parameter	Description
<code>PBS_START_COMM</code>	Set this to <code>1</code> if a communication daemon is to run on this host.
<code>PBS_START_MOM</code>	Default is <code>0</code> . Set this to <code>1</code> if a MoM is to run on this host.
<code>PBS_START_SCHED</code>	Set this to <code>1</code> if a scheduler is to run on this host.
<code>PBS_START_SERVER</code>	Set this to <code>1</code> if server is to run on this host.

### 8.1.1 Shutting Down Host

When a host running PBS is shut down or rebooted, PBS is shut down via the `start/stop` script or `systemd`.

## 8.2 When to Restart PBS Daemons

- Restart PBS if you make changes to the hardware or a change occurs in the number of CPUs or amount of memory that is available to PBS. You should restart PBS by typing the following:  
`<path-to-script>/pbs restart`
- Restart PBS after making changes to the `/etc/hosts` file. See [section 2.1.3, "Name Resolution and Network Configuration", on page 8](#)
- Restart PBS after changing the name of the PBS service account
- Restart the scheduler(s) if you added a new custom resource to the `resources:` line in `sched_config`

## 8.3 Methods for Starting, Stopping, or Restarting PBS

The PBS daemons can be started by different types of methods. These types are not equivalent. You can use `init`, `systemd`, or the PBS command that starts the daemon.

The following table shows how to start, stop, restart, or status PBS on the local host:

**Table 8-2: Commands to Start, Stop, Restart, Status PBS**

Effect	init	systemd	Command
Start PBS	<code>/etc/init.d/pbs start</code> or <code>/etc/rc.d/init.d/pbs start</code>	<code>systemctl start pbs</code>	<code>pbs_server</code> <code>pbs_sched</code> <code>pbs_mom</code> <code>pbs_comm</code>
Stop PBS	<code>/etc/init.d/pbs stop</code> or <code>/etc/rc.d/init.d/pbs stop</code>	<code>systemctl stop pbs</code>	<code>qterm</code> (stops server, scheduler(s), MoM) <code>kill -INT &lt;PID of pbs_comm&gt;</code>
Status PBS	<code>/etc/init.d/pbs status</code> or <code>/etc/rc.d/init.d/pbs status</code>	<code>systemctl status pbs</code>	<code>qstat</code>
Restart PBS	<code>/etc/init.d/pbs restart</code> or <code>/etc/rc.d/init.d/pbs restart</code>	<code>systemctl restart pbs</code>	---

### 8.3.1 Using systemd

When you use `systemctl` to start PBS, it uses a PBS unit file. PBS supports `systemd` where it's available.

#### 8.3.1.1 Required Privilege

You must be root to run `systemctl`.

#### 8.3.1.2 Effect of `systemctl` on Jobs

When you use `systemctl` to start or stop PBS, any running jobs and subjobs are killed. When you use `systemd` to stop PBS, MoM kills her jobs and exits. When you use it to restart PBS, jobs are queued.

When you use `systemd` by typing "`systemctl stop pbs`", the following take place:

- The server gets a `qterm -t quick`
- MoM gets a `SIGTERM`: MoM terminates all running children and exits
- The communication daemon gets a `SIGTERM` and exits

### 8.3.1.3 Caveats for Using systemctl

PBS supports most `systemctl` options, including `start`, `stop`, `restart`, and `status`. However, PBS does not support the `reload` option.

`systemd` uses the settings in `pbs.conf` to determine which daemons to start and stop. If you specify in `pbs.conf` that a daemon should not start, `systemd` also will not stop it if it is running. For example, setting `PBS_START_MOM` to `0` effectively makes `systemd` ignore the MoM, and if you do the following steps, the `pbs_mom` process is **not** stopped:

1. Start `pbs_mom`
2. Set `PBS_START_MOM` to `0`
3. Run `systemd` with `stop` as the argument

### 8.3.2 Using init with PBS Start/Stop Script

When you use `init` to start PBS, `init` runs the PBS start/stop script. PBS supports `init` on all Linux systems.

The script starts, stops, or restarts PBS daemons on the local machine. It can also be used to report the PID of any PBS daemon on the local machine. The PBS start/stop script reads the `pbs.conf` file to determine which components should be started. The start/stop script runs automatically at boot time, starting PBS upon bootup. The start/stop script runs on and affects only the local host.

The PBS start/stop script is named `pbs`. To run it, you type the following:

```
<path to script>/pbs [start|stop|restart|status]
```

See ["pbs" on page 29 of the PBS Professional Reference Guide](#).

#### 8.3.2.1 Required Privilege

You must be root to run the start/stop script.

