

# Moab HPC Suite – Basic Edition

Installation and Configuration Guide 8.1.1

August 2015



© 2015 Adaptive Computing Enterprises, Inc. All rights reserved.

Distribution of this document for commercial purposes in either hard or soft copy form is strictly prohibited without prior written consent from Adaptive Computing Enterprises, Inc.

Adaptive Computing, Cluster Resources, Moab, Moab Workload Manager, Moab Viewpoint, Moab Cluster Manager, Moab Cluster Suite, Moab Grid Scheduler, Moab Grid Suite, Moab Access Portal, and other Adaptive Computing products are either registered trademarks or trademarks of Adaptive Computing Enterprises, Inc. The Adaptive Computing logo and the Cluster Resources logo are trademarks of Adaptive Computing Enterprises, Inc. All other company and product names may be trademarks of their respective companies.

Adaptive Computing Enterprises, Inc.

1712 S. East Bay Blvd., Suite 300

Provo, UT 84606

+1 (801) 717-3700

[www.adaptivecomputing.com](http://www.adaptivecomputing.com)



*Scan to open online help*

<b>Welcome To The Moab HPC Suite – Basic Edition Installation And Configuration Guide</b> .....	<b>1</b>
<b>Chapter 1 Requirements</b> .....	<b>3</b>
<b>Chapter 2 Manual Installation Guide</b> .....	<b>11</b>
Installation .....	12
Preparing For Manual Installation .....	12
Installing TORQUE .....	13
Installing Moab Workload Manager .....	20
Installing Moab Web Services .....	25
Additional Configuration .....	37
Configuring SSL In Tomcat .....	37
Setting Up OpenLDAP On CentOS 6 .....	37
Moab Workload Manager Configuration Options .....	44
Trusting Servers In Java .....	46
Upgrade .....	48
Preparing For Upgrade .....	48
Upgrading MongoDB .....	49
Upgrading TORQUE .....	50
Upgrading Moab Workload Manager .....	53
Upgrading MWS .....	55
Migrating The MAM Database From MySQL To PostgreSQL .....	59
<b>Chapter 3 RPM Installation Guide</b> .....	<b>61</b>
Installation .....	62
Installing Moab HPC Suite RPM .....	62
Configuring TORQUE .....	68
Configuring Moab Workload Manager .....	71
Configuring Moab Web Services .....	73
Additional Configuration .....	78
Configuring SSL In Tomcat .....	78
Setting Up OpenLDAP On CentOS 6 .....	78
Trusting Servers In Java .....	85
Upgrade .....	87
Upgrading Moab HPC Suite From 8.0 Or Later .....	87
Upgrading Moab HPC Suite From 7.2 .....	94
Upgrading From MongoDB 2.0 To 2.4.x .....	101
Migrating The MAM Database From MySQL To PostgreSQL .....	102
<b>Chapter 4 Troubleshooting</b> .....	<b>105</b>
<b>Chapter 5 Component Documentation</b> .....	<b>115</b>



# Welcome to the Moab HPC Suite – Basic Edition Installation and Configuration Guide

Welcome to the *Moab HPC Suite – Basic Edition Installation and Configuration Guide 8.1.1*, which will help you install or upgrade and configure your Moab HPC Suite. This guide includes detailed instructions for installing each component of the suite so that you can quickly get up and running.

This guide is intended for system administrators who are responsible for installing the Moab HPC Suite – Basic Edition. The Moab HPC Suite – Basic Edition 8.1.1 contains the following components:

- Moab Workload Manager 8.1.1
- Moab Web Services 8.1.1
- TORQUE 5.1.1

Before commencing the installation or upgrade, please see [Requirements on page 3](#) to verify your system conforms to minimum prerequisites.



# Chapter 1 Requirements



It is highly recommended that you *first* perform installations and upgrades in a *test environment*. Standard installation and upgrade procedures and use cases are tested prior to release. However, due to the wide range of possible configurations and customizations, it is important to exercise caution when deploying new versions of software into your production environments. This is especially true when the workload has vital bearing on your organization's day-to-day operations. We recommend that you test in an environment that mirrors your production environment's configuration, workflow and load as closely as possible. Please contact your Adaptive Computing account manager for suggestions and options for installing / upgrading to newer versions.

There are many different ways to install and configure Moab HPC Suite. Each environment has its own set of requirements and preferences. The following installation instructions are intended to help an administrator understand how each of the Moab HPC Suite components interact and how to install and configure each one. Two approaches have been documented: the "Manual installation" and the "RPM installation". Only one approach is required for installation; do not try to follow both sets of instructions on a single system.

Please note the following:

- Moab Accounting Manager is available only with the Moab HPC Enterprise Suite.
- Smaller environments may elect to consolidate the TORQUE Head Node with the Moab Head Node, including PBS Server in the list of components installed on the Moab Head Node.
- The Requirements section gives further clarification regarding what each component requires.
- Although Moab Workload Manager and Moab Accounting Manager may share the same database instance, it is not a requirement. Two database instances may be used, one for each component. See the Requirements section for more information about what databases are supported.
- Larger systems will require more dedicated resources for each component, in which case it may be necessary to move individual components from the Moab Head Node (i.e. databases, Moab Accounting Manager) to their own respective servers.
- The Message Queue component is fulfilled by ZeroMQ™. The libraries are provided with the components that use the message queue and are enabled via configuration; no special installation is necessary.

## Where to Start

1. Begin by reading the Requirements section below. Whether installing manually or with RPMs, it is important to be familiar with the hardware and software requirements.
2. Decide whether you will perform a manual installation or an RPM installation.
  - The manual installation provides advantages to administrators who want to pick and choose what components to install and administrators who want non-standard configure options.
  - The RPM installation provides advantages to administrators who want a fairly standard installation with TORQUE Resource Manager, Moab Workload Manager, and Moab Web Services installed on one server.

**i** RPM installation only supports installation for small to medium (with out large throughput) installation types and only on CentOS, RHEL, or Scientific Linux systems. For other system types or for other supported operating systems, refer to the manual installation instructions.

Then follow the appropriate installation instructions.

**i** Code samples have been provided for convenience. Some code samples provide sample passwords (i.e. "changeme!"). We strongly recommend that you do not use these passwords during installation, as using the documented passwords could introduce unnecessary security vulnerabilities into your system.

3. The "Additional Configuration" section in both the manual and the RPM installation instructions provide additional information and instructions for optional, but recommended configurations (i.e. Configuring SSL in Tomcat, etc.).
4. See [Troubleshooting on page 105](#) for assistance in addressing common problems during installation and configuration.
5. See [Component Documentation on page 115](#) for links to additional administrator and reference guides.

## Requirements

### Moab HPC Suite

#### *Hardware Requirements*

The following are the minimum hardware requirements for an average environment. Larger environments should consider allocating more resources and/or spreading components across multiple servers. Please consult the table below for recommendations.

Type	# of Compute Nodes	Jobs/week	Minimum Requirements (per Head Node distribution)	Recommended Requirements (targeting minimum number of servers)
Proof of Concept / Small Demo	50	<1k	<b>Moab+TORQUE Head Node:</b> <ul style="list-style-type: none"> <li>• 4 Intel/AMD x86-64 cores</li> <li>• At least 8 GB RAM</li> <li>• At least 100 GB dedicated disk space</li> </ul>	Same as minimum
Medium	500	<100k	<b>Moab+TORQUE Head Node:</b> <ul style="list-style-type: none"> <li>• 8 Intel/AMD x86-64 cores</li> <li>• At least 16 GB RAM</li> <li>• At least 512 GB dedicated disk space</li> </ul>	<b>Moab+TORQUE Head Node:</b> <ul style="list-style-type: none"> <li>• 16 Intel/AMD x86-64 cores</li> <li>• At least 32 GB RAM</li> <li>• At least 1 TB dedicated disk space</li> </ul>
Medium with High Throughput or Larger	>500	>100k	<b>Moab Head Node:</b> <ul style="list-style-type: none"> <li>• 8 Intel/AMD x86-64 cores</li> <li>• At least 16 GB RAM</li> <li>• At least 512 GB dedicated disk space</li> </ul> <b>TORQUE Head Node:</b> <ul style="list-style-type: none"> <li>• 8 Intel/AMD x86-64 cores</li> <li>• At least 16 GB RAM</li> <li>• At least 512 GB dedicated disk space</li> </ul>	We recommend separating components onto separate servers where possible (some components should not be separated; see Requirements below). Specific requirements around the intended configuration and use of Moab HPC Suite will help determine suite topology and resource allocation.

Please note the following:

- All requirements above (minimum and recommended) target a minimum number of management servers. Administrators are encouraged to separate the TORQUE and Moab head nodes where possible for better results, especially when High Throughput is enabled.
- Although many factors may have an impact on performance (network bandwidth, intended use and configuration, etc.), we consider High Throughput as something that makes a significant enough difference between minimum and recommended hardware requirements to merit mention in the table above.
- Moab and TORQUE are both multi-threaded and perform better with more processors.
- Regarding disk space, consideration should be given to requirements related to log files, log depth, number of jobs/nodes/reservations (more objects impact database journal size), average number of events generated (more events take more space), etc.

### *Software Requirements*

The installation documentation provides more details regarding how to install and configure the following software requirements. The information provided below is for your information only. No action is necessary.

Software requirements are listed per-component rather than suite-wide to make it easier for administrators who wish to install components on separate servers.

## **TORQUE**

### *Supported Operating Systems*

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12

**i** CentOS 5.9, RHEL 5.9, and Scientific Linux 5.9 are supported, largely to continue support for clusters where the compute nodes operating systems cannot be upgraded. We recommend that the TORQUE head node run on the supported operating systems listed above.

### *Software Requirements*

- libxml2-devel package (package name may vary)
- openssl-devel package (package name may vary)

- Tcl/Tk version 8 or later if you plan to build the GUI portion of TORQUE or use a Tcl based scheduler
- If your configuration uses cpusets, you must install libhwloc; the corresponding hwloc-devel package is also required. See **Linux Cpuset Support** in the *TORQUE Administrator Guide*

**i** libwloc 1.2 is required for TORQUE 5.1.x or 5.0.x; 1.1 is required for TORQUE 4.2.x.

If you intend to use TORQUE 5.1.1 with Moab Workload Manager, you must run Moab version 8.1.1 or 8.0.x. TORQUE 5.1.1 will not work with versions earlier than Moab 8.0.

If you build TORQUE from source (i.e. clone from github), the following additional software is required:

- gcc
- gcc-c++
- A posix compatible version of make
- libtool 1.5.22
- boost-devel 1.36.0

**i** Version 1.36.0 or newer is supported. RHEL 5, CentOS 5, and Scientific Linux 5 come packaged with an unsupported version. RHEL 6, CentOS 6, and Scientific Linux 6 come packaged with 1.41.0 and RHEL 7, CentOS 7, and Scientific Linux 7 come packaged with 1.53.0. If needed, use the `--with-boost-path=DIR` option to change the packaged boost version. See **Customizing the Install** in the *TORQUE Resource Manager Administrator Guide*.

## Moab Workload Manager

### *Supported Operating Systems*

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12

### *Software Requirements*

- [libcurl](http://curl.haxx.se/libcurl/) (<http://curl.haxx.se/libcurl/>)
- Perl 5.8.8 or later
- perl-CPAN (package name may vary)

- libxml2-devel (package name may vary)
- *(Optional)* Moab Accounting Manager 8.1
- *(Optional)* MySQL, PostgreSQL, or Oracle with ODBC driver; see **Database Configuration** in the *Moab Workload Manager Administrator Guide* for details

### *Supported Resource Managers*

- TORQUE 5.1
- SLURM

## **Moab Accounting Manager**

MAM is commonly installed on the same host as Moab Workload Manager; however, in some cases you might obtain better performance by installing them on separate hosts.

### *Supported Operating Systems*

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12

### *Software Requirements*

- gcc
- perl-suidperl
- httpd
- mod\_ssl
- rrdtool
- Moab® Workload Manager 8.1
- Perl modules

### *Depends On (not necessarily on the same server)*

MAM uses an RDBMS as a back end.

- PostgreSQL 7.2 or later

## **Moab Web Services**

MWS should be installed on the same host as Moab® Workload Manager.

### *Supported Operating Systems*

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12

### Software Requirements

- Moab® Workload Manager 8.1
- Apache Tomcat™
  - 6 for SUSE 11-based systems
  - 7 for Red Hat 6-Based, Red Hat 7-based or SUSE 12-based systems
- Oracle® Java® 7 Runtime Environment

**i** Oracle Java 7 Runtime Environment is the recommended Java environment, but Oracle Java 6 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run Moab Web Services.

- MongoDB® 2.4.x

### Depends On (not necessarily on the same server)

- OpenLDAP or PAM; see **Configuring Moab Web Services** in the *Moab Web Services Reference Guide*.

**i** MWS does not support LDAP *and* PAM authentication at the same time.



## Chapter 2 Manual Installation Guide

This chapter provides installation, configuration, and upgrading information using the manual install process.

### Related Topics

- [Requirements on page 3](#)
- [Preparing for Manual Installation on page 12](#)

# Installation

In this section:

- [Preparing for Manual Installation on page 12](#)
- [Installing TORQUE on page 13](#)
- [Installing Moab Workload Manager on page 20](#)
- [Installing Moab Web Services on page 25](#)

## Preparing for Manual Installation

The manual installation process of the Moab HPC Suite includes installing the separate components in the suite. This guide contains detailed instructions for installing each component.

**i** Many individual components have dependencies on other components (see [Requirements on page 3](#)). However, if you do not require a certain component (Moab Web Services, for example), you do not have to install it.

The install instructions for each component include information about system requirements and dependencies. Some include prerequisite instructions that you will need to complete before you begin the install. Please read this information carefully, and make sure you have installed all the dependencies and packages that are necessary in order to avoid errors during the Moab HPC Suite install process.

**i** Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Please note that the same commands will work for a non-root user with the `sudo` command.

## Enable Extra Packages for the Repository

Many individual components have dependencies that are found in the optional add-on repositories for the distribution. You must enable the respective repository for your distribution on all hosts upon which you install Adaptive Computing software components.

- Red Hat-based systems

```
[root]# yum install epel-release
```

**i** On RHEL systems, if your system is not registered to Red Hat Subscription Management, you will need to install the epel release rpm from the Fedora repo. The exact epel release version can vary over time, but the command should be similar to the following:

#### RHEL 6

```
[root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

#### RHEL 7

```
[root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
```

- SUSE-based systems

1. Verify that you have a licensed installation of SLES 11 or 12.
2. Download the SUSE Linux Enterprise 11 or 12 Software Development Kit e-Media Kit and add the ISO to the repository.
3. Add the `devel:languages:perl` repo for your SLES version to the your software repositories. For example, if you are using SLES 12, you might use the following commands:

```
[root]# zypper addrepo
http://download.opensuse.org/repositories/devel:languages:perl/SLE_
12/devel:languages:perl.repo
[root]# zypper refresh
```

## Install the Moab HPC Suite Software Components

To install the Moab HPC Suite, install the packages in the following order:

1. Install TORQUE. See [Installing TORQUE on page 13](#).
2. Install Moab Workload Manager. See [Installing Moab Workload Manager on page 20](#).
3. Install Moab Web Services. See [Installing Moab Web Services on page 25](#).

## Installing TORQUE

This topic contains instructions on how to install and start TORQUE.

In this topic:

- [Requirements on page 14](#)
- [Prerequisites on page 15](#)
- [Install Dependencies and Packages on page 16](#)
- [Install TORQUE on page 17](#)

## Requirements

### Supported Operating Systems

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12

**i** CentOS 5.9, RHEL 5.9 and Scientific Linux 5.9 are supported, largely to continue support for clusters where the compute nodes operating systems cannot be upgraded. We recommend that the TORQUE head node run on the supported operating systems listed above.

### Software Requirements

- libxml2-devel package (package name may vary)
- openssl-devel package (package name may vary)
- Tcl/Tk version 8 or later if you plan to build the GUI portion of TORQUE or use a Tcl based scheduler
- If your configuration uses cpusets, you must install libhwloc; the corresponding hwloc-devel package is also required. See **Linux Cpuset Support** in the *TORQUE Resource Manager Administrator Guide*.

**i** libhwloc 1.2 is required for TORQUE 5.1.x or 5.0.x; 1.1 is required for TORQUE 4.2.x.

If you intend to use TORQUE 5.1.1 with Moab Workload Manager, you must run Moab version 8.1.1 or 8.0.x. TORQUE 5.1.1 will not work with versions earlier than Moab 8.0.

If you build TORQUE from source (i.e. clone from github), the following additional software is required:

- gcc
- gcc-c++
- A posix compatible version of make
- libtool 1.5.22

- boost-devel 1.36.0

**i** Version 1.36.0 or newer is supported. Red Hat 5 systems come packaged with an unsupported version. Red Hat 6 systems come packaged with 1.41.0 and Red Hat 7 systems packaged with 1.53.0. If needed, use the `--with-boost-path=DIR` option to change the packaged boost version. See **Customizing the Install** in the *TORQUE Resource Manager Administrator Guide*.

## Prerequisites

### Open Necessary Ports

TORQUE requires certain ports to be open for essential communication:

- For client and pbs\_mom communication to pbs\_server, the default port is 15001.
- For pbs\_server communication to pbs\_mom, the default port is 15002.
- For pbs\_mom communication to pbs\_mom, the default port is 15003.

For more information on how to configure the ports that TORQUE uses for communication, see **Configuring Ports** in the *TORQUE Resource Manager Administrator Guide*.

If you have a firewall enabled, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the TORQUE server for client and MOM communication
-A INPUT -p tcp --dport 15001 -j ACCEPT

# Needed on the TORQUE MOM for server and MOM communication
-A INPUT -p tcp --dport 15002 -j ACCEPT
-A INPUT -p tcp --dport 15003 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=15001/tcp --permanent
[root]# firewall-cmd --add-port=15002/tcp --permanent
[root]# firewall-cmd --add-port=15003/tcp --permanent
[root]# firewall-cmd --reload
```

- SUSE 11-based and SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2

# Add the following ports to the FW_SERVICES_EXT_TCP parameter as required

# Needed on the TORQUE server for client and MOM communication
FW_SERVICES_EXT_TCP="15001"

# Needed on the TORQUE MOM for server and MOM communication
FW_SERVICES_EXT_TCP="15002 15003"

[root]# service SuSEfirewall2_setup restart
```

### Verify the hostname

Make sure your host (with the correct IP address) is in your `/etc/hosts` file. To verify that the hostname resolves correctly, make sure that `hostname` and `hostname -f` report the correct name for the host.

## Install Dependencies and Packages

Install the `libxml2-devel`, `openssl-devel`, and `boost-devel` packages.

- Red Hat 6-based and Red Hat 7-based systems

```
[root]# yum install libtool openssl-devel libxml2-devel boost-devel gcc gcc-c++
```

- SUSE 11-based and SUSE 12-based systems

```
[root]# zypper install libopenssl-devel libtool libxml2-devel boost-devel gcc
gcc-c++ make gmake
```

- Red Hat-5 based systems

```
[root]# yum install openssl-devel libtool-devel libxml2-devel gcc gcc-c++ wget
```

Use these instructions for installing `libtool`:

```
[root]# cd /tmp
[root]# wget http://ftpmirror.gnu.org/libtool/libtool-2.4.2.tar.gz
[root]# tar -xzvf libtool-2.4.2.tar.gz
[root]# cd libtool-2.4.2
[root]# ./configure --prefix=/usr
[root]# make
[root]# make install
```

**i** TORQUE requires Boost version 1.36.0 or greater. The boost-devel package provided with Red Hat 5-based systems is older than this requirement. A new option, `--with-boost-path` has been added to configure (see **Customizing the Install** in the *TORQUE Resource Manager Administrator Guide* for more information). This allows you to point TORQUE to a specific version of boost during make. One way to compile TORQUE without installing Boost is to simply download the Boost version you plan to use from: <http://www.boost.org/users/history/>. Next, untar Boost (you do not need to build it or install it). When you run TORQUE configure, use the `--with-boost-path` option pointed to the extracted Boost directory.

## Install TORQUE

Do the following:

1. Switch the user to root.

```
[user]$ su -
```

2. Download the latest 5.1 build from the [Adaptive Computing](#) website. It can also be downloaded via command line (github method or the tarball distribution).

- Clone the source from github.

**i** If git is not installed:

```
# Red Hat-based systems
[root]# yum install git

# SUSE-based systems
[root]# zypper install git
```

```
[root]# git clone https://github.com/adaptivecomputing/torque.git -b 5.1.1 5.1.1
[root]# cd 5.1.1
[root]# ./autogen.sh
```

- Get the tarball source distribution.
  - Red Hat-based systems

```
[root]# yum install wget
[root]# wget http://www.adaptivecomputing.com/download/torque/torque-5.1.1.tar.gz -O torque-5.1.1.tar.gz
[root]# tar -xzf torque-5.1.1.tar.gz
[root]# cd torque-5.1.1/
```

- SUSE-based systems

```
[root]# zypper install wget
[root]# wget http://www.adaptivecomputing.com/download/torque/torque-
5.1.1.tar.gz -O torque-5.1.1.tar.gz
[root]# tar -xzvf torque-5.1.1.tar.gz
[root]# cd torque-5.1.1/
```

3. Run each of the following commands in order.

```
[root]# ./configure
[root]# make
[root]# make install
```

For information on what options are available to customize the `./configure` command, see **Customizing the Install** in the *TORQUE Resource Manager Administrator Guide*.

4. Configure the `trqauthd` daemon to start automatically at system boot.

- Red Hat 6-based systems

```
[root]# cp contrib/init.d/trqauthd /etc/init.d/
[root]# chkconfig --add trqauthd
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
[root]# service trqauthd start
```

- SUSE 11-based systems

```
[root]# cp contrib/init.d/suse.trqauthd /etc/init.d/trqauthd
[root]# chkconfig --add trqauthd
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
[root]# service trqauthd start
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# cp contrib/systemd/trqauthd.service /usr/lib/systemd/system/
[root]# systemctl enable trqauthd.service
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
[root]# systemctl start trqauthd.service
```

5. Verify that the `/var/spool/torque/server_name` file exists and contains the correct name of the server.

```
[root]# echo <pbs_server's_hostname> > /var/spool/torque/server_name
```

6. By default, TORQUE installs all binary files to `/usr/local/bin` and `/usr/local/sbin`. Make sure the path environment variable includes these directories for both the installation user and the root user.

```
[root]# export PATH=/usr/local/bin:/usr/local/sbin:$PATH
```

7. Initialize `serverdb` by executing the `torque.setup` script.

```
[root]# ./torque.setup root
```

8. Add nodes to the `/var/spool/torque/server_priv/nodes` file. For information on syntax and options for specifying compute nodes, see **Specifying Compute Nodes** in the *TORQUE Resource Manager Administrator Guide*.
9. Configure the MOMs if necessary. See **Configuring on Compute Nodes** in the *TORQUE Resource Manager Administrator Guide*.

**i** The `make packages` command can be used to create self-extracting packages that can be copied and executed on your nodes. For information on creating packages and deploying them, see **Compute Nodes** in the *TORQUE Resource Manager Administrator Guide*.

10. On the TORQUE Server, configure `pbs_server` to start automatically at system boot, and then start the daemon.

- Red Hat 6-based systems

```
[root]# cp contrib/init.d/pbs_server /etc/init.d
[root]# chkconfig --add pbs_server
[root]# service pbs_server restart
```

- SUSE 11-based systems

```
[root]# cp contrib/init.d/suse.pbs_server /etc/init.d/pbs_server
[root]# chkconfig --add pbs_server
[root]# service pbs_server restart
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# qterm
[root]# cp contrib/systemd/pbs_server.service /usr/lib/systemd/system/
[root]# systemctl enable pbs_server.service
[root]# systemctl start pbs_server.service
```

11. Configure `pbs_mom` to start automatically at system boot on each compute node, and then start the daemon.

There are several methods to get the following `inti.d` scripts on to each node. The following instructions assume the entire contents of `contrib/init.d` in the TORQUE git repository or source tarball are copied(`scp`)/cloned to the compute node.

**i** These options can be added to the self-extracting packages.

On the TORQUE MOM, do the following:

- Red Hat 6-based systems

```
[root]# cp contrib/init.d/pbs_mom /etc/init.d
[root]# chkconfig --add pbs_mom
[root]# service pbs_mom start
```

- SUSE 11-based systems

```
[root]# cp contrib/init.d/suse.pbs_mom /etc/init.d/pbs_mom
[root]# chkconfig --add pbs_mom
[root]# service pbs_mom start
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# cp contrib/systemd/pbs_mom.service /usr/lib/systemd/system/
[root]# systemctl enable pbs_mom.service
[root]# systemctl start pbs_mom.service
```

#### Related Topics

[Preparing for Manual Installation on page 12](#)

[Installing Moab Workload Manager on page 20](#)

## Installing Moab Workload Manager

This topic contains instructions on how to install and start Moab Workload Manager (Moab).

In this topic:

- [Open Necessary Ports on page 20](#)
- [Install Dependencies and Packages on page 21](#)
- [\(Optional\) Building a Custom RPM on page 22](#)
- [Install Moab Workload Manager on page 22](#)

### Open Necessary Ports

Moab Workload Manager uses a configurable server port (default 42559) for client-server communication. If you intend to run client commands on a host other than the Moab Head Node, or if you will be using Moab in a grid, and if you have a firewall enabled, then you will need to configure the firewall to allow the server port.

Do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the Moab server for off-host client communication
-A INPUT -p tcp --dport 42559 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
firewall-cmd --add-port=42559/tcp --permanent
firewall-cmd --reload
```

- SUSE 11-based and SUSE 12-based systems

```
[root]# vi /etc/sysconfig/SuSEfirewall2

# Add the following ports to the FW_SERVICES_EXT_TCP parameter as required

# Needed on the Moab server for off-host client communication
FW_SERVICES_EXT_TCP="42559"

[root]# service SuSEfirewall2_setup restart
```

## Install Dependencies and Packages

Use the following commands to install the required Moab Workload Manager dependencies and packages.

- Red Hat-based systems

```
[root]# yum update
[root]# yum install make libcurl perl-CPAN libxml2-devel gcc
```

- SUSE-based systems

```
[root]# zypper update
[root]# zypper install make curl libxml2-devel
```

### *Installing TORQUE on a Separate Server*

If you are using TORQUE and are installing the TORQUE server on a separate TORQUE Head Node, you will need to install TORQUE on the Moab Head Node as well in order for Moab to interact with TORQUE. Follow the instructions in [Installing TORQUE on page 13](#) with these exceptions:

- Use the configure options `--disable-server`, `--disable-mom` and `--disable-gui`
- No need to create self-extracting packages with `make packages`
- Omit the step initializing the serverdb and all of the steps thereafter

## (Optional) Building a Custom RPM

Do the following:

1. Install `rpm-build`.

```
[root]# yum install rpm-build
```

2. Download the latest Moab build (`moab-<version>-<OS>.tar.gz`) from the [Adaptive Computing](#) website.

**i** The variable marked `<version>` is the desired version of the suite; for example, `8.1.0-2015010514-ed0c40a` would be Moab 8.1.0 revision 2015010514 at changeset `ed0c40a`. The variable marked `<OS>` indicates the OS for which the build was designed.

3. Untar the downloaded package.
4. Change directories into the untarred directory.
5. Edit the `./moab.spec` file for RPM customization.
6. Run `./rpm-build`.
7. Locate the custom RPM in `rpm/RPMS/x86_64`.

## Install Moab Workload Manager

1. Download the latest Moab build (`moab-<version>-<OS>.tar.gz`) from the [Adaptive Computing](#) website.

**i** The variable marked `<version>` is the desired version of the suite; for example, `8.1.0-2015010514-ed0c40a` would be Moab 8.1.0 revision 2015010514 at changeset `ed0c40a`. The variable marked `<OS>` indicates the OS for which the build was designed.

2. As the root user, run each of the following commands in order.

```
[root]# tar xzvf moab-<version>-<OS>.tar.gz
[root]# cd moab-<version>-<OS>
```

3. Configure Moab. For a complete list of `./configure` options, use `./configure --help` or refer to [Moab Workload Manager Configuration Options on page 44](#) for a list of commonly used options.

It is strongly recommended that you configure Moab with the `--with-init` and `--with-profile` options. The `--with-profile` option makes it easier to execute Moab commands. The `--with-init` option allows Moab to automatically start at OS startup.

```
[root]# ./configure <options>
```

4. *ONLY* if you are using green computing, *or* if you are using a resource manager other than TORQUE.

Run the `make perldeps` command to install the necessary perl modules using CPAN. When first running CPAN, you will be asked for configuration information. It is recommended that you choose an automatic configuration. You will be prompted to provide input during module installation; running the `make perldeps` command with a script is not recommended.

```
[root]# make perldeps
```

5. Install Moab.

```
[root]# make install
```

6. *ONLY* if you are installing on SLES 11.

Copy the appropriate `init.d` file, set the permissions on it, and configure Moab to start automatically at system boot.

```
# SLES 11
[root]# cp OS/SLES/etc/init.d/moab /etc/init.d/moab

[root]# chmod 755 /etc/init.d/moab
[root]# chkconfig --add moab
```

7. Modify the Moab configuration file.

```
[root]# vi /opt/moab/etc/moab.cfg
```

Do the following:

- a. Verify that **SUBMITCMD** is set up for your TORQUE resource manager and that it points to a valid `qsub` executable. For example:

```
RMCFG[torque] SUBMITCMD=/usr/local/bin/qsub
```

If you use a SLURM resource manager, see **Moab-SLURM Integration Guide** in the *Moab Workload Manager Administrator Guide* for configuration information. If you use a NATIVE resource manager, see **Managing Resources Directly with the Native Interface** in the *Moab Workload Manager Administrator Guide* for configuration information.

- b. *ONLY* if you are using Moab Web Services, add `tomcat` to the list of administrator **USERS**. For example:

```
ADMINCFG[1] USERS=root,tomcat
```

8. If you configured with the `./configure --with-profile` option, source the following file to add the Moab executable directories to your current shell `$PATH` environment.

```
[root]# . /etc/profile.d/moab.sh
```

9. Copy your license file into the same directory as `moab.cfg` (`/opt/moab/etc/` by default). For example:

```
[root]# cp moab.lic $MOABHOMEDIR/etc/moab.lic
```

To verify the current status of your license, use `moab --about`.

Moab checks the status of the license every day just after midnight. At 60 and 45 days before, and daily from 30 days before license expiration to and including the license expiration date, Moab sends an e-mail to all level 1 administrators informing them of the pending Moab license expiration. A log record is also made of the upcoming expiration event. For the notifications to occur correctly, you must enable administrator email notification (see **Notifying Administrators of Failures** in the *Moab Workload Manager Administrator Guide*) and `moab.cfg` must contain email addresses for level 1 administrators. For example:

```
ADMINCFG [1] USERS=u1,u2,u3[,...]
USERCFG [u1] EMAILADDRESS=u1@company.com
USERCFG [u2] EMAILADDRESS=u2@company.com
USERCFG [u3] EMAILADDRESS=u3@company.com
MAILPROGRAM DEFAULT
```

 Moab will not run without a license. For information about obtaining a trial license, please contact [Adaptive Computing](#).

10. Start Moab (assumes Moab configured with the `--with-int` option).

- Red Hat 6-based or SUSE 11-based systems

```
[root]# chkconfig moab on
[root]# service moab start
```

- Red Hat 7-based or SUSE 12-based systems

```
[root]# systemctl start moab.service
```

11. Submit a sleep job as a non-root user (adaptive is used in this example) and verify the job is running.

```
[root]# su - adaptive
[adaptive]$ echo sleep 150 | msub
[adaptive]$ showq
[adaptive]$ exit
```

## Related Topics

[Preparing for Manual Installation on page 12](#)

[Installing TORQUE on page 13](#)

## Installing Moab Web Services

This topic contains instructions on how to install Moab Web Services (MWS).

In this topic:

- [Open Necessary Ports on page 25](#)
- [Install Dependencies and Packages on page 26](#)
- [Install Moab Web Services on page 29](#)

### Open Necessary Ports

Moab Web Services requires certain ports to be open for essential communication. For communication with the tomcat web server, the default port is 8080. For communication with the Mongo database, the default port is 27017.

If you have a firewall enabled, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the MWS server for communication with the tomcat web server
-A INPUT -p tcp --dport 8080 -j ACCEPT

# Needed on the Mongo server if installed on a separate host from MWS
-A INPUT -p tcp --dport 27017 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=8080/tcp --permanent
[root]# firewall-cmd --add-port=27017/tcp --permanent
[root]# firewall-cmd --reload
```

- SUSE 11-based and SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2

# Add the following ports to the FW_SERVICES_EXT_TCP parameter as required

# Needed on the MWS server for communication with the tomcat web server
FW_SERVICES_EXT_TCP="8080"

# Needed on the Mongo server if installed on a separate host from MWS
FW_SERVICES_EXT_TCP="27017"

[root]# service SuSEfirewall2_setup restart
```

## Install Dependencies and Packages

Use the following commands to install the required Moab Web Services dependencies and packages.

### Install Java

Install the Linux x64 RPM version of Oracle® Java® 7 Runtime Environment.

**i** Oracle Java 7 Runtime Environment is the recommended Java environment, but Oracle Java 6 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run Moab Web Services.

**!** The Oracle® Java® download page has moved and requires a web-enabled workstation to accept the license agreement and download the software.