#### 8.3.2.2 Using Start/Stop Script to Check Status of Daemons

You can check whether or not each daemon is running by using the PBS start/stop script with the `status` option. To check the status of MoM, do the following on MoM's host:

```
<path to script>/pbs status
```

#### 8.3.2.3 Location of the PBS Start/Stop Script

If `/etc/init.d` exists

```
/etc/init.d/pbs
```

Else

```
/etc/rc.d/init.d/pbs
```

#### 8.3.2.4 Effect of Start/Stop Script on Jobs

When you use the PBS start/stop script to start or stop PBS, any running jobs and subjobs are killed on the host where you run the script. When you use the PBS start/stop script to stop PBS on the local host, MoM kills her jobs and exits. When you use it to restart PBS, jobs are requeued; note that there is a short but non-zero amount of time after MoM and the server are restarted, when jobs from MoM's previous session are visible via `qstat` but not running, before the server requeues them. If you stop one MoM for a multihost job, that job will probably be killed.

When you use the PBS start/stop script by typing "pbs stop", the following take place:

- The server gets a `qterm -t quick`
- MoM gets a SIGTERM: MoM terminates all running children and exits
- The communication daemon gets a SIGTERM and exits

### 8.3.2.5 Start/Stop Script Caveats

- The PBS start/stop script uses the settings in `pbs.conf` to determine which daemons to start and stop. If you specify in `pbs.conf` that a daemon should not start, the script also will not stop it if it is running. For example, setting `PBS_START_MOM` to `0` effectively makes the start/stop script or `systemd` ignore the MoM, and if you do the following steps, the `pbs_mom` process is **not** stopped:
  - a. Start `pbs_mom`
  - b. Set `PBS_START_MOM` to `0`
  - c. Run the PBS start/stop script or `systemd` with `stop` as the argument
- If you start PBS using the start/stop script, you cannot use `systemctl` to status PBS.

## 8.3.3 Using the `qterm` Command to Stop PBS

You use the `qterm` command to shut down your choice of the following PBS daemons:

- Primary server
- Secondary server
- Whichever default scheduler is running (primary or secondary)
- All MoMs

The `qterm` command does not shut down `pbs_comm`.

If you have failover configured, you can choose to shut down either or both servers, or you can shut down the primary and leave the secondary idle.

You can specify how running jobs and subjobs are treated during shutdown by specifying the type of shutdown. The type of shutdown performed by the `qterm` command defaults to "-t quick", which preserves running jobs and subjobs:

```
qterm -t quick
```

The following command shuts down the primary server, the scheduler(s), and all MoMs in the complex. If configured, the secondary server becomes active. Running jobs and subjobs continue to run:

```
qterm -s -m
```

The following command shuts down the primary server, the secondary server, the scheduler(s), and all MoMs in the complex. Running jobs and subjobs continue to run:

```
qterm -s -m -f
```

See "[qterm](#)" on page 234 of the [PBS Professional Reference Guide](#).

### 8.3.3.0.i `qterm` Caveats

- The `qterm` command does not stop the `pbs_comm` daemon. You must stop `pbs_comm` using the start/stop script, `systemd`, the service command, or the `kill` command.
- Shutting PBS down using the `qterm` command does not perform any of the other cleanup operations that are performed by the PBS start/stop script.

---

## 8.4 Starting, Stopping, and Restarting PBS Daemons

### 8.4.1 Daemon Execution Requirements

The server, scheduler(s), communication, and MoM processes must run with the real and effective UID of root.

### 8.4.2 Required Privilege

You must be root to run `pbs_server`, `pbs_mom`, `pbs_comm`, and `pbs_sched`.

### 8.4.3 Recommendation for Daemon Start Order

We recommend starting the communication daemon before starting the MoMs, but you can also start it after the MoMs and before the server.

We recommend starting MoMs before starting the server. This way, MoM will be ready to respond to the server's "are you there?" ping, preventing the server from attempting to contact a MoM that is still down. This will cut down on inter-daemon traffic, especially in larger complexes.

### 8.4.4 Creation of MoM Home Directory

When you run `systemctl` or the PBS start/stop script on an execution host, PBS creates MoM's home directory if it does not already exist.

### 8.4.5 Server: Starting, Stopping, Restarting

#### 8.4.5.1 Starting Server Without Failover

On the local host:

```
PBS_EXEC/sbin/pbs_server [options]
```

#### 8.4.5.2 Starting Servers With Failover

You can start the servers in any order. If you want to let running jobs and subjobs continue running, use the `pbs_server` command to start the servers. Starting via the start/stop script or `systemctl` kills running jobs and subjobs. If you want to start the primary server when the secondary server is the active server, you do not need to stop the secondary. When the primary server starts, it will inform the secondary that the primary is taking over and the secondary can become idle.

- On the primary host, start the primary server:  
`pbs_server`
- You can start the secondary server while it is the active server. On the secondary server host:  
`pbs_server -F -1`

The secondary server makes one attempt to contact the primary server, and becomes active immediately if it cannot.