Do the following:

1. Download Java 7 on a web-enabled workstation.
  - a. Open a web browser and connect to the [Java downloads page](http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html) (<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>).
  - b. Select the radio button to accept the license agreement.
  - c. Click the download link for the Linux x64 RPM file.
2. Copy the Java 7 RPM to the MWS server using `scp`, `rsync`, or any similar network copy utility.
3. On the MWS server, run the following to install Java 7:

```
[root]# rpm -Uh <RPMfilename>
```

### Install Tomcat

- Red Hat 6-based and Red Hat 7-based systems

```
[root]# yum install tomcat
```

- SUSE 11-based systems

```
[root]# zypper ar --refresh -r
http://download.opensuse.org/evergreen/11.4/openSUSE:Evergreen:11.4.repo
[root]# zypper in tomcat6
[root]# zypper mr -d openSUSE_Evergreen_11.4
```

- SUSE-12 based systems

```
[root]# zypper install tomcat
```

## Install MongoDB

To install and enable MongoDB, do the following:

### 1. Install MongoDB.

- Red Hat 6-based and Red Hat 7-based systems

```
[root]# cat > /etc/yum.repos.d/mongodb.repo <<End-of-file
[mongodb]
name=MongoDB Repository
baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86_64
gpgcheck=0
enabled=1
exclude=mongodb-org mongodb-org-server
End-of-file
[root]# yum install mongo-10gen-server
```

- SUSE 11-based systems

```
[root]# zypper ar --refresh -r
http://download.opensuse.org/repositories/server:/database/SLE_11_
SP3/server:database.repo
[root]# zypper install mongodb
```

- SUSE-12 based systems

```
[root]# zypper ar --refresh -r
http://download.opensuse.org/repositories/server:/database/SLE_
12/server:database.repo
[root]# zypper install mongodb
```

### 2. Start MongoDB

**i** There may be a short delay (approximately three minutes) for Mongo to start the first time.

- Red Hat 6-based systems

```
[root]# chkconfig mongod on
[root]# service mongod start
```

- Red Hat 7-based systems

```
[root]# cat > /usr/lib/systemd/system/mongodb.service <<End-of-file
[Unit]
Description=High-performance, schema-free document-oriented database
After=syslog.target network.target

[Service]
Type=forking
User=mongod
Group=mongod
Environment=CONFIG=/etc/mongod.conf
Environment=OPTIONS=
EnvironmentFile=-/etc/sysconfig/mongod
ExecStart=/usr/bin/mongod -f \${CONFIG} \${OPTIONS}
PrivateTmp=true
LimitNOFILE=65536
TimeoutStartSec=180
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
End-of-file
[root]# rm -f /etc/init.d/mongod
[root]# systemctl enable mongodb.service
[root]# systemctl start mongodb.service
```

- SUSE 11-based systems

```
[root]# chkconfig mongodb on
[root]# service mongodb start
```

- SUSE 12-based systems

```
[root]# systemctl enable mongodb.service
[root]# systemctl start mongodb.service
```

3. Prepare the MongoDB database by doing the following:

- a. Add the required MongoDB users.

**i** The passwords used below (`secret1`, `secret2`, and `secret3`) are examples. Choose your own passwords for these users.

```
[root]# mongo
> use admin;
> db.addUser("admin_user", "secret1");
> db.auth("admin_user", "secret1");

> use moab;
> db.addUser("moab_user", "secret2");
> db.addUser("mws_user", "secret3", true);

> use mws;
> db.addUser("mws_user", "secret3");
> exit
```

**i** Because the `admin_user` has read and write rights to the `admin` database, it also has read and write rights to all other databases. See [Control Access to MongoDB Instances with Authentication](http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication) (at <http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication>) for more information.

d. Enable authentication in MongoDB.

- Red Hat 6-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# service mongod restart
```

- Red Hat 7-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# systemctl restart mongod.service
```

- SUSE 11-based and SUSE 12-based systems

MongoDB authentication is enabled (`auth = true`) by default. No further action is needed.

## Install Moab Web Services

**i** You must deploy Moab Web Services on the same server as Moab Workload Manager.

1. Verify Moab is installed and configured as desired. See [Installing Moab Workload Manager on page 20](#).
2. Start Moab.

- Red Hat 6-based and SUSE 11-based systems

```
[root]# service moab start
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# systemctl start moab.service
```

3. Create the MWS home directory and subdirectories. See **Configuration** in the *Moab Web Services Reference Guide*.

**i** The default location for the MWS home directory is `/opt/mws`. These instructions assume the default location.

Here is a sample script for this setup:

```
[root]# mkdir -p \
/opt/mws/etc/mws.d \
/opt/mws/hooks \
/opt/mws/log \
/opt/mws/plugins \
/opt/mws/spool/hooks \
/opt/mws/utils
[root]# chown -R tomcat:tomcat /opt/mws
[root]# chmod -R 555 /opt/mws
[root]# chmod u+w \
/opt/mws/log \
/opt/mws/plugins \
/opt/mws/spool \
/opt/mws/spool/hooks \
/opt/mws/utils
```

4. Download the latest MWS build (`mws-<version>.tar.gz`) from the [Adaptive Computing](#) website.

**i** The variable marked `<version>` is the desired version of the suite; for example, 8.1.1.

5. Extract the contents of the MWS download tarball into a temporary directory. For example:

```
[root]# mkdir /tmp/mws-install
[root]# cd /tmp/mws-install
[root]# tar xvzf $HOME/Downloads/mws-8.1.1.tar.gz
```

6. Copy the extracted utility files to the utility directory created above and give the tomcat user ownership of the directory.

```
[root]# cd /tmp/mws-install/mws-8.1.1/utils
[root]# cp * /opt/mws/utils
[root]# chown tomcat:tomcat /opt/mws/utils/*
```

7. Connect Moab to MongoDB.

On the Moab head node, connect Moab to MongoDB.

**i** The `USEDATABASE` parameter is unrelated to the MongoDB configuration.

- a. Set the **MONGOSERVER** parameter in `/opt/moab/etc/moab.cfg` to the MongoDB server hostname. Use `localhost` as the hostname if Moab and MongoDB are hosted on the same server.

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

If your **MONGOSERVER** host is set to anything other than `localhost`, edit the `/etc/mongod.conf` file on the MongoDB server host and either comment out any `bind_ip` parameter or set it to the correct IP address:

```
# Listen to local interface only. Comment out to listen on all interfaces.
#bind_ip=127.0.0.1
```

- b. In the `/opt/moab/etc/moab-private.cfg` file, set the **MONGOUSER** and **MONGOPASSWORD** parameters to the MongoDB `moab_user` credentials you set. See [Install MongoDB in the Dependencies and Packages Installation](#) section earlier in this topic.

```
MONGOUSER      moab_user
MONGOPASSWORD  secret2
```

- c. Verify that Moab is able to connect to MongoDB.

- Red Hat 6-based and SUSE 11-based systems

```
[root]# service moab restart
[root]# mdiag -S
...
Mongo connection (localhost) is up (credentials are set)
...
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# systemctl restart moab.service
[root]# mdiag -S
...
Mongo connection (localhost) is up (credentials are set)
...
```

## 8. Secure communication using secret keys.

- a. (Required) Moab and MWS use Message Authentication Codes (MAC) to ensure messages have not been altered or corrupted in transit. On the Moab head node, generate a key and store the result in `/opt/moab/etc/.moab.key`.

- Red Hat 6-based and SUSE 11-based systems

```
[root]# service moab stop
[root]# dd if=/dev/urandom count=18 bs=1 2>/dev/null | base64 >
/opt/moab/etc/.moab.key
[root]# chown root:root /opt/moab/etc/.moab.key
[root]# chmod 400 /opt/moab/etc/.moab.key
[root]# service moab start
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# systemctl stop moab.service
[root]# dd if=/dev/urandom count=18 bs=1 2>/dev/null | base64 >
/opt/moab/etc/.moab.key
[root]# chown root:root /opt/moab/etc/.moab.key
[root]# chmod 400 /opt/moab/etc/.moab.key
[root]# systemctl start moab.service
```

The key you specify in the `.moab.key` file is the same key you must also specify in the `moab.secretKey` property when installing and configuring MWS in the next step.

- b. (Optional) Moab supports message queue security using AES. This feature requires a Base64-encoded 16-byte (128-bit) shared secret. On the Moab head node, generate a key and append the result to `/opt/moab/etc/moab-private.cfg`.

- Red Hat 6-based and SUSE 11-based systems

```
[root]# service moab stop
[root]# echo "MESSAGEQUEUESECRETKEY $(dd if=/dev/urandom count=16 bs=1
2>/dev/null | base64)" >> /opt/moab/etc/moab-private.cfg
[root]# service moab start
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# systemctl stop moab.service
[root]# echo "MESSAGEQUEUESECRETKEY $(dd if=/dev/urandom count=16 bs=1
2>/dev/null | base64)" >> /opt/moab/etc/moab-private.cfg
[root]# systemctl start moab.service
```

The key you specify in the `moab-private.cfg` file is the same key you must also specify in the `moab.messageQueue.secretKey` property when installing and configuring MWS in the next step.

**i** If MWS is configured to encrypt the message queue and Moab is not (or vice versa), then MWS will ignore the messages from Moab. Furthermore, all attempts to access the MWS service resource will fail.

- c. (Optional) Verify that encryption is on for the ZeroMQ connection.

```
[root]# mdiag -S|grep 'ZeroMQ MWS'
ZeroMQ MWS connection is bound on port 5570 (encryption is on)
```

9. Set up the MWS configuration files. In the extracted directory are several configuration files.

- a. Copy the configuration files into place and grant the tomcat user read access.

```
[root]# cd /tmp/mws-install/mws-8.1.1
[root]# cp mws-config.groovy /opt/mws/etc
[root]# cp mws-config-hpc.groovy /opt/mws/etc/mws.d
[root]# chown tomcat:tomcat /opt/mws/etc/mws-config.groovy
/opt/mws/etc/mws.d/mws-config-hpc.groovy
[root]# chmod 400 /opt/mws/etc/mws-config.groovy /opt/mws/etc/mws.d/mws-config-
hpc.groovy
```

- b. In the `/opt/mws/etc/mws-config.groovy` file, change these settings:
- **moab.secretKey**: Must match the Moab secret key you generated earlier (contained in `/opt/moab/etc/.moab.key`).
  - **auth.defaultUser.username**: Any value you like, or leave as is.

- **auth.defaultUser.password**: Any value you like, but choose a strong password.
- **moab.messageQueue.secretKey**: If you opted to configure a message queue security key in MWS, this parameter value should match exactly that key specified in `/opt/moab/etc/moab-private.cfg` for the `MESSAGEQUEUESECRETKEY` Moab configuration parameter on the Moab head node in the previous step..

**i** Important: If MWS is configured to encrypt the message queue and Moab is not (or vice versa), then the messages from Moab will be ignored. Furthermore, all attempts to access the MWS service resource will fail.

```
[root]# vi /opt/mws/etc/mws-config.groovy

// Replace <ENTER-KEY-HERE> with the contents of /opt/moab/etc/.moab.key.

moab.secretKey = "<ENTER-KEY-HERE>"
moab.server = "localhost"
moab.port = 42559

// Replace <ENTER-KEY-HERE> with the value of MESSAGEQUEUESECRETKEY in
/opt/moab/etc/moab-private.cfg.

moab.messageQueue.secretKey = "<ENTER-KEY-HERE>"

// Change these to be whatever you like.

auth.defaultUser.username = "moab-admin"
auth.defaultUser.password = "changeme!"
```

**⚠** If you do not change **auth.defaultUser.password**, your MWS will not be secure (because anyone reading these instructions would be able to log into your MWS). Here are some [tips](#) for choosing a good password.

c. Do *one* of the following:

**i** You can configure only one authentication method in `mws-config.groovy`—LDAP or PAM, but not both. If you have configured both LDAP and PAM, MWS defaults to using LDAP. If you need multiple authentication methods, you must add them to your local PAM configuration. See your distribution documentation for details.

- If you are configuring an MWS connection to your LDAP server, add the following parameters to `/opt/mws/etc/mws-config.groovy`:

```
ldap.server = "192.168.0.5"
ldap.port = 389
ldap.baseDNs = ["dc=acme,dc=com"]
ldap.bindUser = "cn=Manager,dc=acme,dc=com"
ldap.password = "*****"
ldap.directory.type = "OpenLDAP Using InetOrgPerson Schema"
```

*This is just an example LDAP connection. Be sure to use the appropriate domain controllers (dc) and common names (cn) for your environment.*

**i** If you followed the Adaptive Computing tutorial, [Setting Up OpenLDAP on CentOS 6 on page 78](#), your `ldap.directory.type` should be set to "OpenLDAP Using InetOrgPerson Schema." However, the use of other schemas is supported. See **LDAP Configuration Using `mws-config.groovy`** in the *Moab Web Services Reference Guide*.

**i** To see how to configure a secure connection to the LDAP server, see **Securing the LDAP Connection** in the *Moab Web Services Reference Guide*.

- If you are configuring MWS to use PAM, add the `pam.configuration.service` parameter to the `mws-config.groovy` file. For example:

```
pam.configuration.service = "login"
```

**i** For more information about PAM configuration with MWS, see **PAM (Pluggable Authentication Module) Configuration Using `mws-config.groovy`** in the *Moab Web Services Reference Guide*.

**⚠** There is a security risk when authenticating local users through your PAM configuration. This behavior is highly discouraged and not supported by Adaptive Computing.

- d. Add the `grails.mongo.username` and `grails.mongo.password` parameters to the `mws-config.groovy` file. Use the MWS credentials you added to MongoDB in the [Preparing for Manual Installation on page 12](#).

```
...
grails.mongo.username = "mws_user"
grails.mongo.password = "secret3"
```

## 10. Configure Tomcat

- Red Hat 6-based and Red Hat 7-based systems

Add the following lines to the end of `/etc/tomcat/tomcat.conf`.

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -XX:MaxPermSize=384m -
Dfile.encoding=UTF8"
JAVA_HOME="/usr/java/latest"
```

- **SUSE 11-based systems**

Add the following lines to the end of `/etc/tomcat6/tomcat6.conf`.

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -XX:MaxPermSize=384m -
Dfile.encoding=UTF8"
JAVA_HOME="/usr/java/latest"
```

- **SUSE 12-based systems**

Add the following lines to the end of `/etc/tomcat/tomcat.conf`.

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -XX:MaxPermSize=384m -
Dfile.encoding=UTF8"
JAVA_HOME="/usr/java/latest"
```

## 11. Deploy the `mws.war` file and start Tomcat.

- **Red Hat 6-based systems**

```
[root]# chkconfig tomcat on
[root]# service tomcat stop
[root]# cp /tmp/mws-install/mws-8.1.1/mws.war /usr/share/tomcat/webapps
[root]# service tomcat start
```

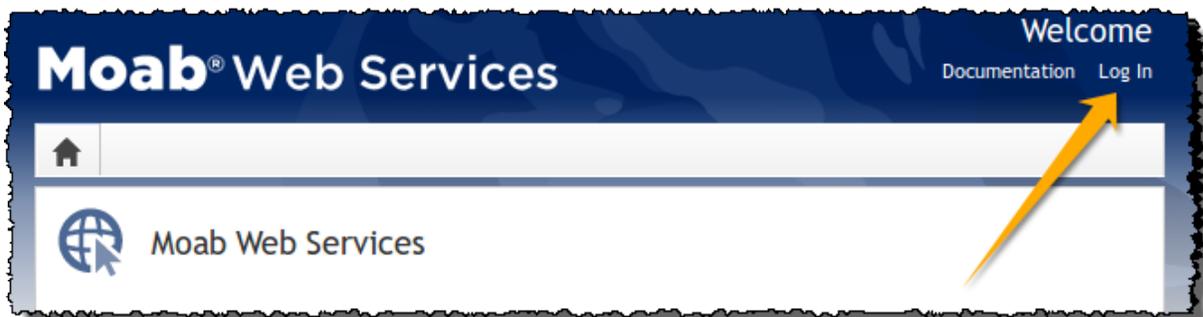
- **SUSE 11-based systems**

```
[root]# chkconfig tomcat6 on
[root]# service tomcat6 stop
[root]# cp /tmp/mws-install/mws-8.1.1/mws.war /usr/share/tomcat6/webapps
[root]# service tomcat6 start
```

- **Red Hat 7-based and SUSE 12-based systems**

```
[root]# systemctl enable tomcat.service
[root]# systemctl stop tomcat.service
[root]# cp /tmp/mws-install/mws-8.1.1/mws.war /usr/share/tomcat/webapps
[root]# systemctl start tomcat.service
```

12. Navigate to `http://<localhost>:8080/mws/` in a web browser to verify that MWS is running (you will see some sample queries and a few other actions).
13. Log in to MWS to verify that your credentials are working. (Your login credentials are the `auth.defaultUser.username` and `auth.defaultUser.password` values you set in the `/opt/mws/etc/mws-config.groovy` file.)



**i** If you encounter problems, or if the application does not seem to be running, see the steps in [Moab Web Services Issues](#) on page 109.

## Additional Configuration

In this section:

- [Configuring SSL in Tomcat on page 78](#)
- [Setting Up OpenLDAP on CentOS 6 on page 78](#)
- [Moab Workload Manager Configuration Options on page 44](#)
- [Trusting Servers in Java on page 85](#)

### Configuring SSL in Tomcat

To configure SSL in Tomcat, please refer to the [Apache Tomcat documentation](#).

### Setting Up OpenLDAP on CentOS 6

These instructions are intended to help first-time LDAP administrators get up and running. The following procedures contain instructions for getting started using OpenLDAP on a CentOS 6 system. For more complete information on how to set up OpenLDAP see the [OpenLDAP documentation](http://www.openldap.org/doc/admin24/) (<http://www.openldap.org/doc/admin24/>).

In this topic:

- [Installing and Configuring OpenLDAP on Centos 6 on page 37](#)
- [Adding an Organizational Unit \(OU\) on page 41](#)
- [Adding a User on page 42](#)
- [Adding a Group on page 43](#)
- [Adding a User to a Group on page 43](#)

**i** Adaptive Computing is not responsible for creating, maintaining, or supporting customer LDAP or Active Directory configurations.

### Installing and Configuring OpenLDAP on Centos 6

First, you will need to install OpenLDAP. These instructions explain how you can do this on a CentOS 6 system.

## To install and configure OpenLDAP on Centos 6

1. Run the following command:

```
[root]# yum -y install openldap openldap-clients openldap-servers
```

2. Generate a password hash to be used as the admin password. This password hash will be used when you create the root user for your LDAP installation. For example:

```
[root]# slappasswd
New password : p@ssw0rd
Re-enter new password : p@ssw0rd
{SSHA}51PFVw19zeh7LT53hQH69znzj8TuBrLv
```

3. Add the root user and the root user's password hash to the OpenLDAP configuration in the `olcDatabase={2}bdb.ldif` file. The root user will have permissions to add other users, groups, organizational units, etc. Do the following:

- a. Run this command:

```
[root]# cd /etc/openldap/slapd.d/cn=config
[root]# vi olcDatabase=\{2\}bdb.ldif
```

- b. If the **olcRootPW** attribute does not already exist, create it. Then set the value to be the hash you created from `slappasswd`. For example:

```
olcRootPW: {SSHA}51PFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

4. While editing this file, change the distinguished name (DN) of the **olcSuffix** to something appropriate. The suffix typically corresponds to your DNS domain name, and it will be appended to the DN of every other LDAP entry in your LDAP tree.

For example, let's say your company is called Acme Corporation, and that your domain name is "acme.com." You might make the following changes to the `olcDatabase={2}bdb.ldif` file:

```
olcSuffix: dc=acme,dc=com
...
olcRootDN: cn=Manager,dc=acme,dc=com
...
olcRootPW: {SSHA}51PFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

**i** Throughout the following examples in this topic, you will see `dc=acme,dc=com`. "acme" is only used as an example to illustrate what you would use as your own domain controller if your domain name was "acme.com." You should replace any references to "acme" with your own organization's domain name.

**⚠ Do not set the cn of your root user to "root" (cn=root, dc=acme, dc=com), or OpenLDAP will have problems.**

5. Modify the DN of the root user in the `olcDatabase={1}monitor.ldif` file to match the **olcRootDN** line in the `olcDatabase={2}bdb.ldif` file. Do the following:

- a. Run this command to edit the `olcDatabase={2}bdb.ldif` file:

```
[root]# vi olcDatabase=\{1\}monitor.ldif
```

- b. Modify the **olcAccess** line so that the **dn.base** matches the **olcRootDN** from the `olcDatabase={2}bdb.ldif` file. (In this example, **dn.base** should be `"cn=Manager,dc=acme,dc=com"`.)

```
olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by
dn.base="cn=Manager,dc=acme,dc=com" read by * none
```

- c. Now the root user for your LDAP is `cn=Manager,dc=acme,dc=com`. The root user's password is the password that you entered using `slappasswd` (see step 2), which, in this example, is **p@ssw0rd**
6. Hide the password hashes from users who should not have permission to view them.

**i** A full discussion on configuring access control in OpenLDAP is beyond the scope of this tutorial. For help, see [the OpenLDAP Access Control documentation](http://www.openldap.org/doc/admin24/access-control.html) (<http://www.openldap.org/doc/admin24/access-control.html>).

- a. Run this command to edit the `olcDatabase=\{2\}bdb.ldif` file:

```
[root]# vi olcDatabase=\{2\}bdb.ldif
```

- b. Add the following two lines to the end of the file to restrict users from viewing other users' password hashes.

```
olcAccess: {0}to attrs=userPassword by self write by
dn.base="cn=Manager,dc=acme,dc=com" write by anonymous auth by * none
olcAccess: {1}to * by dn.base="cn=Manager,dc=acme,dc=com" write by self write by
* read
```

*These lines allow a user to read and write his or her own password. It also allows a manager to read and write anyone's password. Anyone, including anonymous users, is allowed to view non-password attributes of other users.*

7. Make sure that OpenLDAP is configured to start when the machine starts up, and start the OpenLDAP service.

```
[root]# chkconfig slapd on
[root]# service slapd start
```

8. Now, you must manually create the "dc=acme,dc=com" LDAP entry in your LDAP tree.

An LDAP directory is analogous to a tree. Nodes in this tree are called LDAP "entries" and may represent users, groups, organizational units, domain controllers, or other objects. The attributes in each entry are determined by the LDAP schema. In this tutorial we will build entries based on the InetOrgPerson schema (which ships with OpenLDAP by default).

In order to build our LDAP tree we must first create the root entry. Root entries are usually a special type of entry called a domain controller (DC). Because we are assuming that the organization is called Acme Corporation, and that the domain is "acme.com," we will create a domain controller LDAP entry called `dc=acme,dc=com`. Again, you will need to replace "acme" with your organization's domain name. Also note that `dc=acme,dc=com` is what is called an LDAP distinguished name (DN). An LDAP distinguished name uniquely identifies an LDAP entry.

Do the following:

- a. Create a file called `acme.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi acme.ldif
```

- b. Add the following lines in `acme.ldif`:

```
dn: dc=acme,dc=com
objectClass: dcObject
objectClass: organization
dc: acme
o : acme
```

- c. Now add the contents of this file to LDAP. Run this command:

```
[root]# ldapadd -f acme.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

- d. Verify that your entry was added correctly.

```
[root]# ldapsearch -x -LLL -b dc=acme,dc=com
dn: dc=acme,dc=com
objectClass: dcObject
objectClass: organization
dc: acme
o: acme
```

9. Run the following:

```
[root]# sudo iptables -L
[root]# sudo service iptables save
```

10. By default, the CentOS 6 firewall will block external requests to OpenLDAP. In order to allow MWS to access LDAP, you will have to configure your firewall to allow connections on port 389. (Port 389 is the default LDAP port.)

Configuring your firewall is beyond the scope of this tutorial; however, it may be helpful to know that the default firewall on CentOS is a service called `iptables`. (For more information, see the documentation on [iptables](#).) In the most basic case, you may be able to add a rule to your firewall that accepts connections to port 389 by doing the following:

- a. Edit your `iptables` file:

```
[root]# vi /etc/sysconfig/iptables
```

- b. Add the following line *after* all the **ACCEPT** lines but *before* any of the **REJECT** lines in your `iptables` file:

```
# ... lines with ACCEPT should be above
-A INPUT -p tcp --dport 389 -j ACCEPT
# .. lines with REJECT should be below
```

For example, here is a sample `iptables` file with this line added:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -p tcp --dport 389 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT
```

- c. Now reload `iptables`.

```
[root]# service iptables reload
```

**i** Although providing instructions is beyond the scope of this tutorial, it is also highly recommended that you set up OpenLDAP to use SSL or TLS security to prevent passwords and other sensitive data from being sent in plain text. For information on how to do this, see the [OpenLDAP TLS documentation](http://www.openldap.org/doc/admin24/tls.html) (<http://www.openldap.org/doc/admin24/tls.html>).

Now that you have installed and set up Open LDAP, you are ready to add organizational units. See [Adding an Organizational Unit \(OU\)](#) on page 41.

## Adding an Organizational Unit (OU)

These instructions will describe how to populate the LDAP tree with organizational units (OUs), groups, and users, all of which are different types of LDAP entries. The examples that follow also presume an InetOrgPerson schema, because the InetOrgPerson schema is delivered with OpenLDAP by default.

### To add an organizational unit (OU) entry to the LDAP tree

In this example, we are going to add an OU called "Users."

1. Create a temporary file called `users.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi users.ldif
```

2. Add these lines to `users.ldif`:

```
dn: ou=Users,dc=acme,dc=com
objectClass: organizationalUnit
ou: Users
```

3. Add the contents of `users.ldif` file to LDAP.

```
[root]# ldapadd -f users.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Adding a User

### To add a user to LDAP

In this example, we will add a user named "Bob Jones" to LDAP inside the "Users" OU.

1. Create a temporary file called `bob.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi bob.ldif
```

2. Add these lines to `bob.ldif`:

```
dn: cn=Bob Jones,ou=Users,dc=acme,dc=com
cn: Bob Jones
sn: Jones
objectClass: inetOrgPerson
userPassword: p@ssw0rd
uid: bjones
```

3. Add the contents of `bob.ldif` file to LDAP.

```
[root]# ldapadd -f bob.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Adding a Group

### To add a group to LDAP

In this example, we will add a group called "Engineering" to LDAP inside the "Users" OU.

1. Create a temporary file called `engineering.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi engineering.ldif
```

2. Add these lines to `engineering.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
cn: Engineering
objectClass: groupOfNames
member: cn=Bob Jones,ou=Users,dc=acme,dc=com
```

3. Add the contents of `engineering.ldif` file to LDAP.

```
[root]# ldapadd -f engineering.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Adding a User to a Group

### To add a user to an LDAP group

In this example, we will add an LDAP member named "Al Smith" to the "Engineering" LDAP group. This example assumes that user, Al Smith, has already been added to LDAP.

**i** Before you add a user to an LDAP group, the user must first be added to LDAP. For more information, see [Adding a User on page 42](#).

1. Create a temporary file called `addUserToGroup.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi addUserToGroup.ldif
```

2. Add these lines to `addUserToGroup.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
changetype: modify
add: member
member: cn=Al Smith,ou=Users,dc=acme,dc=com
```

### 3. Now add the contents of `addUserToGroup.ldif` file to LDAP.

```
[root]# ldapadd -f addUserToGroup.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Moab Workload Manager Configuration Options

The following is a list of commonly used configure options. For a complete list, use `./configure --help` when configuring Moab.

Option	Description	Example
<code>--with-flexlm</code>	Causes Moab to install the <code>license.mon.flexLM.pl</code> script in the <code>/opt/moab/tools</code> directory. See <b>Interfacing to FLEXlm</b> in the <i>Moab Workload Manager Administrator Guide</i> for more information about this script.	<pre>[root]# ./configure --with-flexlm</pre>
<code>--with-homedir</code>	Specifies the location of the Moab configuration directory and the <code>MOABHOMEDIR</code> environment variable. The default location is <code>/opt/moab</code> . <div data-bbox="483 1108 834 1325" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>i</b> <code>MOABHOMEDIR</code> is automatically set on some distributions during installation, when the <code>--with-profile</code> option is enabled.</p> </div>	<pre>[root]# ./configure --with-homedir=/var/moab</pre> <p><i>The Moab HPC Suite home directory will be <code>/var/moab</code> instead of the default <code>/opt/moab</code>.</i></p>

Option	Description	Example
<b>--with-init</b>	<p>Enables the installation of a distribution-specific <code>/etc/init.d/moab</code> service startup script.</p> <p>This option is required if you want to install this script onto a new system. If you do not set this option, you must manually set up the Moab daemon service.</p> <p>The startup script is located at <code>OS/EL/etc/init.d/moab</code>.</p> <div data-bbox="506 688 859 968" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>i</b> The TORQUE and Moab HPC Suite initialization scripts are provided in the <code>contrib/init.d</code> directory as a courtesy and may be modified at your discretion to work on your system.</p> </div>	<pre>[root]# ./configure --with-init</pre>
<b>--prefix</b>	<p>Specifies the location of the binaries and libraries of the Moab install.</p> <p>The default location is <code>/opt/moab</code>.</p>	<pre>[root]# ./configure --prefix=/usr/local</pre>
<b>--with-profile</b>	<p>Enables the installation of distribution-specific <code>/etc/profile.d/moab.[c]sh</code> setup script for bash and cshell.</p> <p>The <code>MOABHOMEDIR</code>, <code>PERL5LIB</code>, <code>PATH</code> and <code>MANPATH</code> environment variables are setup to specify where the new moab configuration, scripts, binaries and man pages reside. If you do not set this option, these scripts are not installed, and you must manually perform this set up.</p> <p>The environment setup scripts are located at <code>OS/EL/etc/profile.d/moab.[c]sh</code>.</p>	<pre>[root]# ./configure --with-profile</pre>

Option	Description	Example
<b>--with-am</b>	<p>Specifies that you want to configure Moab with Moab Accounting Manager.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>i</b> There is a similar <code>--with-torque</code> option that configures Moab with TORQUE, but you do not need to specify this option if you install the "torque" tarball version.</p> </div>	<pre>[root]# ./configure --with-am</pre>

## Trusting Servers in Java

### Prerequisites

Some of these instructions refer to `JAVA_HOME`, which must point to the same directory that Tomcat uses. To set `JAVA_HOME`, do this:

```
[root]# source /etc/tomcat6/tomcat6.conf
```

Your system administrator might have defined Tomcat's `JAVA_HOME` in a different file.

### Retrieve the Server's X.509 Public Certificate

To retrieve the server's certificate, use the following command:

```
[root]# $JAVA_HOME/bin/keytool -printcert -rfc -sslserver <servername>:<port> > /tmp/public.cert.pem
```

Replace `<servername>` with the server's host name and `<port>` with the secure port number. The default port for https is 443. The default port for ldaps is 636. If successful, `/tmp/public.cert.pem` contains the server's public certificate. Otherwise, `/tmp/public.cert.pem` contains an error message. This message is typical: `keytool error: java.lang.Exception: No certificate from the SSL server.` This message suggests that the server name or port is incorrect. Consult your IT department to determine the correct server name and port.

### Add the Server's Certificate to Java's Keystore

Java stores trusted certificates in a database known as the keystore. Because each new version of Java has its own keystore, you need to add the server

certificate to the Java keystore (using the steps below) every time you install a new version of Java.

Java's keystore is located at `$JAVA_HOME/lib/security/cacerts`. If Tomcat's `JAVA_HOME` points to a JDK, then the keystore is located at `$JAVA_HOME/jre/lib/security/cacerts`. To add the server certificate to the keystore, run the following command:

```
[root]# $JAVA_HOME/bin/keytool -import -trustcacerts -file /tmp/public.cert.pem -alias  
<servername> -keystore $JAVA_HOME/lib/security/cacerts
```

You will be prompted for the keystore password, which is "changeit" by default.

**i** Your system administrator might have changed this password.

After you've entered the keystore password, you'll see the description of the server's certificate. At the end of the description it prompts you to trust the certificate.

```
Trust this certificate? [no]:
```

Type `yes` and press **Enter** to add the certificate to the keystore.

# Upgrade

In this section:

- [Preparing for Upgrade on page 48](#)
- [Upgrading MongoDB on page 49](#)
- [Upgrading TORQUE on page 50](#)
- [Upgrading Moab Workload Manager on page 53](#)
- [Upgrading MWS on page 55](#)

## Preparing for Upgrade

The upgrade process of the Moab HPC Suite includes upgrading the database and separate components in the suite. This guide contains detailed instructions for upgrading each component.

 It is highly recommended that you *first* perform upgrades in a *test environment*. Installation and upgrade procedures are tested prior to release; however, due to customizable variations that may be utilized by your configuration, it is not recommended to drop new versions of software directly into production environments. This is especially true when the workload has vital bearing. Contact Adaptive Computing Professional Services for more information.

 Because many system-level files and directories are accessed during the upgrade, the upgrade instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Please note that the same commands will work for a non-root user with the `sudo` command.

Upgrade the Moab HPC Suite in the following order:

1. Mongo database. See [Upgrading MongoDB on page 49](#)
2. TORQUE. See [Upgrading TORQUE on page 50](#)
3. Moab Workload Manager. See [Upgrading Moab Workload Manager on page 53](#)
4. Moab Web Services. See [Upgrading MWS on page 55](#)

## Related Topics

- [Requirements on page 3](#)

## Upgrading MongoDB

Adaptive Computing recommends MongoDB version 2.4.x.

When upgrading, you must add 'exclude=mongodb-org mongodb-org-server' to the `/etc/yum.repos.d/10gen.repo` file to maintain 2.4.x:

```
[10gen]
name=MongoDB Repository
baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86_64/
gpgcheck=0
enabled=1
exclude=mongodb-org mongodb-org-server
```

### Upgrading from MongoDB Version 2.0

Support for environments using MongoDB 2.0 is now deprecated and will be removed in a future release.

1. Verify you can connect to the Mongo database.
  - a. Obtain the Mongo username and password.

```
[root]# grep grails.mongo /opt/mws/etc/mws-config.groovy
grails.mongo.username = "mws_user"
grails.mongo.password = "secret3"
```

- b. Using the Mongo username and password (in our example, username is "mws\_user" and password is "secret3"), confirm you can log in.

```
[root]# service mongod start
[root]# mongo -u mws_user -p secret3 mws
MongoDB shell version: 2.4.12
connecting to: mws
> show collections
event
mongeez
pluginInstance
...
```

2. Refer to [docs.mongodb.org](http://docs.mongodb.org) for instructions on how to upgrade MongoDB. Note that you must pay close attention to the information regarding instances with auth enabled (as this is the recommended setup for Moab HPC Suite).

### 3. Remove version 2.0 and install 2.4

```
[root]# service mongod stop
[root]# yum remove mongo20-10gen-server mongo20-10gen
[root]# yum install mongo-10gen-server
[root]# service mongod start
```

**i** Note that the settings in the `/etc/mongod.conf` file were saved in `/etc/mongod.conf.rpmsave` while removing MongoDB 2.0. You may need to restore any custom settings after MongoDB 2.4.x is installed in the new `/etc/mongod.conf` file (for example, "auth = true").

4. After upgrading from 2.0 to 2.4.x, you should verify that the MongoDB credentials were preserved. Refer to step 1.

## Upgrading TORQUE

TORQUE 5.1.1 binaries are backward compatible with TORQUE 5.0.x. However they are not backward compatible with TORQUE versions prior to 5.0.x. When you upgrade to TORQUE 5.1.1 from versions prior to 5.0.x, all MOM and server daemons must be upgraded at the same time.

The job format is compatible between 5.1.1 and previous versions of TORQUE. Any queued jobs will upgrade to the new version with the exception of job arrays in TORQUE 2.4 and earlier. It is not recommended to upgrade TORQUE while jobs are in a running state.

### Job Arrays

Job arrays from TORQUE version 2.5 and 3.0 are compatible with TORQUE 5.1.1. Job arrays were introduced in TORQUE version 2.4 but modified in 2.5. If upgrading from TORQUE 2.4, you need to make sure all job arrays are complete before upgrading.