If there is a network outage while the primary starts and the secondary cannot contact it, the secondary will assume the primary is still down, and remain active, resulting in two active servers. In this case, stop the secondary server, and restart it when the network is working:

```
qterm -F
pbs_server
```

### 8.4.5.3 Stopping Server Without Failover

To stop the server and leave running jobs and subjobs running:

```
qterm
```

#### 8.4.5.3.i Stopping Server via Signals

If you send the server a SIGTERM, the server does a quick shutdown, equivalent to receiving a `qterm -t quick`. See ["pbs\\_server" on page 109 of the PBS Professional Reference Guide](#) and ["qterm" on page 234 of the PBS Professional Reference Guide](#).

### 8.4.5.4 Stopping Servers With Failover

If you have failover configured, and want to stop the servers but allow running jobs and subjobs to continue running, use the `qterm` command. Both the start/stop script and `systemctl` kill running jobs and subjobs.

- To stop both servers when the primary server is active, and the secondary server is running and idle, do the following:

```
qterm -f
```

- To stop the primary server and leave the secondary server idle:

```
qterm -i
```

- To stop the secondary server only:

```
qterm -F
```

### 8.4.5.5 Restarting Server Without Failover

```
qterm -t quick
PBS_EXEC/sbin/pbs_server
```

### 8.4.5.6 Restarting Servers with Failover

#### 8.4.5.6.i Stopping Servers

If you have failover configured, and want to stop the servers but allow running jobs and subjobs to continue running, use the `qterm` command. Both the start/stop script and `systemctl` kill running jobs and subjobs.

- To stop both servers when the primary server is active, and the secondary server is running and idle, do the following:

```
qterm -f
```

- To stop the primary server and leave the secondary server idle:

```
qterm -i
```

- To stop the secondary server only:

```
qterm -F
```

### 8.4.5.6.ii Starting Servers

You can start the servers in any order. If you want to let running jobs and subjobs continue running, use the `pbs_server` command to start the servers. Starting via the start/stop script or `systemctl` kills running jobs and subjobs. If you want to start the primary server when the secondary server is the active server, you do not need to stop the secondary. When the primary server starts, it will inform the secondary that the primary is taking over and the secondary can become idle.

- On the primary host, restart the primary server:  
`pbs_server`
- To restart the secondary server while it is the active server:  
`pbs_server -F -1`

The secondary server makes one attempt to contact the primary server, and becomes active immediately if it cannot.

### 8.4.5.6.iii Network Outage

If there is a network outage while the primary starts and the secondary cannot contact it, the secondary will assume the primary is still down, and remain active, resulting in two active servers. In this case, stop the secondary server, and restart it when the network is working:

```
qterm -F
pbs_server
```

### 8.4.5.7 Restarting Server To Resume Previously-running Jobs

If, when the server was shut down, running jobs and subjobs were killed and requeued, then starting the server with the `-t hot` option puts those jobs back in the *Running* state first. See ["pbs\\_server" on page 109 of the PBS Professional Reference Guide](#) for details and the options to the `pbs_server` command.

## 8.4.6 Scheduler(s): Starting, Stopping, Restarting

### 8.4.6.1 Starting Default Scheduler

To start the default scheduler directly, do the following:

```
PBS_EXEC/sbin/pbs_sched [options]
```

### 8.4.6.2 Starting Multisched

To start a multisched, call `pbs_sched` and specify the name and port you already gave it:

```
pbs_sched -I <name of multisched> -S <same value as sched_port for this multisched>
```

For example:

```
pbs_sched -I multisched_1 -S 15050
```

When you start a multisched, you must specify its name.

See ["pbs\\_sched" on page 106 of the PBS Professional Reference Guide](#) for more information and a description of available options.

### 8.4.6.3 Stopping Scheduler or Multisched

1. Find the PID you want:

```
ps -ef | grep pbs_sched
```

---

For the default scheduler, you'll see "pbs\_sched", but for multischeds, you'll see "pbs\_sched -I <multisched name>".

2. Stop the scheduler or multisched:

```
kill <scheduler PID>
```

### 8.4.6.4 Stopping Scheduler(s) via Signals

You can stop a scheduler by sending it SIGTERM or SIGINT. These result in an orderly shutdown of the scheduler.

### 8.4.6.5 Restarting and Reinitializing Scheduler or Multisched

#### 8.4.6.5.i When to Restart or Reinitialize Scheduler or Multisched

- Restart the scheduler(s) after you change `pbs.conf`.
- HUP the scheduler(s) if you added any custom resources to the `resources:` line in `<sched_priv directory>/sched_config`.

#### 8.4.6.5.ii Restarting Scheduler or Multisched

1. Find the PID you want:

```
ps -ef | grep pbs_sched
```

For the default scheduler, you'll see "pbs\_sched", but for multischeds, you'll see "pbs\_sched -I <multisched name>".

2. Stop the scheduler or multisched:

```
kill <scheduler PID>
```

3. Start the scheduler or multisched:

- To start the default scheduler:

```
PBS_EXEC/sbin/pbs_sched [options]
```

- To start a multisched, call `pbs_sched` and specify the name and port you already gave it:

```
pbs_sched -I <name of multisched> -S <same value as sched_port for this multisched>
```

#### 8.4.6.5.iii Reinitializing Scheduler or Multisched

Find the PID you want:

```
ps -ef | grep pbs_sched
```

For the default scheduler, you'll see "pbs\_sched", but for multischeds, you'll see "pbs\_sched -I <multisched name>".

```
kill -HUP <scheduler PID>
```

## 8.4.7 MoMs: Starting, Stopping, Restarting

### 8.4.7.1 Starting MoM

You start the PBS MoM directly via the `pbs_mom` command. See ["pbs\\_mom" on page 72 of the PBS Professional Reference Guide](#).



## 8.4.7.2 Stopping MoM

### 8.4.7.2.i Stopping MoM via Signals

You can stop MoM using the following signals:

**Table 8-3: Signals Handled by MoM**

Signal	Effect
SIGTERM	If a MoM is killed with the signal SIGTERM, jobs are killed before MoM exits. Notification of the terminated jobs is not sent to the server until the MoM is restarted. Jobs will still appear to be in the "R" (running) state.
SIGINT	If a MoM is killed with this signal, jobs are not killed before the MoM exits. MoM exits after cleanly closing network connections.
SIGKILL	If a MoM is killed with this signal, jobs are not killed before the MoM exits.

### 8.4.7.2.ii Recommendation to Offline Vnodes Before Stopping MoM

We recommend that you offline vnodes before stopping the MoM. The server tries to keep continual contact with each MoM. If you offline the vnode before stopping the MoM, the server does not try to stay in contact with the MoM. This reduces network traffic.

## 8.4.7.3 Restarting and Reinitializing MoM

### 8.4.7.4 Whether to Restart or Reinitialize MoM

When you change configuration files on Linux, whether the MoM must be restarted or reinitialized depends on which MoM configuration file has been changed.

- If only the Version 1 MoM configuration file was changed, you only need to HUP the MoM.
- If you used the `pbs_mom -s insert` command to add to or change anything in the Version 2 MoM config file, you can HUP the MoM.
- If you used the `pbs_mom -s insert` command to remove anything from the Version 2 MoM config file, you must **restart** the MoM.

## 8.4.7.5 Restarting MoM

You can restart MoM with the following options:

**Table 8-4: MoM Restart Options**

Option	Effect on Jobs
<code>pbs_mom</code>	Job processes continue to run, but the jobs themselves are requeued.
<code>pbs_mom -r</code>	Running processes associated with jobs that were running before MoM was terminated are killed. Running jobs and subjobs are requeued or deleted. Do not use this option after a reboot, because process numbers will be incorrect and processes unrelated to jobs may be killed.
<code>pbs_mom -p</code>	Jobs which are running when MoM is terminated remain running. Do not use after reboot.

See "[pbs\\_mom](#)" on page 72 of the [PBS Professional Reference Guide](#).

### 8.4.7.5.i Preserving Existing Jobs When Restarting MoM

By default, when MoM is started, she allows running processes to continue to run, but tells the server to requeue her jobs. You can direct MoM to preserve running jobs and subjobs and to track them, by using the `-p` option to the `pbs_mom` command. If you have not just rebooted, you can preserve existing jobs:

1. Use the `ps` command to determine MoM's process ID. Note that `ps` arguments vary among Linux systems, thus `"-ef"` may need to be replaced by `"-aux"`.

```
ps -ef | grep pbs_mom
```

2. Terminate MoM using the `kill` command, with MoM's PID as an argument. The syntax will vary depending on your system:

```
kill -INT <MoM PID>
```

or

```
kill -s INT <MoM PID>
```

3. Restart MoM, allowing running jobs and subjobs to continue running through the restart. If your custom resource query script/program takes longer than the default ten seconds, you can change the alarm timeout via the `-a alarm` command line start option:

```
PBS_EXEC/sbin/pbs_mom -p [ -a timeout]
```

### 8.4.7.5.ii Caveats for Restarting MoM After a Reboot

Never restart `pbs_mom` with the `-p` or the `-r` option following a reboot of the host system.

When a Linux operating system is first booted, it begins to assign process IDs (PIDs) to processes as they are created. PID 1 is always assigned to the system "init" process. As new processes are created, they are either assigned the next PID in sequence or the first empty PID found, which depends on the operating system implementation. Generally, the session ID of a session is the PID of the top process in the session.

The PBS MoM keeps track of the session IDs of the jobs. If only MoM is restarted on a system, those session IDs/PIDs have not changed and apply to the correct processes.