### serverdb

The `pbs_server` configuration is saved in the file `$TORQUE_HOME/server_priv/serverdb`. When running TORQUE 4.1.0 or later for the first time, this file converts from a binary file to an XML-like format. This format can be used by TORQUE versions 2.5 and 3.0, but not earlier versions. Back up the `$TORQUE_HOME/server_priv/serverdb` file before moving to TORQUE 4.1.0 or later.

### Jobs

Before upgrading the system, all running jobs must complete. To prevent queued jobs from starting, nodes can be set to offline or all queues can be disabled. Once all running jobs are complete, the upgrade can be made.

Remember to allow any job arrays in version 2.4 to complete before upgrading. Queued array jobs will be lost.

## Cray

For upgrading TORQUE to 5.1.1 on a Cray system, refer to **Installation Notes for Moab and TORQUE for Cray** in the *Moab Workload Manager Administrator Guide*

### To Upgrade TORQUE

#### 1. Shut down TORQUE.

```
[root]# qterm
[root]# momctl -s

* If running TORQUE 4.6.0 or later *
[root]# trqauthd -d

*If running a version of TORQUE earlier than 4.6.0 *
[root]# ps -efw | grep trqauthd
root      1487      1  0 Dec18 ?        00:00:00 /usr/sbin/trqauthd
adaptive 4830  4374  0 15:07 pts/0    00:00:00 grep trqauthd

[root]# kill -9 1487
```

#### 2. Back up your server\_priv directory.

```
[root]# tar -cvf backup.tar.gz $TORQUE_HOME/server_priv
```

#### 3. If not already installed, install the Boost C++ headers.

```
[root]# yum install boost-devel
```

**i** For SLES, use `zypper install <package names>` instead of `yum install <package names>`.

#### 4. Install the latest TORQUE tarball.

```
[root]# cd /tmp
[root]# tar xzvf torque-<version>-<build number>.tar.gz
[root]# cd torque-<version>-<build number>
[root]# ./configure
[root]# make
[root]# make install
```

#### 5. Follow this step to copy in the new init.d scripts and to merge in any local customizations or merge changes in the stock scripts into your customized ones. Also, follow this step to configure `pbs_server` and `pbs_mom` to start automatically at system boot (if you have not done so already).

a. Update the TORQUE pbs\_server service.

On the TORQUE Server Head Node, update the pbs\_server service startup script.

```
# Make a backup of your current service startup script
[root]# cp /etc/init.d/pbs_server pbs_server.bak

# Copy in the new stock service startup script
* If Debian distribution, do the following *
[root]# cp contrib/init.d/debian.pbs_server /etc/init.d/pbs_server
* If SLES distribution, do the following *
[root]# cp contrib/init.d/suse.pbs_server /etc/init.d/pbs_server
* If RHEL distribution, do the following *
[root]# cp contrib/init.d/pbs_server /etc/init.d

# Merge in any customizations
[root]# vi /etc/init.d/pbs_server

# If you have not done so already, configure the services to start automatically
at system boot.
[root]# chkconfig --add pbs_server
```

b. Update the TORQUE pbs\_mom service.

On each TORQUE Mom node, update the pbs\_mom service startup script.

```
# Make a backup of your current service startup script
[root]# cp /etc/init.d/pbs_mom pbs_mom.bak

# Copy in the new stock service startup script
* If Debian distribution, do the following *
[root]# cp contrib/init.d/debian.pbs_mom /etc/init.d/pbs_mom
* If SLES distribution, do the following *
[root]# cp contrib/init.d/suse.pbs_mom /etc/init.d/pbs_mom
* If RHEL distribution, do the following *
[root]# cp contrib/init.d/pbs_mom /etc/init.d

# Merge in any customizations
[root]# vi /etc/init.d/pbs_mom

# If you have not done so already, configure the services to start automatically
at system boot.
[root]# chkconfig --add pbs_mom
```

c. Update the TORQUE trqauthd service.

On each node on which TORQUE clients are installed, update the trqauthd service startup script.

```
# Copy in the new stock service startup script
* If Debian distribution, do the following *
[root]# cp contrib/init.d/debian.trqauthd /etc/init.d/trqauthd
* If SLES distribution, do the following *
[root]# cp contrib/init.d/suse.trqauthd /etc/init.d/trqauthd
* If RHEL distribution, do the following *
[root]# cp contrib/init.d/trqauthd /etc/init.d

# If you have not done so already, configure the services to start automatically
at system boot.
[root]# chkconfig --add trqauthd
```

## 6. Start the services.

```
[root]# service trqauthd start
[root]# service pbs_mom start
[root]# service pbs_server start
```

## 7. Check the status of jobs in the queue and perform other checks for errors.

```
[root]# qstat
[root]# grep -i error /var/spool/torque/server_logs/*
[root]# grep -i error /var/spool/torque/mom_logs/*
```

# Upgrading Moab Workload Manager

The following instructions will guide you through a 6.1.x, 7.0.x, 7.1.x, 7.2.x, 7.7.x, 8.0.x to 8.1.1 upgrade. Depending on which version of Moab you are presently running, upgrade instructions may vary, so unless otherwise noted, all instructions assume use of a RHEL operating system; notes for SLES users are added in appropriate places.

You might want to test (see the newest version of Moab on your system (before making the new version live) to verify your policies, scripts, and queues work the way you want them to. See **Testing New Releases** in the *Moab Workload Manager Administrator Guide*.

If you are also upgrading TORQUE from an older version (pre-4.0), you may encounter a problem where Moab HPC Suite core files are regularly created in `/opt/moab`. This can be caused by old TORQUE library files used by Moab that try to authorize with the old TORQUE `pbs_iff` authorization daemon. You can resolve the problem by removing the old version library files from `/usr/local/lib`.

**i** Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Please note that the same commands will work for a non-root user with the `sudo` command.

## To Upgrade Moab

1. Verify `/etc/yum.repos.d/epel.repo` exists and has the following lines. If not, create it and add these lines.

```
[epel]
name=Extra Packages for Enterprise Linux 6 - x86_64
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=epel-6&arch=x86_64
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=http://download.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-6
```

**i** SLES users must add a repository to YaST. The URL of the repository is [http://download.opensuse.org/repositories/server:/database/SLE\\_11\\_SP3/](http://download.opensuse.org/repositories/server:/database/SLE_11_SP3/).

2. Untar the distribution file. For example:

```
[root]# tar -xzvf moab-<version>-<OS>.tar.gz
```

3. Change directory into the extracted directory.
4. Configure the installation package.

Use the same configure options as when Moab was installed previously. If you cannot remember which options were used previously, check the `config.log` file in the directory where the previous version of Moab was installed from.

For a complete list of configure options, use `./configure --help`.

5. Stop Moab.

```
[root]# mschedctl -k
moab will be shutdown immediately
```

**i** While Moab is down, all currently running jobs continue to run on the nodes, the job queue remains intact, and new jobs cannot be submitted to Moab.

6. Before proceeding to the following steps, consider backing up your Moab Workload Manager home directory (`/opt/moab/` by default).
7. If you are using green computing, or if you are using a resource manager other than TORQUE, run the `make perldeps` command to install the necessary perl modules using CPAN (and install CPAN if you have not already done so). When first running CPAN, you will be asked for configuration information. It is recommended that you choose an automatic configuration. You will be prompted to provide input during module installation; running the `make perldeps` command with a script is not recommended.

```
[root]# yum install perl-CPAN
[root]# make perldeps
```

## 8. Install Moab.

```
[root]# make install
```

**i** Default configuration files are installed during `make install`. Existing configuration files are not overwritten and the new files are given a `.dist` extension.

- If you use ODBC, you must upgrade to the 8.1.1 schema. See **Migrating Your Database to Newer Versions of Moab** in the *Moab Workload Manager Administrator Guide* for more information.
- Verify the version number is correct before starting the new server version.

```
[root]# moab --about

Defaults:      server=:42559  cfgdir=/opt/moab  vardir=/opt/moab
Build dir:    /tmp/develop
Build host:   crom
Build date:   Mon Jun 16 16:00:00 MST 2014
Build args:   NA
Compiler Flags:  -D_M64 -D_BUILDDATETIME="2014061616" -DMUSEWEBSERVICES -
DMUSEZEROMQ -DMUSEMONGODB -DMMAX_GRES=512 -DMMAX_RANGE=2048 -DMMAX_TASK=32768 -fPIC
-gdwarf-3 -Wall -Wextra -DVALGRIND -x c++ -std=c++11 -DDMAX_PJOB=512 -D_GNU_SOURCE
Compiled as little endian.
Version: moab server master (revision 2014061616, changeset
90ce9f804ddd09b061238e438ecb4d117cc83e81)
Copyright (C) 2000-2014 by Adaptive Computing Enterprises, Inc. All Rights
Reserved.
```

## 11. Start Moab.

```
[root]# moabd
```

- If you will be using Moab Web Services, you must configure a secret key. See [Secure communication using secret keys. on page 31](#)

## Upgrading MWS

Before upgrading MWS, we recommend you upgrade to Java 7 and MongoDB 2.4.x. To upgrade Java, repeat the instructions to [Install Java on page 26](#). To upgrade MongoDB, see [Upgrading MongoDB on page 49](#).

**⚠** It is highly recommended that you perform a full database backup before updating your database. This can be done using the `mongodump` utility documented in the [MongoDB documentation](#) (<http://www.mongodb.org/display/DOCS/Backups>).

## To Upgrade MWS

1. Extract the contents of the MWS download tarball into a temporary directory. For example:

```
[root]# mkdir /tmp/mws-install
[root]# cd /tmp/mws-install
[root]# mv mws-8.1.1.tar.gz .
[root]# tar xvzf mws-8.1.1.tar.gz
[root]# cd /tmp/mws-install/mws-8.1.1
```

2. Stop Tomcat, re-deploy `mws.war`, and remove the exploded `mws` directory.

```
# CentOS 6 example

[root]# service tomcat6 stop
[root]# cp /tmp/mws-install/mws-8.1.1/mws.war /usr/share/tomcat6/webapps
[root]# rm -rf /usr/share/tomcat6/webapps/mws
```

3. Create the MWS home directory and subdirectories. See **Configuration** in the *Moab Web Services Reference Guide*.

**i** The default location for the MWS home directory is `/opt/mws`. These instructions assume the default location.

Here is a sample script for this setup:

```
[root]# mkdir -p \
/opt/mws/etc/mws.d \
/opt/mws/hooks \
/opt/mws/log \
/opt/mws/plugins \
/opt/mws/spool/hooks \
/opt/mws/utils
[root]# chown -R tomcat:tomcat /opt/mws # Depending on your OS, the Tomcat username
might be tomcat6.
[root]# chmod -R 555 /opt/mws
[root]# chmod u+w \
/opt/mws/log \
/opt/mws/plugins \
/opt/mws/spool \
/opt/mws/spool/hooks \
/opt/mws/utils
```

4. Copy the extracted utility files to the utility directory created above and give the tomcat user ownership of the directory.

```
[root]# cd /tmp/mws-install/mws-8.1.1/utils
[root]# cp * /opt/mws/utils
[root]# chown tomcat:tomcat /opt/mws/utils/*
```

5. Set up the MWS configuration files. In the extracted directory are several configuration files.

- a. Merge the changes in the new `mws-config.groovy` file into your existing `/opt/mws/etc/mws-config.groovy`.

```

moab.messageQueue.port = 5563
moab.messageQueue.port = 5570

log4j = {
  // Configure an appender for the events log.
  def eventAppender = new org.apache.log4j.rolling.RollingFileAppender(
    name: 'events', layout: pattern(conversionPattern: "%m%n"))
  def rollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
    fileNamePattern: '/opt/mws/log/events.%d{yyyy-MM-dd}',
    activeFileName: '/opt/mws/log/events.log')
  rollingPolicy.activateOptions()
  eventAppender.setRollingPolicy(rollingPolicy)

  // Configure an appender for the audit log.
  def auditAppender = new org.apache.log4j.rolling.RollingFileAppender(
    name: 'audit',
    layout: new com.ace.mws.logging.ACPatternLayout("%j\t\t\t%c{1}
\t\t\t%m%n"))
  def auditRollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
    fileNamePattern: '/opt/mws/log/audit.%d{yyyy-MM-dd}',
    activeFileName: '/opt/mws/log/audit.log')
  auditRollingPolicy.activateOptions()
  auditAppender.setRollingPolicy(auditRollingPolicy)

  appenders {
    rollingFile name: 'stacktrace',
      file: '/opt/mws/log/stacktrace.log',
      maxFileSize: '100MB'
    rollingFile name: 'rootLog',
      file: '/opt/mws/log/mws.log',
      maxFileSize: '100MB', //The maximum file size for a single log file
      maxBackupIndex: 10, //Retain only the 10 most recent log files, delete
older logs to save space
      layout: pattern(conversionPattern: '%d %p %c %m%n'), //Configures the
output format of each log entry
      layout: new com.ace.mws.logging.ACPatternLayout(), //Configures the
output format of each log entry
      threshold: org.apache.log4j.Level.ERROR //Ignore any logging entries
less verbose than this threshold

    appender eventAppender
    appender auditAppender
  }

  // NOTE: This definition is a catch-all for any logger not defined below
  root {
    error 'rootLog'
  }

  // Individual logger configurations
  ...

  // Logs event information to the events log, not the rootLog
  trace additivity:false, events:'com.ace.mws.events.EventFlatFileWriter'
  // Logs audit information to the audit log, not the rootLog
  trace additivity:false, audit:'mws.audit'
}

```

Additions are noted in red. Removed content is stricken out.

Pay special attention to these changes:

- default moab message queue port
  - addition of the "auditAppender" in the default logging configuration of `/tmp/mws-install/mws-8.1.1/mws-config.groovy`
  - **mws.suite** parameter and the **mam.\*** parameters have been moved to a suite-specific file in `/opt/mws/etc/mws.d/` and do not need to exist in `/opt/mws/etc/mws-config.groovy`
  - value for **moab.messageQueue.secretKey** should match the value located in `/opt/moab/etc/moab-private.cfg`; if you have not yet configured a secret key, see [Secure communication using secret keys. on page 31](#)
- b. Merge any changes supplied in the new `mws-config-hpc.groovy` file in to your installed `/opt/mws/etc/mws.d/mws-config-hpc.groovy`.
  - c. Verify the Tomcat user has read access to `/opt/mws/etc/mws-config.groovy` and `/opt/mws/etc/mws.d/mws-config-hpc.groovy` file.
6. Upgrade the schema of the `mws` database in MongoDB.

 You *must* perform this step, regardless of whether you upgraded MongoDB to version 2.4.x or not.

Run the database migration script provided with MWS. (It is safe to run this script more than once. If for any reason, errors occur during the execution of the script, run it again.)

```
[root]# mongo -u mws_user mws /opt/mws/utils/db-migrate.js -p
```

 The script might take several minutes to execute.

7. Start Tomcat.

```
[root]# service tomcat6 start
```

8. Visit `http://<localhost>:8080/mws/` in a web browser to verify that MWS is running again.

You will see some sample queries and a few other actions.

9. Log into MWS to verify configuration. (The credentials are the values of **auth.defaultUser.username** and **auth.defaultUser.password** set in `/opt/mws/etc/mws-config.groovy`.)

**i** If you encounter problems, or if MWS does not seem to be running, see the steps in [Troubleshooting on page 105](#).

## Migrating the MAM Database from MySQL to PostgreSQL

PostgreSQL is now the preferred DBMS for MAM. Customers who have already installed MySQL as the DBMS for MAM are not required to migrate their database to use PostgreSQL at this time. However, MySQL is considered deprecated and new installations will only use PostgreSQL.

**i** PostgreSQL does not provide a standard procedure for migrating an existing database from MySQL to PostgreSQL. Adaptive Computing has had success using the `py-mysql2pgsql` tools for migrating/converting/exporting data from MySQL to PostgreSQL. See <https://github.com/philipsoutham/py-mysql2pgsq> for additional details.

### To Migrate the MAM Database

This procedure was successfully tested on an actual customer MySQL database with millions of transactions on CentOS 6.4. It completed in less than an hour.

1. Make a backup copy of your MySQL mam database.

```
[root]# mysqldump mam > /archive/mam.mysql
```

2. Follow the instructions to Install PostgreSQL in the *Moab HPC Suite Installation and Configuration Guide*.

- Manual Install - [Preparing for Manual Installation on page 12](#)
- RPM Install - [Installing Moab HPC Suite RPM on page 62](#)

3. Install the prerequisite packages.

```
[root]# yum install git postgresql-devel gcc MySQL-python python-psycopg2 PyYAML
termcolor python-devel
```

4. Install `pg-mysql2pgsql` (from source).

```
[root]# cd /software
[root]# git clone git://github.com/philipsoutham/py-mysql2pgsql.git
[root]# cd py-mysql2pgsql
[root]# python setup.py install
```

5. Run `pg-mysql2pgsql` once to create a template yaml config file.

```
[root]# py-mysql2pgsql -v
```

6. Edit the config file to specify the MySQL database connection information and a file to output the result.

```
[root]# vi mysql2pgsql.yml
```

```
mysql:
hostname: localhost
port: 3306
socket:
username: mam
password: changeme
database: mam
compress: false
destination:
# if file is given, output goes to file, else postgres
file: /archive/mam.pgsql
postgres:
hostname: localhost
port: 5432
username:
password:
database:
```

7. Run the pg-mysql2pgsql program again to convert the database.

```
[root]# py-mysql2pgsql -v
```

8. Create the mam database in PostgreSQL.

```
[root]# su - postgres
[postgres]$ psql
postgres=# create database "mam";
postgres=# create user mam with password 'changeme!';
postgres=# \q
[postgres]$ exit
```

9. Import the converted data into the PostgreSQL database.

```
[root]# su - mam
[mam]$ psql mam < /archive/mam.pgsql
```

10. Point MAM to use the new postgresql database.

```
[mam]$ cd /software/mam-latest
[mam]$ ./configure # This will generate an etc/goldd.conf.dist
file
[mam]$ vi /opt/mam/etc/goldd.conf # Merge in the database.datasources from
etc/goldd.conf.dist
```

11. Restart Moab Accounting Manager.

```
[mam]$ goldd -r
```

## Chapter 3 RPM installation Guide

This chapter provides installation, configuration, and upgrading information using the RPM install process.