If the entire system is rebooted, the assignment of PIDs by the system will start over. Therefore the PID which MoM thinks belongs to an earlier job will now belong to a different later process. If you restart MoM with `-p`, she will believe the jobs are still valid jobs and the PIDs belong to those jobs. When she kills the processes she believes to belong to one of her earlier jobs, she will now be killing the wrong processes, those created much later but with the same PID as she recorded for that earlier job.

### 8.4.7.5.iii Killing Existing Jobs When Restarting MoM

If you wish to kill all existing processes, use the `-r` option to `pbs_mom`.

To kill existing jobs, start MoM with the command line:

```
PBS_EXEC/sbin/pbs_mom -r
```

### 8.4.7.5.iv Starting MoM on the HPE MC990X, HPE Superdome Flex, or HPE 8600

For a cpusetted MC990X, Superdome Flex, or 8600, start MoM using the PBS `start/stop` script or `systemd`.

### 8.4.7.5.v Using Existing CPU and Memory for cpusets

By default, MoM removes existing cpusets when she starts. You can specify that MoM is to use existing CPU and memory allocations for cpusets by using the `-p` option to the `pbs_mom` command. This option also preserves running jobs and subjobs. See ["Options to pbs\\_mom" on page 73 of the PBS Professional Reference Guide](#).

Vnode definition files are not created when the `pbs_mom` command is used; use it only when you know that they are already up to date.

### 8.4.7.5.vi Effect of Stopping Sister MoM on Multihost Jobs

Stopping a sister MoM for a multi-vnode job may cause the job to be requeued if the primary MoM loses contact with the sister MoM.

### 8.4.7.6 Reinitializing MoM

1. Use the `ps` command to determine MoM's process ID. Note that `ps` arguments vary among Linux systems, thus `-ef` may need to be replaced by `-aux`.

```
ps -ef | grep pbs_mom
```

2. HUP MoM using the `kill` command, with MoM's PID as an argument:

```
kill -HUP <MoM PID>
```

See ["pbs\\_mom" on page 72 of the PBS Professional Reference Guide](#).

## 8.4.8 Comms: Starting, Stopping, Restarting

### 8.4.8.1 Starting Communication Daemon

To start the communication daemon directly, do the following on the local host:

```
PBS_EXEC/sbin/pbs_comm [-N] [ -r <other routers>] [-t <number of threads>]
```

See ["pbs\\_comm" on page 58 of the PBS Professional Reference Guide](#).

### 8.4.8.2 Stopping Communication Daemon via Signals

You can stop the communication daemon using a SIGTERM.

## 8.5 Impact of Stop-Restart on Running Linux Jobs

### 8.5.1 Whether to Use Script, Command, or Signal for Shutdown and Restart

Use the `qterm` command to shut the server down when running jobs and subjobs must be checkpointed before shutdown, allowed to run to completion before shutdown, or preserved through shutdown and restart. To preserve running jobs and subjobs, stop MoM using `KILL -INT` and use the `pbs_mom -p` command when restarting MoM.

When you use the PBS start/stop script or `systemd` to stop PBS, MoM kills her jobs and exits. When you use it to restart MoM, jobs are requeued.

### 8.5.2 Scenarios for Stopping Then Restarting Daemons

Choose one of the following recommended sequences, based on the desired impact on jobs, to stop and restart PBS:

- To allow running jobs and subjobs to continue to run:

Shutdown:

```
qterm -t quick -m -s
```

```
<path to start/stop script>/pbs stop (on communication-only host)
```

Restart:

```
pbs_server -t warm
```

```
pbs_mom -p
```

```
pbs_sched
```

```
pbs_comm (on server host)
```

```
<path to start/stop script>/pbs start (on communication-only host)
```

- To checkpoint and requeue checkpointable jobs, requeue rerunnable jobs, kill any non-rerunnable jobs, then restart and run jobs that were previously running:

Shutdown:

```
qterm -t immediate -m -s
```

```
<path to start/stop script>/pbs stop (on communication-only host)
```

Restart:

```
pbs_mom
```

```
pbs_server -t hot
```

```
pbs_sched
```

```
pbs_comm (on server host)
```

```
<path to start/stop script>/pbs start (on communication-only host)
```

- To checkpoint and requeue checkpointable jobs, requeue rerunnable jobs, kill any non-rerunnable jobs, then restart and run jobs without taking prior state into account:

Shutdown:

```
qterm -t immediate -m -s
```

```
<path to start/stop script>/pbs stop (on communication-only host)
```

Restart:

```
pbs_mom
```

```
pbs_server -t warm
```

```
pbs_sched
```

```
pbs_comm (on server host)
```

```
<path to start/stop script>/pbs start (on communication-only host)
```

# Starting & Stopping MoM on Windows

## 9.1 Automatic Start on Bootup

On Windows, the PBS MoM daemons are registered as system services, and are automatically started and stopped when the system boots and shuts down.