### Related Topics

- [Requirements on page 3](#)
-

# Installation

In this section:

- [Installing Moab HPC Suite RPM on page 62](#)
- [Configuring TORQUE on page 68](#)
- [Configuring Moab Workload Manager on page 71](#)
- [Configuring Moab Web Services on page 73](#)

## Installing Moab HPC Suite RPM

This topic contains instructions on how to install TORQUE Resource Manager, Moab Workload Manager, and Moab Web Services installed on one server using the 8.1.1 suite RPM.

**i** The RPM installation only supports installation on CentOS 6.x, 7.x, RHEL 6.x, 7.x, or Scientific Linux 6.x, 7.x. See [Preparing for Manual Installation on page 12](#) if installing on other supported operating systems.

**i** Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Please note that the same commands will work for a non-root user with the `sudo` command.

**i** If you want to build a custom RPM for component documentation (Moab Workload Manager or Moab Web Services), see the instructions in the [Installing](#) topic for the respective component (Manual Installation section).

In this topic:

- [Install Java on page 62](#)
- [Installing the Suite RPM on page 63](#)
- [Installing PostgreSQL on page 67](#)

## Install Java

Install the Linux x64 RPM version of Oracle® Java® 7 Runtime Environment.



The Oracle® Java® download page has moved and requires a web-enabled workstation to accept the license agreement and download the software.



Oracle Java 7 Runtime Environment is the recommended Java environment, but Oracle Java 6 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run Moab Web Services.

Do the following:

1. Download Java 7 on the web-enabled workstation.
  - a. Open the web browser and connect to the [Java downloads page](http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html) (<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>).
  - b. Select the radio button to accept the license agreement.
  - c. Click the download link for the Linux x64 rpm file.
2. Copy the Java 7 download to the server using scp, rsync, or any similar network copy utility.
3. On the server, run the following to install Java 7:

```
[root]# rpm -Uh jre-7-linux-x64.rpm
```

## Installing the Suite RPM

1. If you are upgrading from a previous installation of Moab, back up your `/opt/moab/tools` directory to prevent losing modifications made to the perl scripts. If you are performing a clean installation of Moab HPC Suite, skip this step.

```
[root]# tar czf backup-tools.tar.gz /opt/moab/tools
```

2. Download the latest 8.1.1 RPM suite tarball (`moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz`, for example) from the [Adaptive Computing](#) website.
3. Untar the downloaded package.

```
[root]# tar zxf moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz
```

4. Change directories into the untarred directory.



Consider reviewing the README file for additional details on using the RPM distribution tarball.

5. Install the suite repositories. The `-y` option installs with the default settings for the RPM suite.

**i** For a description of the options of the repository installer script, run:

```
[root]# ./install-rpm-repos.sh -h
```

```
[root]# ./install-rpm-repos.sh [<repository-directory>] [-y]
```

**i** If the installation returns the following warning line:

```
Warning: RPMDDB altered outside of yum.
```

This is normal and can safely be ignored.

The [*<repository-directory>*] option is the directory where you want to copy the RPMs. If no argument is given, run "`install-rpm-repos.sh -h`" and note the default directory location. If the [*<repository-directory>*] already exists, RPMs will be added to the existing directory. No files are overwritten in [*<repository-directory>*]. A repository file is also created in `/etc/yum.repos.d/` and points to the [*<repository-directory>*] location.

For ease in repository maintenance, the install script fails if Adaptive Computing RPMs are copied to different directories. If a non-default [*<repository-directory>*] is specified, please use the same directory for future updates.

The script installs the `createrepo` package and its dependencies. You must answer "y" to all the questions in order for the RPM install of the suite to work. Additionally, the script installs the EPEL and 10gen repositories.

6. Test the repository.

```
[root]# yum search moab
```

If no error is given, the repository is correctly installed. The output will look similar to the following (varying slightly depending on the suite and build type):

```

...
moab-hpc-basic-suite.noarch : Moab HPC Basic Suite virtual package
moab-perl-RRDs.noarch : Moab RRDs
moab-tomcat-config.x86_64 : Tomcat Configuration for Web Services
moab-verify-oracle-java.noarch : Java Validator for Web Services
moab-web-services.x86_64 : Moab Web Services
moab-workload-manager.x86_64 : Moab Workload Manager
moab-workload-manager-client.x86_64 : Moab Workload Manager Client
moab-workload-manager-common.x86_64 : Moab Workload Manager Common Files
moab-perl-data.noarch : Perl Configuration for perl packages by Adaptive Computing
moab-torque-client.x86_64 : TORQUE Client
moab-torque-common.x86_64 : TORQUE Common Files
moab-torque-devel.x86_64 : TORQUE Development Files
moab-torque-mom.x86_64 : TORQUE MOM agent
moab-torque-server.x86_64 : TORQUE Server
moab-web-services-hpc-configuration.x86_64 : MWS configuration for HPC
moab-workload-manager-hpc-configuration.x86_64 : MWM configuration for HPC

```

## 7. Install the suite package.

```
[root]# yum install moab-hpc-basic-suite
```

### **i** If you encounter the following error:

```

...
--> Finished Dependency Resolution
krb5-workstation-1.6.1-62.el5.x86_64 from installed has depsolving problems
--> Missing Dependency: krb5-libs = 1.6.1-62.el5 is needed by package
krb5-workstation-1.6.1-62.el5.x86_64 (installed)
krb5-workstation-1.6.1-62.el5.x86_64 from installed has depsolving problems
--> Missing Dependency: krb5-libs = 1.6.1-62.el5 is needed by package
krb5-workstation-1.6.1-62.el5.x86_64 (installed)
Error: Missing Dependency: krb5-libs = 1.6.1-62.el5 is needed by package
krb5-workstation-1.6.1-62.el5.x86_64 (installed)
You could try using --skip-broken to work around the problem
You could try running: package-cleanup --problems
package-cleanup --dupes
rpm -Va --nofiles --nodigest

```

Install the `krb5-workstation` package, then execute the install suite package again.

```
[root]# yum install krb5-workstation
[root]# yum install moab-hpc-basic-suite
```

### **i** If you encounter CURL library errors, make sure you are installing the correct version for your OS.

## 8. Install and prepare the MongoDB database by doing the following:

### a. Install `mongo-10gen-server`.

```
[root]# yum install mongo-10gen-server
```

## b. Start MongoDB.

**i** There may be a short delay (approximately three minutes) for MongoDB to start the first time.

- Red Hat 6-based systems

```
[root]# chkconfig mongod on
[root]# service mongod start
```

- Red Hat 7-based systems

```
[root]# cat > /usr/lib/systemd/system/mongodb.service <<End-of-file
[Unit]
Description=High-performance, schema-free document-oriented database
After=syslog.target network.target

[Service]
Type=forking
User=mongod
Group=mongod
Environment=CONFIG=/etc/mongod.conf
Environment=OPTIONS=
EnvironmentFile=-/etc/sysconfig/mongod
ExecStart=/usr/bin/mongod -f \${CONFIG} \${OPTIONS}
PrivateTmp=true
LimitNOFILE=65536
TimeoutStartSec=180
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
End-of-file
[root]# rm -f /etc/init.d/mongod
[root]# systemctl enable mongodb.service
[root]# systemctl start mongodb.service
```

## c. Add the required MongoDB users.

**i** The passwords used below (`secret1`, `secret2`, and `secret3`) are examples. Choose your own passwords for these users.

```
[root]# mongo
> use admin;
> db.addUser("admin_user", "secret1");
> db.auth("admin_user", "secret1");

> use moab;
> db.addUser("moab_user", "secret2");
> db.addUser("mws_user", "secret3", true);

> use mws;
> db.addUser("mws_user", "secret3");
> exit
```

**i** Because the `admin_user` has read and write permissions to the `admin` database, it also has read and write permissions to all other databases. See [Control Access to MongoDB Instances with Authentication](http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication/) (<http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication/>) for more information.

d. Enable authentication in MongoDB.

- Red Hat 6-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# service mongod restart
```

- Red Hat 7-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# systemctl restart mongod.service
```

## Installing PostgreSQL

If you plan to use Moab Workload Manager with ODBC, you must install a PostgreSQL database.

### To install PostgreSQL

1. Install and initialize PostgreSQL.

- Red Hat 6-based systems

```
[root]# yum install postgresql-server
[root]# service postgresql initdb
```

- Red Hat 7-based systems

```
[root]# yum install postgresql-server
[root]# postgresql-setup initdb
```

2. Configure trusted connections.

Modify the "host" lines in the `pg_hba.conf` file for the interfaces from which the server(s) (for example, Moab Workload Manager and/or Moab Accounting Manager) will be connecting to the database and ensure that it specifies a secure password-based authentication method (for example, `md5`).

```
[root]# vi /var/lib/pgsql/data/pg_hba.conf

# IPv4 local connections:
host    all             all             127.0.0.1/32      md5
# IPv6 local connections:
host    all             all             ::1/128          md5
```

**i** If the "host" lines are not present, add them as they appear above.

### 3. Configure PostgreSQL to accept connections from your host.

```
[root]# vi /var/lib/pgsql/data/postgresql.conf

# Uncomment the listen addresses line in the configuration:

listen_addresses = 'localhost'          # what IP address(es) to listen on;
```

### 4. Start or restart the database.

- Red Hat 6-based systems

```
[root]# chkconfig postgresql on
[root]# service postgresql restart
```

- Red Hat 7-based systems

```
[root]# systemctl enable postgresql.service
[root]# systemctl restart postgresql.service
```

## Configuring TORQUE

This topic contains instructions on how to configure and start TORQUE.

**⚠** If you intend to use TORQUE 5.1.1 with Moab, you must run Moab version 8.1.1 or 8.0.x. TORQUE 5.1.1 will not work with versions earlier than Moab 8.0.

In this topic:

- [Prerequisites on page 68](#)
- [Configure TORQUE on page 69](#)
- [Install TORQUE MOMs on page 70](#)

## Prerequisites

### Open Necessary Ports

TORQUE requires certain ports to be open for essential communication:

- For client communication to `pbs_server`, all privileged ports must be

open (ports under 1024).

- For `pbs_server` communication to `pbs_mom`, the default port is 15003.
- For `pbs_mom` to `pbs_server`, the default port is 15001.

For more information on how to configure the ports that TORQUE uses for communication, see **Configuring Ports** in the *TORQUE Resource Manager Administrator Guide*.

If you have a firewall enabled, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the TORQUE server for client and MOM communication
-A INPUT -p tcp --dport 15001 -j ACCEPT

# Needed on the TORQUE MOM for server and MOM communication
-A INPUT -p tcp --dport 15002 -j ACCEPT
-A INPUT -p tcp --dport 15003 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=15001/tcp --permanent
[root]# firewall-cmd --add-port=15002/tcp --permanent
[root]# firewall-cmd --add-port=15003/tcp --permanent
[root]# firewall-cmd --reload
```

### Verify the hostname

Make sure your host (with the correct IP address) is in your `/etc/hosts` file. To verify that the hostname resolves correctly, make sure that `hostname` and `hostname -f` report the correct name for the host.

## Configure TORQUE

Do the following:

1. Add nodes to the `/var/spool/torque/server_priv/nodes` file. For information on syntax and options for specifying compute nodes, see **Managing Nodes** in the *TORQUE Resource Manager Administrator Guide*.
2. Start the servers.
  - Red Hat 6-based systems

```
[root]# service trqauthd start
[root]# service pbs_server start
[root]# service pbs_mom start
```

- Red Hat 7-based systems

```
[root]# systemctl start trqauthd.service
[root]# systemctl start pbs_server.service
[root]# systemctl start pbs_mom.service
```

## Install TORQUE MOMs

Do the following:

1. In most installations, you will install a TORQUE MOM on each of your compute nodes. To install the TORQUE MOMs in the directory of the unpackaged tarball, copy the `moab-torque-common` and `moab-torque-mom` RPM files to each MOM node.

```
[root]# scp RPMs/moab-torque-common-*.rpm <mom-node>
[root]# scp RPMs/moab-torque-mom-*.rpm <mom-node>
```

2. On each MOM node, install the RPMs, making sure that `moab-torque-common` is installed first.

```
[root]# ssh root@<mom-node>
[root]# yum install moab-torque-common-*.rpm moab-torque-mom-*.rpm
```

3. On each MOM node, create or edit the `/var/spool/torque/server_name` file to contain the hostname of the TORQUE server.

```
[root]# echo <torque_server_hostname> > /var/spool/torque/server_name
```

4. Edit the `/var/spool/torque/mom_priv/config` file on each TORQUE MOM node. This file is identical for all compute nodes and can be created on the head node and distributed in parallel to all systems.

```
[root]# vi /var/spool/torque/mom_priv/config

$pbsserver      <torque_server_hostname> # hostname running pbs server
$logevent       225                      # bitmap of which events to log
```

5. On each TORQUE server MOM node, start the `pbs_mom` daemon.

- Red Hat 6-based systems

```
[root]# service pbs_mom start
```

- Red Hat 7-based systems

```
[root]# systemctl start pbs_mom.service
```

### Related Topics

[Installing Moab HPC Suite RPM on page 62](#)

[Configuring Moab Workload Manager on page 71](#)

## Configuring Moab Workload Manager

This topic contains instructions on how to configure and start Moab Workload Manager (Moab).

In this topic:

- [Open Necessary Ports on page 71](#)
- [Configure Moab Workload Manager on page 71](#)

### Open Necessary Ports

Moab Workload Manager uses a configurable server port (default 42559) for client-server communication. If you intend to run client commands on a host other than the Moab Head Node, or if you will be using Moab in a grid, and if you have a firewall enabled, then you will need to configure the firewall to allow the server port.

Do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the Moab server for off-host client communication
-A INPUT -p tcp --dport 42559 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=42559/tcp --permanent
[root]# firewall-cmd --reload
```

### Configure Moab Workload Manager

Do the following:

1. Source the following file to add the Moab executable directories to your current shell *\$PATH* environment.

```
[root]# . /etc/profile.d/moab.sh
```

2. Copy your license file into the same directory as `moab.cfg` (`/opt/moab/etc/` by default). For example:

```
[root]# cp moab.lic $MOABHOMEDIR/etc/moab.lic
```

To verify the current status of your license, use `moab --about`.

Moab checks the status of the license every day just after midnight. At 60 and 45 days before, and daily from 30 days before license expiration to and including the license expiration date, Moab sends an e-mail to all level 1 administrators informing them of the pending Moab license expiration. A log record is also made of the upcoming expiration event. For the notifications to occur correctly, you must enable administrator email notification (see **Notifying Administrators of Failures** in the *Moab Workload Manager Administrator Guide*) and `moab.cfg` must contain email addresses for level 1 administrators. For example:

```
ADMINCFG [1] USERS=u1,u2,u3[,...]
USERCFG [u1] EMAILADDRESS=u1@company.com
USERCFG [u2] EMAILADDRESS=u2@company.com
USERCFG [u3] EMAILADDRESS=u3@company.com
MAILPROGRAM DEFAULT
```

**i** Moab will not run without a license. For information about obtaining a trial license, please contact [Adaptive Computing](#).

### 3. Start Moab (assumes Moab configured with the `--with-int` option).

- Red Hat 6-based systems

```
[root]# chkconfig moab on
[root]# service moab start
```

- Red Hat 7-based systems

```
[root]# systemctl enable moab.service
[root]# systemctl start moab.service
```

### 4. If you have a resource manager configured, submit a sleep job as a non-root user (adaptive is used in this example) and verify the job is running.

```
[root]# su - adaptive
[adaptive]$ echo sleep 150 | msub
[adaptive]$ showq
[adaptive]$ exit
```

### 5. Connecting Moab to MongoDB

If you will be installing Moab Web Services, connect Moab to MongoDB using the following instructions:

**i** The `USEDATABASE` parameter is unrelated to the MongoDB configuration.

- In `/opt/moab/etc/moab.cfg`, set the **MONGOSERVER** parameter to the correct location of the MongoDB server. By default, Moab assumes it is on the same server.

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

- b. In the `/opt/moab/etc/moab-private.cfg` file, set the **MONGOUSER** and **MONGOPASSWORD** parameters to the MongoDB `moab_user` credentials you set.

```
MONGOUSER      moab_user
MONGOPASSWORD  secret2
```

- c. Verify that Moab is able to connect to MongoDB.

- **Red Hat 6-based systems**

```
[root]# service moab restart
[root]# mdiag -S
...
Mongo connection (localhost) is up (credentials are set)
...
```

- **Red Hat 7-based systems**

```
[root]# systemctl restart moab.service
[root]# mdiag -S
...
Mongo connection (localhost) is up (credentials are set)
...
```

#### Related Topics

[Installing Moab HPC Suite RPM on page 62](#)

[Configuring TORQUE on page 68](#)

## Configuring Moab Web Services

This topic contains instructions on how to configure Moab Web Services (MWS).