- The auto-startup of MoM is controlled by the PBS `pbs.conf` file and the *Services* dialog. You invoke this via *Settings->Control Panel->Administrative Tools->Services*. Make sure that in `pbs.conf` your setting for `PBS_START_MOM` is correct. If this is set to `0`, the service will fail to start up with the message, "incorrect environment".
- On Windows, sometimes MoM may fail to start automatically after the boot. We recommend that you change the startup mode from "*[Startup type: Automatic]*" to "*[Startup type: Automatic (Delayed Start)]*", which means "shortly after boot".

At the command prompt:

```
sc config <service name> start= delayed-auto
```

## 9.2 When to Restart PBS MoMs

Restart MoM:

- If you make changes to the hardware or a change occurs in the number of CPUs or amount of memory that is available to PBS
- After creating a Version 2 configuration file
- After changing the name of the PBS service account
- After changing the PBS service account to a non-domain administrator account
- After making changes to the `%WINDIR%\system32\drivers\etc\hosts` file

## 9.3 Starting, Stopping, and Restarting PBS

### 9.3.1 Required Privilege

To stop or start MoM, you must have Administrator privilege.

### 9.3.2 Recommendation for Service Start Order

We recommend starting the communication daemon before starting the MoMs, but you can also start it after the MoMs and before the server.

We recommend starting MoMs before starting the server. This way, MoM will be ready to respond to the server's "are you there?" ping, preventing the server from attempting to contact a MoM that is still down. This will cut down on inter-daemon traffic, especially in larger complexes.

### 9.3.3 Creation of MoM Home Directory

When you run `systemctl` or the PBS start/stop script on an execution host, PBS creates MoM's home directory if it does not already exist.

### 9.3.4 Windows-specific Service Options

The Windows MoM has the following Windows-only option:

`-N`

The service runs in standalone mode, not as a Windows service.

### 9.3.5 Configuring Startup Options to MoM

You can use the startup options to the `pbs_mom` command when starting the MoM.

The procedure to specify startup options to the MoM is as follows:

1. Go to the *Services* menu.
2. Select "*PBS\_MOM*". The MoM service dialog box comes up.
3. Enter the desired options in the "*Start parameters*" entry line. For example, to specify an alternate MoM configuration file, you might specify the following input:

On Windows systems:

```
-c "%Program Files (x86)\PBS\home\mom_priv\config2"
```

4. Click on "*Start*" to start the MoM service.

#### 9.3.5.1 Saving Startup Options

You can save your options for the future. If `PBS_EXEC` and `PBS_HOME` are set:

```
sc config pbs_mom binpath="%PBS_EXEC%\sbin\pbs_mom.exe -c "%PBS_HOME%\mom_priv\config2""
```

If you don't save your startup options, the Windows services dialog does not remember the "*Start parameters*" value when you close the dialog. You will have to specify the "*Start parameters*" value for each future restart.

### 9.3.6 MoMs: Starting, Stopping, Restarting

On Windows, you must restart MoM when any MoM configuration file has been changed.

#### 9.3.6.1 Starting MoM as a Service

On the local host:

```
net start pbs_mom
```

### 9.3.6.2 Starting MoM in Standalone Mode

On the local host:

```
pbs_mom -N <options>
```

### 9.3.6.3 Stopping MoMs

On the local host:

```
net stop pbs_mom
```

#### 9.3.6.3.i Effect of Stopping Sister MoM on Multihost Jobs

Stopping a sister MoM for a multi-vnode job may cause the job to be requeued if the primary MoM loses contact with the sister MoM.

#### 9.3.6.3.ii Recommendation: Offline Vnodes Before Stopping MoM

We recommend that you offline vnodes before stopping the MoM. The server tries to keep continual contact with each MoM. If you offline the vnode before stopping the MoM, the server does not try to stay in contact with the MoM. This reduces network traffic.

### 9.3.6.4 Restarting MoMs

You can restart MoM with the following options:

**Table 9-1: MoM Restart Options**

Option	Effect on Jobs
<code>pbs_mom</code>	Job processes will continue to run, but the jobs themselves are requeued.
<code>pbs_mom -p</code>	Jobs which were running when MoM terminated remain running.
<code>pbs_mom -r</code>	Processes associated with the job are killed. Running jobs and subjobs are returned to the server to be requeued or deleted. This option should not be used if the system has just been rebooted as the process numbers will be incorrect and a process not related to the job would be killed.

See [section 9.3.5, "Configuring Startup Options to MoM", on page 172](#).

On the local host:

```
Admin> net stop pbs_mom
Admin> net start pbs_mom
```

#### 9.3.6.4.i Preserving Existing Jobs When Restarting MoM

By default, when MoM is started, she allows running processes to continue to run, but tells the server to requeue her jobs. You can direct MoM to preserve running jobs and subjobs and to track them, by using the `-p` option to the `pbs_mom` command.