In this topic:

- [Open Necessary Ports on page 73](#)
- [Configure Moab Web Services on page 74](#)

### Open Necessary Ports

Moab Web Services requires certain ports to be open for essential communication. For communication with the tomcat web server, the default port is 8080. For communication with the Mongo database, the default port is 27017.

If you have a firewall enabled, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the MWS server for communication with the tomcat web server
-A INPUT -p tcp --dport 8080 -j ACCEPT

# Needed on the Mongo server if installed on a separate host from MWS
-A INPUT -p tcp --dport 27017 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=8080/tcp --permanent
[root]# firewall-cmd --add-port=27017/tcp --permanent
[root]# firewall-cmd --reload
```

## Configure Moab Web Services

### 1. Start Moab.

- Red Hat 6-based systems

```
[root]# service moab start
```

- Red Hat 7-based systems

```
[root]# systemctl start moab.service
```

### 2. Set up the MWS configuration file.

#### a. In the `/opt/mws/etc/mws-config.groovy` file, change these settings:

- **auth.defaultUser.username:** Any value you like, or leave as is.
- **auth.defaultUser.password:** Any value you like, but choose a strong password.

```
[root]# vi /opt/mws/etc/mws-config.groovy
```

```
// Change these to be whatever you like.
auth.defaultUser.username = "moab-admin"
auth.defaultUser.password = "changeme!"
```



If you do not change **auth.defaultUser.password**, your MWS will not be secure (because anyone reading these instructions would be able to log into your MWS). See [tips](http://www.us-cert.gov/cas/tips/ST04-002.html) (<http://www.us-cert.gov/cas/tips/ST04-002.html>) for choosing a good password.

b. Do *one* of the following:

- If you are configuring an MWS connection to your LDAP server, add the following parameters:

```
ldap.server = "192.168.0.5"
ldap.port = 389
ldap.baseDNs = ["dc=acme,dc=com"]
ldap.bindUser = "cn=Manager,dc=acme,dc=com"
ldap.password = "*****"
ldap.directory.type = "OpenLDAP Using InetOrgPerson Schema"
```

*This is just an example LDAP connection. Be sure to use the appropriate domain controllers (dc) and common names (cn) for your environment.*

**i** If you followed the Adaptive Computing tutorial, [Setting Up OpenLDAP on CentOS 6 on page 78](#), your `ldap.directory.type` should be set to "OpenLDAP Using InetOrgPerson Schema." However, the use of other schemas is supported. For more information see **LDAP Configuration Using mws-config.groovy** in the *Moab Web Services Reference Guide*.

**i** To see how to configure a secure connection to the LDAP server, see **Securing the LDAP Connection** in the *Moab Web Services Reference Guide*.

- If you are configuring MWS to use PAM, add the the `pam.configuration.service` parameter to the `mws-config.groovy` file. For example:

```
pam.configuration.service = "login"
```

*This is just an example PAM configuration file name. Make sure you specify the name of the configuration file you want MWS to use.*

**i** For more information about PAM configuration with MWS, see **PAM (Pluggable Authentication Module) Configuration Using mws-config.groovy** in the *Moab Web Services Reference Guide*.

**!** There is a security risk when authenticating local users through your PAM configuration. This behavior is highly discouraged and not supported by Adaptive Computing.

**i** You can configure only one authentication method in `mws-config.groovy`—LDAP or PAM, but not both. If you have configured both LDAP and PAM, MWS defaults to using LDAP.

If you need multiple authentication methods, you must add them to your local PAM configuration. See your distribution documentation for details.

- c. Add the `grails.mongo.username` and `grails.mongo.password` parameters to the `mws-config.groovy` file. Use the MWS credentials you added to MongoDB in the [Installing Moab HPC Suite RPM on page 62](#).

```
...
grails.mongo.username = "mws_user"
grails.mongo.password = "secret3"
```

### 3. Start or restart Tomcat.

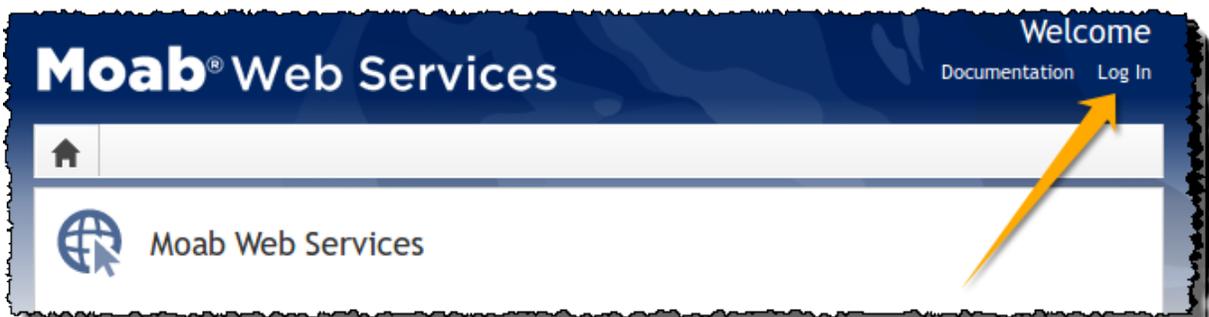
- Red Hat 6-based systems

```
[root]# chkconfig tomcat on
[root]# service tomcat restart
```

- Red Hat 7-based systems

```
[root]# systemctl enable tomcat.service
[root]# systemctl restart tomcat.service
```

4. Navigate to `http://<localhost>:8080/mws/` in a web browser to verify that MWS is running (you will see some sample queries and a few other actions).
5. Log in to MWS to verify that your credentials are working. (Your login credentials are the `auth.defaultUser.username` and `auth.defaultUser.password` values you set in the `/opt/mws/etc/mws-config.groovy` file.)



**i** If you encounter problems, or if the application does not seem to be running, see the steps in [Moab Web Services Issues on page 109](#).

## Related Topics

- [Installing Moab HPC Suite RPM on page 62](#)

## Additional Configuration

In this section:

- [Configuring SSL in Tomcat on page 78](#)
- [Setting Up OpenLDAP on CentOS 6 on page 78](#)
- [Trusting Servers in Java on page 85](#)

### Configuring SSL in Tomcat

To configure SSL in Tomcat, please refer to the [Apache Tomcat documentation](#).

### Setting Up OpenLDAP on CentOS 6

These instructions are intended to help first-time LDAP administrators get up and running. The following procedures contain instructions for getting started using OpenLDAP on a CentOS 6 system. For more complete information on how to set up OpenLDAP see the [OpenLDAP documentation](http://www.openldap.org/doc/admin24/) (<http://www.openldap.org/doc/admin24/>).

In this topic:

- [Installing and Configuring OpenLDAP on Centos 6 on page 78](#)
- [Adding an Organizational Unit \(OU\) on page 82](#)
- [Adding a User on page 83](#)
- [Adding a Group on page 83](#)
- [Adding a User to a Group on page 84](#)

**i** Adaptive Computing is not responsible for creating, maintaining, or supporting customer LDAP or Active Directory configurations.

### Installing and Configuring OpenLDAP on Centos 6

First, you will need to install OpenLDAP. These instructions explain how you can do this on a CentOS 6 system.

To install and configure OpenLDAP on Centos 6

1. Run the following command:

```
[root]# yum -y install openldap openldap-clients openldap-servers
```

2. Generate a password hash to be used as the admin password. This password hash will be used when you create the root user for your LDAP installation. For example:

```
[root]# slappasswd
New password : p@ssw0rd
Re-enter new password : p@ssw0rd
{SSHA}51PFVw19zeh7LT53hQH69znzj8TuBrLv
```

3. Add the root user and the root user's password hash to the OpenLDAP configuration in the `olcDatabase={2}bdb.ldif` file. The root user will have permissions to add other users, groups, organizational units, etc. Do the following:

- a. Run this command:

```
[root]# cd /etc/openldap/slapd.d/cn\=config
[root]# vi olcDatabase\=\{2\}bdb.ldif
```

- b. If the **olcRootPW** attribute does not already exist, create it. Then set the value to be the hash you created from `slappasswd`. For example:

```
olcRootPW: {SSHA}51PFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

4. While editing this file, change the distinguished name (DN) of the **olcSuffix** to something appropriate. The suffix typically corresponds to your DNS domain name, and it will be appended to the DN of every other LDAP entry in your LDAP tree.

For example, let's say your company is called Acme Corporation, and that your domain name is "acme.com." You might make the following changes to the `olcDatabase={2}bdb.ldif` file:

```
olcSuffix: dc=acme,dc=com
...
olcRootDN: cn=Manager,dc=acme,dc=com
...
olcRootPW: {SSHA}51PFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

**i** Throughout the following examples in this topic, you will see `dc=acme,dc=com`. "acme" is only used as an example to illustrate what you would use as your own domain controller if your domain name was "acme.com." You should replace any references to "acme" with your own organization's domain name.

**!** Do not set the cn of your root user to "root" (`cn=root,dc=acme,dc=com`), or OpenLDAP will have problems.

5. Modify the DN of the root user in the `olcDatabase={1}monitor.ldif` file to match the **olcRootDN** line in the `olcDatabase={2}bdb.ldif` file. Do the following:

- a. Run this command to edit the `olcDatabase={2}bdb.ldif` file:

```
[root]# vi olcDatabase=\{1\}monitor.ldif
```

- b. Modify the **olcAccess** line so that the **dn.base** matches the **olcRootDN** from the `olcDatabase={2}bdb.ldif` file. (In this example, **dn.base** should be `"cn=Manager,dc=acme,dc=com"`.)

```
olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by
dn.base="cn=Manager,dc=acme,dc=com" read by * none
```

- c. Now the root user for your LDAP is `cn=Manager,dc=acme,dc=com`. The root user's password is the password that you entered using `slappasswd` (see step 2), which, in this example, is **p@ssw0rd**
6. Hide the password hashes from users who should not have permission to view them.

**i** A full discussion on configuring access control in OpenLDAP is beyond the scope of this tutorial. For help, see [the OpenLDAP Access Control documentation](http://www.openldap.org/doc/admin24/access-control.html) (<http://www.openldap.org/doc/admin24/access-control.html>).

- a. Run this command to edit the `olcDatabase=\{2\}bdb.ldif` file:

```
[root]# vi olcDatabase=\{2\}bdb.ldif
```

- b. Add the following two lines to the end of the file to restrict users from viewing other users' password hashes.

```
olcAccess: {0}to attrs=userPassword by self write by
dn.base="cn=Manager,dc=acme,dc=com" write by anonymous auth by * none
olcAccess: {1}to * by dn.base="cn=Manager,dc=acme,dc=com" write by self write by
* read
```

*These lines allow a user to read and write his or her own password. It also allows a manager to read and write anyone's password. Anyone, including anonymous users, is allowed to view non-password attributes of other users.*

7. Make sure that OpenLDAP is configured to start when the machine starts up, and start the OpenLDAP service.

```
[root]# chkconfig slapd on
[root]# service slapd start
```

8. Now, you must manually create the `"dc=acme,dc=com"` LDAP entry in your LDAP tree.

An LDAP directory is analogous to a tree. Nodes in this tree are called LDAP "entries" and may represent users, groups, organizational units, domain controllers, or other objects. The attributes in each entry are determined by the LDAP schema. In this tutorial we will build entries based on the InetOrgPerson schema (which ships with OpenLDAP by default).

In order to build our LDAP tree we must first create the root entry. Root entries are usually a special type of entry called a domain controller (DC). Because we are assuming that the organization is called Acme Corporation, and that the domain is "acme.com," we will create a domain controller LDAP entry called `dc=acme,dc=com`. Again, you will need to replace "acme" with your organization's domain name. Also note that `dc=acme,dc=com` is what is called an LDAP distinguished name (DN). An LDAP distinguished name uniquely identifies an LDAP entry.

Do the following:

- a. Create a file called `acme.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi acme.ldif
```

- b. Add the following lines in `acme.ldif`:

```
dn: dc=acme,dc=com
objectClass: dcObject
objectClass: organization
dc: acme
o : acme
```

- c. Now add the contents of this file to LDAP. Run this command:

```
[root]# ldapadd -f acme.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

- d. Verify that your entry was added correctly.

```
[root]# ldapsearch -x -LLL -b dc=acme,dc=com
dn: dc=acme,dc=com
objectClass: dcObject
objectClass: organization
dc: acme
o: acme
```

9. Run the following:

```
[root]# sudo iptables -L
[root]# sudo service iptables save
```

10. By default, the CentOS 6 firewall will block external requests to OpenLDAP. In order to allow MWS to access LDAP, you will have to configure your firewall to allow connections on port 389. (Port 389 is the default LDAP port.)

Configuring your firewall is beyond the scope of this tutorial; however, it may be helpful to know that the default firewall on CentOS is a service called `iptables`. (For more information, see the documentation on [iptables](#).) In the most basic case, you may be able to add a rule to your firewall that accepts connections to port 389 by doing the following:

- a. Edit your `iptables` file:

```
[root]# vi /etc/sysconfig/iptables
```

- b. Add the following line *after* all the **ACCEPT** lines but *before* any of the **REJECT** lines in your `iptables` file:

```
# ... lines with ACCEPT should be above
-A INPUT -p tcp --dport 389 -j ACCEPT
# .. lines with REJECT should be below
```

For example, here is a sample `iptables` file with this line added:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -p tcp --dport 389 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT
```

- c. Now reload `iptables`.

```
[root]# service iptables reload
```

**i** Although providing instructions is beyond the scope of this tutorial, it is also highly recommended that you set up OpenLDAP to use SSL or TLS security to prevent passwords and other sensitive data from being sent in plain text. For information on how to do this, see the [OpenLDAP TLS documentation](http://www.openldap.org/doc/admin24/tls.html) (<http://www.openldap.org/doc/admin24/tls.html>).

Now that you have installed and set up Open LDAP, you are ready to add organizational units. See [Adding an Organizational Unit \(OU\) on page 82](#).

## Adding an Organizational Unit (OU)

These instructions will describe how to populate the LDAP tree with organizational units (OUs), groups, and users, all of which are different types of LDAP entries. The examples that follow also presume an `InetOrgPerson` schema, because the `InetOrgPerson` schema is delivered with OpenLDAP by default.

### To add an organizational unit (OU) entry to the LDAP tree

In this example, we are going to add an OU called "Users."

1. Create a temporary file called `users.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi users.ldif
```

2. Add these lines to `users.ldif`:

```
dn: ou=Users,dc=acme,dc=com
objectClass: organizationalUnit
ou: Users
```

3. Add the contents of `users.ldif` file to LDAP.

```
[root]# ldapadd -f users.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Adding a User

### To add a user to LDAP

In this example, we will add a user named "Bob Jones" to LDAP inside the "Users" OU.

1. Create a temporary file called `bob.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi bob.ldif
```

2. Add these lines to `bob.ldif`:

```
dn: cn=Bob Jones,ou=Users,dc=acme,dc=com
cn: Bob Jones
sn: Jones
objectClass: inetOrgPerson
userPassword: p@ssw0rd
uid: bjones
```

3. Add the contents of `bob.ldif` file to LDAP.

```
[root]# ldapadd -f bob.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Adding a Group

### To add a group to LDAP

In this example, we will add a group called "Engineering" to LDAP inside the "Users" OU.

1. Create a temporary file called `engineering.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi engineering.ldif
```

2. Add these lines to `engineering.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
cn: Engineering
objectClass: groupOfNames
member: cn=Bob Jones,ou=Users,dc=acme,dc=com
```

3. Add the contents of `engineering.ldif` file to LDAP.

```
[root]# ldapadd -f engineering.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Adding a User to a Group

### To add a user to an LDAP group

In this example, we will add an LDAP member named "Al Smith" to the "Engineering" LDAP group. This example assumes that user, Al Smith, has already been added to LDAP.

**i** Before you add a user to an LDAP group, the user must first be added to LDAP. For more information, see [Adding a User on page 83](#).

1. Create a temporary file called `addUserToGroup.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi addUserToGroup.ldif
```

2. Add these lines to `addUserToGroup.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
changetype: modify
add: member
member: cn=Al Smith,ou=Users,dc=acme,dc=com
```

3. Now add the contents of `addUserToGroup.ldif` file to LDAP.

```
[root]# ldapadd -f addUserToGroup.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

## Trusting Servers in Java

### Prerequisites

Some of these instructions refer to `JAVA_HOME`, which must point to the same directory that Tomcat uses. To set `JAVA_HOME`, do this:

```
[root]# source /etc/tomcat6/tomcat6.conf
```

Your system administrator might have defined Tomcat's `JAVA_HOME` in a different file.

### Retrieve the Server's X.509 Public Certificate

To retrieve the server's certificate, use the following command:

```
[root]# $JAVA_HOME/bin/keytool -printcert -rfc -sslserver <servername>:<port> >
/tmp/public.cert.pem
```

Replace `<servername>` with the server's host name and `<port>` with the secure port number. The default port for https is 443. The default port for ldaps is 636. If successful, `/tmp/public.cert.pem` contains the server's public certificate. Otherwise, `/tmp/public.cert.pem` contains an error message. This message is typical: `keytool error: java.lang.Exception: No certificate from the SSL server.` This message suggests that the server name or port is incorrect. Consult your IT department to determine the correct server name and port.

### Add the Server's Certificate to Java's Keystore

Java stores trusted certificates in a database known as the keystore. Because each new version of Java has its own keystore, you need to add the server certificate to the Java keystore (using the steps below) every time you install a new version of Java.