#### 9.3.6.4.ii Caveats for Preserving Existing Jobs When Restarting MoM

- If you restart a sister MoM for a multi-vnode job, the job may be killed because the primary MoM may lose contact with the sister MoM and requeue the job.
- Never use the `-p` option to `pbs_mom` after a reboot.

## 9.4 Stopping PBS Using the `qterm` Command

The `qterm` command is used to shut down, selectively or inclusively, the PBS server, scheduler(s), and MoMs. The `qterm` command does not shut down `pbs_comm`. If you have a failover server configured, then when the primary server is shut down, the secondary server becomes active unless you shut it down as well. The `qterm` command can be run at any PBS host.

You can specify how running jobs and subjobs are treated during shutdown by specifying the type of shutdown. The type of shutdown performed by the `qterm` command defaults to "-t quick", which preserves running jobs and subjobs:

```
qterm -t quick
```

The following command shuts down the primary server, the scheduler(s), and all MoMs in the complex. If configured, the secondary server becomes active. Running jobs and subjobs continue to run:

```
qterm -s -m
```

The following command shuts down the primary server, the secondary server, the scheduler(s), and all MoMs in the complex. Running jobs and subjobs continue to run:

```
qterm -s -m -f
```

See "[qterm](#)" on page 234 of the [PBS Professional Reference Guide](#).

### 9.4.0.0.i `qterm` Caveats

- The `qterm` command does not stop the `pbs_comm` service. You must stop `pbs_comm` using the start/stop script or the `kill` command.
- Shutting PBS down using the `qterm` command does not perform any of the other cleanup operations that are performed by the `net stop` command.

## 9.5 Impact of Stop-Restart on Running Windows Jobs

The methods you can use to shut down PBS, and which daemons are shut down, will affect running jobs and subjobs differently. You can leave jobs and subjobs running during shutdown.

The impact of a shutdown (and subsequent restart) on running jobs and subjobs depends on whether you use `net stop` or the `qterm` command to shut down PBS, and how `pbs_mom` is restarted.

You can use the `qterm` command to shut the server down.

Jobs are not killed when `pbs_mom` is stopped via `net stop`; whether they are killed depends on how MoM is restarted.

Use the `qterm` command to shut the server down when running jobs and subjobs must be checkpointed before shutdown, allowed to run to completion before shutdown, or preserved through shutdown and restart.

To preserve running jobs and subjobs, use the `-p` option to the `pbs_mom` command when restarting MoM.

### 9.5.1 Scenarios for Stopping Then Restarting Services

Choose one of the following recommended sequences, based on the desired impact on jobs, to stop and restart PBS.

The start/stop script is located in `/etc/init.d/pbs` or `/etc/rc.d/init.d/pbs`.

- To allow running jobs and subjobs to continue to run:



Shutdown:

```
qterm -t quick -m -s
<path to start/stop script>/pbs stop (on communication-only host)
```

Restart:

```
PBS_EXEC/sbin/pbs_server -t warm
pbs_mom -p
PBS_EXEC/sbin/pbs_sched
PBS_EXEC/sbin/pbs_comm (on server host)
<path to start/stop script>/pbs start (on communication-only host)
net start pbs_mom (with -p startup option set)
```

- To checkpoint and requeue checkpointable jobs, requeue rerunnable jobs, kill any non-rerunnable jobs, then restart and run jobs that were previously running:

```
qterm -t immediate -m -s
<path to start/stop script>/pbs stop (on communication-only host)
```

Restart:

```
net start pbs_mom
PBS_EXEC/sbin/pbs_server -t hot
PBS_EXEC/sbin/pbs_sched
PBS_EXEC/sbin/pbs_comm (on server host)
<path to start/stop script>/pbs start (on communication-only host)
```

- To checkpoint and requeue checkpointable jobs, requeue rerunnable jobs, kill any non-rerunnable jobs, then restart and run jobs without taking prior state into account:

Shutdown:

```
qterm -t immediate -m -s
<path to start/stop script>/pbs stop (on communication-only host)
```

Restart:

```
net start pbs_mom
PBS_EXEC/sbin/pbs_server -t warm
PBS_EXEC/sbin/pbs_sched
PBS_EXEC/sbin/pbs_comm (on server host)
<path to start/stop script>/pbs start (on communication-only host)
```



# Index

## A

account  
    installation [IG-13](#)  
    PBS service [IG-14](#)  
Active Directory [IG-13](#)  
Admin [IG-13](#)  
administrators [IG-13](#)  
authorization [IG-12](#)

## B

backup directory  
    overlay upgrade [IG-72](#), [IG-73](#), [IG-83](#), [IG-85](#), [IG-96](#)  
    Windows upgrade [IG-111](#), [IG-126](#), [IG-127](#)

## C

capmc [IG-139](#)  
CentOS [IG-23](#)  
CLE 6 and 7 [IG-139](#)  
client commands [IG-4](#)  
commands [IG-4](#)

## D

delegation [IG-13](#)  
DIS [IG-59](#)  
DNS [IG-38](#)  
Domain Admin Account [IG-13](#)  
Domain Admins [IG-13](#)  
Domain User Account [IG-13](#)  
Domain Users [IG-13](#)  
domains  
    mixed [IG-17](#)

## E

empty queue, node configurations  
    migration under Linux [IG-100](#), [IG-115](#), [IG-116](#),  
        [IG-130](#)  
Enterprise Admins [IG-13](#)

## F

failover  
    migration [IG-73](#), [IG-85](#), [IG-97](#), [IG-112](#), [IG-128](#)  
file  
    .rhosts [IG-12](#)  
    .shosts [IG-12](#)  
    hosts.equiv [IG-15](#), [IG-39](#)

pbs.conf [IG-43](#)  
services [IG-59](#)

## G

gethostbyaddr [IG-58](#)

## H

headnode [IG-21](#)

## I

IETF [IG-9](#), [IG-58](#)  
IMPS [IG-139](#)  
installation  
    Windows MoMs [IG-37](#)  
installation account [IG-13](#)

## M

migration upgrade [IG-65](#)  
    Linux [IG-93](#)  
    Windows [IG-109](#), [IG-125](#)  
mixed domains [IG-17](#)  
MoM [IG-4](#)  
moving jobs  
    migration upgrade under Linux [IG-107](#), [IG-123](#)

## N

network  
    ports [IG-58](#)  
    services [IG-58](#)  
NTFS [IG-41](#)

## O

output files [IG-12](#)  
overlay upgrade [IG-65](#)  
    backup directory [IG-72](#), [IG-73](#), [IG-83](#), [IG-85](#), [IG-96](#)  
    Linux [IG-70](#)

## P

PBS service account [IG-14](#)  
PBS\_BATCH\_SERVICE\_PORT [IG-59](#)  
PBS\_BATCH\_SERVICE\_PORT\_DIS [IG-59](#)  
PBS\_DATA\_SERVICE\_PORT [IG-59](#)  
PBS\_EXEC [IG-21](#), [IG-43](#)  
PBS\_EXEC/pbs\_sched\_config

## Index

---

overlay upgrade [IG-76](#), [IG-88](#), [IG-101](#), [IG-117](#),  
[IG-131](#)  
PBS\_HOME [IG-21](#), [IG-43](#)  
PBS\_LEAF\_NAME [IG-62](#)  
PBS\_MAIL\_HOST\_NAME [IG-62](#)  
PBS\_MANAGER\_SERVICE\_PORT [IG-59](#)  
pbs\_mom [IG-4](#)  
    starting during overlay [IG-78](#)  
PBS\_MOM\_HOST\_NAME [IG-62](#)  
PBS\_MOM\_SERVICE\_PORT [IG-59](#)  
PBS\_OUTPUT\_HOST\_NAME [IG-62](#)  
PBS\_PRIMARY [IG-62](#)  
pbs\_probe [IG-63](#)  
pbs\_sched [IG-3](#), [IG-4](#)  
PBS\_SCHEDULER\_SERVICE\_PORT [IG-59](#)  
PBS\_SECONDARY [IG-62](#)  
PBS\_SERVER [IG-62](#)  
pbs\_server [IG-3](#), [IG-4](#)  
PBS\_SERVER\_HOST\_NAME [IG-62](#)  
PBS\_START\_COMM [IG-159](#)  
PBS\_START\_MOM [IG-159](#)  
PBS\_START\_SCHED [IG-159](#)  
PBS\_START\_SERVER [IG-159](#)  
primary server [IG-62](#)

### Q

qalter [IG-16](#)  
qsub [IG-16](#)

### R

Red Hat Enterprise Linux [IG-23](#)  
Release Notes  
    upgrade recommendations [IG-65](#), [IG-93](#)

### S

scheduler [IG-4](#)  
Schema Admins [IG-14](#)  
scp [IG-12](#)  
secondary server [IG-62](#)  
secure copy [IG-12](#)  
server [IG-4](#)  
    primary [IG-62](#)  
    secondary [IG-62](#)  
service account  
    PBS [IG-14](#)  
ssh [IG-12](#)  
starting  
    MoM [IG-166](#)  
SuSE [IG-23](#)

### T

tar file

overlay upgrade [IG-73](#), [IG-85](#)

### U

upgrade  
    migration [IG-65](#)  
    migration under Linux [IG-93](#)  
    migration under Windows [IG-109](#), [IG-125](#)  
    overlay [IG-65](#)  
upgrading  
    Linux [IG-70](#)  
    Windows [IG-109](#), [IG-125](#)

### W

Windows [IG-15](#), [IG-17](#), [IG-23](#)

### X

X forwarding [IG-63](#)  
xauth [IG-63](#)