Java's keystore is located at `$JAVA_HOME/lib/security/cacerts`. If Tomcat's `JAVA_HOME` points to a JDK, then the keystore is located at `$JAVA_HOME/jre/lib/security/cacerts`. To add the server certificate to the keystore, run the following command:

```
[root]# $JAVA_HOME/bin/keytool -import -trustcacerts -file /tmp/public.cert.pem -alias
<servername> -keystore $JAVA_HOME/lib/security/cacerts
```

You will be prompted for the keystore password, which is "changeit" by default.

 Your system administrator might have changed this password.

After you've entered the keystore password, you'll see the description of the server's certificate. At the end of the description it prompts you to trust the certificate.

```
Trust this certificate? [no]:
```

Type `yes` and press **Enter** to add the certificate to the keystore.

# Upgrade

In this section:

- [Upgrading Moab HPC Suite from 8.0 or later on page 87](#)
- [Upgrading Moab HPC Suite from 7.2 on page 94](#)
- [Upgrading from MongoDB 2.0 to 2.4.x on page 101](#)

## Upgrading Moab HPC Suite from 8.0 or later

**i** Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Please note that the same commands will work for a non-root user with the `sudo` command.

### To Upgrade the RPM Suite

#### 1. Shut down all Adaptive services.

```
[root]# service moab stop           # you can also run mschedctl -k
[root]# service tomcat6 stop
[root]# service pbs_server stop
[root]# service pbs_mom stop        # if running pbs_mom on the head node
[root]# service trqauthd stop
```

#### 2. Download the latest 8.1.1 build executable (`moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz`, for example) from the [Adaptive Computing](#) website.

#### 3. Untar the package.

```
[root]# tar xzf moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz
```

#### 4. Change directories into the root directory of the untarred directory.

**i** Consider reviewing the README file for additional details on using the RPM distribution tarball.

#### 5. Install the suite repositories.

```
[root]# ./install-rpm-repos.sh [repository-directory] -y
```

**i** The `-y` option will install with the default settings for the RPM suite.

**i** The installation returns the following warning line:

```
Warning: RPMDB altered outside of yum.
```

This is normal and can safely be ignored.

The `[<repository-directory>]` option is the directory where you want to copy the RPMs. If no argument is given, `[<repository-directory>]` defaults to `/opt/adaptive-rpm-repository/rpm`. If the `[<repository-directory>]` already exists, RPMs will be added to the existing directory. No files are overwritten in `[<repository-directory>]`. A repository file is also created in `/etc/yum.repos.d/` and points to the `[<repository-directory>]` location.

For ease in repository maintenance, the install script fails if Adaptive Computing RPMs are copied to different directories. If a non-default `[<repository-directory>]` is specified, please use the same directory for future updates.

The script installs the `createrepo` package and its dependencies. You must answer "y" to all the questions in order for the RPM install to work. Additionally, the script installs the EPEL and 10gen repositories.

## 6. Merge the new `.repo` files in `/etc/yum.repos.d/` with the existing ones.

The `install-rpm-repos.sh` script will not overwrite existing RPM, GPG key or `.repo` files. Because some `.repo` files may have changed from previous releases, some merging of the `.repo` files is necessary. The newest files will have the `.new` extension.

Please compare older `.repo` files with the newer ones to ensure that the latest changes are reflected. In some cases, there is no change, and you can remove the new file. In most cases, however, it is safe to overwrite the old `.repo` file with the new one. For example:

```
[root]# mv /etc/yum.repos.d/AC.repo.new /etc/yum.repos.d/AC.repo
```

After making changes in the `/etc/yum.repos.d` directory, make sure to run the following command to update the yum cache:

```
[root]# yum clean all
```

## 7. Update the 8.1.1 suite packages.

```
[root]# yum update moab*
```

**i** The Moab and MWS RPMs automatically create a backup of all relevant files. These backups are stored in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`.

If changes are detected between any existing configuration files and new configuration files, a version of the new configuration file will be saved under `<configurationFileLocation>/<fileName>.rpmnew`.

8. If you use ODBC, you must upgrade to the 8.1.1 schema. See **Migrating Your Database to Newer Versions of Moab** in the *Moab Workload Manager Administrator Guide* for more information.
9. Adaptive Computing recommends MongoDB version 2.4.x. Support for environments using 2.0 is now deprecated and will be removed in a future release. If you are running a MongoDB version less than 2.4.x, see [Upgrading from MongoDB 2.0 to 2.4.x on page 101](#) for instructions.
10. Upgrade the schema of the `mws` database in MongoDB.

**!** You *must* perform this step, regardless of whether you upgraded MongoDB to version 2.4.x or not. (See previous step.)

**!** Before updating this database, you should perform a full backup. This can be done by using the `mongodump` utility documented in the [MongoDB documentation](#) (<http://www.mongodb.org/display/DOCS/Backups>).

Run the database migration script provided with MWS. (It is safe to run this script more than once. If for any reason, errors occur during the execution of the script, run it again.)

```
[root]# mongo -u mws_user mws /opt/mws/utils/db-migrate.js -p
```

**i** You may be prompted for the mongo password. The password can be found in the `/opt/mws/etc/mws-config.groovy` file under the "grails.mongo.password" key.

**i** Depending on the number of events and services in the system, the script may take several minutes to execute.

11. (Optional, but recommended for MWS) Upgrade to Java 7.

**i** Oracle Java 7 Runtime Environment is the recommended Java environment, but Java 6 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, etc. cannot run Moab Web Services.

**!** The Oracle® Java® download page has moved and requires a web-enabled workstation to accept the license agreement and download the software.

Do the following:

- a. Download Java 7 on a web-enabled workstation.
  - i. Open a web browser and connect to the [Java downloads page](http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html) (<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>).
  - ii. Select the radio button to accept the license agreement.
  - iii. Click the download link for the Linux x64 RPM file.
- b. Copy the Java 7 RPM to the MWS server using `scp`, `rsync`, or any similar network copy utility.
- c. On the MWS server, run the following to install Java 7:

```
[root]# rpm -Uh <RPMfilename>
```

## 12. Merge the configuration files.

**i** Whether or not to start with the old configuration file and add newer configuration options (or vice versa) depends on the amount of customization you previously made in earlier versions. In instances where you have modified very little, you should consider using the newer configuration and merging site-specific settings from the old file into the new one. The following steps highlight important changes between the 7.2.x default configuration and the 8.1.1 default configuration. Also note that new configuration files may have auto-generated content for secret keys and default passwords—be careful to ensure that secret keys shared between components are configured correctly.

**i** The recommended layout for the `/opt/moab/etc/` directory appears as follows:

```
[root]# ls -l /opt/moab/etc
total 29
-rw-r--r--. 1 root moab  2323 Nov 13 13:41 config.moab.pl
-rw-r--r--. 1 root moab   989 Nov 13 13:41 config.sql.pl
lrwxrwxrwx. 1 root root    14 Nov 13 15:46 moab.cfg -> moab.hpc.cfg
-rw-r--r--. 1 root moab 23500 Nov 13 15:43 moab.hpc.cfg
drwxr-xr-x. 2 root moab  4096 Nov 13 15:41 moab.d
-rw-r--r--. 1 root moab   391 Nov 13 13:41 moab.dat
-r--r--r--. 1 root root   493 Nov  6 16:14 moab.lic
-rw-----. 1 root moab   288 Nov 13 15:39 moab-private.cfg
lrwxrwxrwx. 1 root root    14 Nov 13 15:46 nami.cfg -> nami.hpc.cfg
-rw-r--r--. 1 root moab   563 Nov 13 15:43 nami.hpc.cfg
```

Do the following:

- a. Merge the `/opt/moab/etc/moab-private.cfg` file. Make sure that unique items in `/opt/moab/etc/moab-private.cfg.rpmnew` are added to the existing `/opt/moab/etc/moab-private.cfg` file. Include the new MWS RM credentials if you configure MWS as a resource manager:

```
CLIENTCFG[RM:mws] USERNAME=moab-admin PASSWORD=changeme!
```

**i** The default MWS credentials in 7.2.x were `admin:adminpw`. For releases after 7.2.x, the default credentials were changed to `moab-admin:changeme!`. Use whatever credentials you have configured in `/opt/mws/etc/mws-config.groovy`.

- b. Merge customizations from `/opt/moab/etc/moab.cfg` and `/opt/moab/etc/moab.d/*` into `/opt/moab/etc/moab.hpc.cfg`.

Although there are several ways to configure and merge changes into the `/opt/moab/etc/moab.cfg` file, the following instructions outline the recommended best practices. *Deviations from these best practices may result in unexpected behavior or added difficulty in future upgrades.*

It is best to use the new default configuration file (`/opt/moab/etc/moab.hpc.cfg`) and merge changes from previous files into that one. You will notice that content from the `/opt/moab/etc/moab.d/` directory has been merged into `/opt/moab/etc/moab.hpc.cfg`. Ensure that custom configuration options in all files located in `/opt/moab/etc/moab.d/` directory get merged in to `/opt/moab/etc/moab.hpc.cfg`.

You should avoid `#include` configurations.

Although the upgrade should have created a backup of the `moab.cfg` file (in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`), it is best to

create your own backup until you can confirm the updated configuration behaves as expected.

```
[root]# cp /opt/soab/etc/soab.cfg /opt/soab/etc/soab.cfg.bak
```

c. Merge the `/opt/mws/etc/mws-config.groovy` file.

Merge the `/opt/mws/etc/mws-config.groovy.rpmnew` file with the old `/opt/mws/etc/mws-config.groovy` file by editing `/opt/mws/etc/mws-config.groovy`. (Note the addition of the "auditAppender" in the default logging configuration of `/opt/mws/etc/mws-config.groovy.rpmnew`.)



Note that the **mws.suite** parameter and the **mam.\*** parameters have been moved to a suite-specific file in `/opt/mws/etc/mws.d/` and do not need to exist in `/opt/mws/etc/mws-config.groovy`.

Also note the new **\*messageQueue** parameters in `/opt/mws/etc/mws-config.groovy.rpmnew`. These are required and the **moab.messageQueue.secretKey** value should match the value located in `/opt/moab/etc/moab-private.cfg`.

### 13. Start all Adaptive services.

```
[root]# service pbs_server start
[root]# service moab start
[root]# service tomcat6 start
[root]# service pbs_mom start      # if running pbs_mom on the head node
[root]# service trqauthd start
```

## Upgrading Moab HPC Suite from 7.2

**⚠** If using Moab Web Services, this upgrade removes all MWS roles and permissions and recreates the default roles. If you have modified any MWS permissions or roles, you will need to recreate them after the upgrade is complete.

**i** Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Please note that the same commands will work for a non-root user with the `sudo` command.

### To Upgrade the RPM Suite

#### 1. Shut down all Adaptive services.

```
[root]# service moab stop      # you can also run mschedctl -k
[root]# service tomcat6 stop
[root]# service pbs_server stop
[root]# service pbs_mom stop   # if running pbs_mom on the head node
[root]# service trqauthd stop
```

#### 2. Download the latest 8.1.1 build executable (`moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz`, for example) from the [Adaptive Computing](#) website.

#### 3. Untar the package.

```
[root]# tar xzf moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz
```

## 4. Change directories into the root directory of the untarred directory.

**i** Consider reviewing the README file for additional details on using the RPM distribution tarball.

## 5. Install the suite repositories.

```
[root]# ./install-rpm-repos.sh [repository-directory] -y
```

**i** The `-y` option will install with the default settings for the RPM suite.

**i** The installation returns the following warning line:

```
Warning: RPMDB altered outside of yum.
```

This is normal and can safely be ignored.

The `[<repository-directory>]` option is the directory where you want to copy the RPMs. If no argument is given, `[<repository-directory>]` defaults to `/opt/adaptive-rpm-repository/rpm`. If the `[<repository-directory>]` already exists, RPMs will be added to the existing directory. No files are overwritten in `[<repository-directory>]`. A repository file is also created in `/etc/yum.repos.d/` and points to the `[<repository-directory>]` location.

For ease in repository maintenance, the install script fails if Adaptive Computing RPMs are copied to different directories. If a non-default `[<repository-directory>]` is specified, please use the same directory for future updates.

The script installs the `createrepo` package and its dependencies. You must answer "y" to all the questions in order for the RPM install to work. Additionally, the script installs the EPEL and 10gen repositories.

6. Merge the new `.repo` files in `/etc/yum.repos.d/` with the existing ones.

The `install-rpm-repos.sh` script will not overwrite existing RPM, GPG key or `.repo` files. Because some `.repo` files may have changed from previous releases, some merging of the `.repo` files is necessary. The newest files will have the `.new` extension.

Please compare older `.repo` files with the newer ones to ensure that the latest changes are reflected. In some cases, there is no change, and you can remove the new file. In most cases, however, it is safe to overwrite the old `.repo` file with the new one. For example:

```
[root]# mv /etc/yum.repos.d/AC.repo.new /etc/yum.repos.d/AC.repo
```

After making changes in the `/etc/yum.repos.d` directory, make sure to run the following command to update the `yum` cache:

```
[root]# yum clean all
```

## 7. Update the 8.1.1 suite packages.

```
[root]# yum update moab*
```

**i** The Moab and MWS RPMs automatically create a backup of all relevant files. These backups are stored in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`.

If changes are detected between any existing configuration files and new configuration files, a version of the new configuration file will be saved under `<configurationFileLocation>/<fileName>.rpmnew`.

8. If you use ODBC, you must upgrade to the 8.1.1 schema. See **Migrating Your Database to Newer Versions of Moab** in the *Moab Workload Manager Administrator Guide* for more information.
9. Adaptive Computing recommends MongoDB version 2.4.x. Support for environments using 2.0 is now deprecated and will be removed in a future release. If you are running a MongoDB version less than 2.4.x, see [Upgrading from MongoDB 2.0 to 2.4.x on page 101](#) for instructions.
10. Upgrade the schema of the `mws` database in MongoDB.

**!** You *must* perform this step, regardless of whether you upgraded MongoDB to version 2.4.x or not. (See previous step.)

**!** Before updating this database, you should perform a full backup. This can be done by using the `mongodump` utility documented in the [MongoDB documentation](#) (<http://www.mongodb.org/display/DOCS/Backups>).

Run the database migration script provided with MWS. (It is safe to run this script more than once. If for any reason, errors occur during the execution of the script, run it again.)

```
[root]# mongo -u mws_user mws /opt/mws/utils/db-migrate.js -p
```

**i** You may be prompted for the mongo password. The password can be found in the `/opt/mws/etc/mws-config.groovy` file under the "grails.mongo.password" key.

**i** Depending on the number of events and services in the system, the script may take several minutes to execute.

11. (Optional, but recommended for MWS) Upgrade to Java 7.

**i** Oracle Java 7 Runtime Environment is the recommended Java environment, but Java 6 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, etc. cannot run Moab Web Services.

**!** The Oracle® Java® download page has moved and requires a web-enabled workstation to accept the license agreement and download the software.

Do the following:

- a. Download Java 7 on a web-enabled workstation.
  - i. Open a web browser and connect to the [Java downloads page](http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html) (<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>).
  - ii. Select the radio button to accept the license agreement.
  - iii. Click the download link for the Linux x64 RPM file.
- b. Copy the Java 7 RPM to the MWS server using scp, rsync, or any similar network copy utility.
- c. On the MWS server, run the following to install Java 7:

```
[root]# rpm -Uh <RPMfilename>
```

## 12. Merge the configuration files.

**i** Whether or not to start with the old configuration file and add newer configuration options (or vice versa) depends on the amount of customization you previously made in earlier versions. In instances where you have modified very little, you should consider using the newer configuration and merging site-specific settings from the old file into the new one. The following steps highlight important changes between the 7.2.x default configuration and the 8.1.1 default configuration. Also note that new configuration files may have auto-generated content for secret keys and default passwords—be careful to ensure that secret keys shared between components are configured correctly.

**i** The recommended layout for the `/opt/moab/etc/` directory appears as follows:

```
[root]# ls -l /opt/moab/etc
total 29
-rw-r--r--. 1 root moab 2323 Nov 13 13:41 config.moab.pl
-rw-r--r--. 1 root moab 989 Nov 13 13:41 config.sql.pl
lrwxrwxrwx. 1 root root 14 Nov 13 15:46 moab.cfg -> moab.hpc.cfg
-rw-r--r--. 1 root moab 23500 Nov 13 15:43 moab.hpc.cfg
drwxr-xr-x. 2 root moab 4096 Nov 13 15:41 moab.d
-rw-r--r--. 1 root moab 391 Nov 13 13:41 moab.dat
-r--r--r--. 1 root root 493 Nov 6 16:14 moab.lic
-rw-----. 1 root moab 288 Nov 13 15:39 moab-private.cfg
lrwxrwxrwx. 1 root root 14 Nov 13 15:46 nami.cfg -> nami.hpc.cfg
-rw-r--r--. 1 root moab 563 Nov 13 15:43 nami.hpc.cfg
```

Do the following:

- a. Merge the `/opt/moab/etc/moab-private.cfg` file. Make sure that unique items in `/opt/moab/etc/moab-private.cfg.rpmnew` are added to the existing `/opt/moab/etc/moab-private.cfg` file. Include the new MWS RM credentials if you configure MWS as a resource manager:

```
CLIENTCFG[RM:mws] USERNAME=moab-admin PASSWORD=changeme!
```

**i** The default MWS credentials in 7.2.x were `admin:adminpw`. For releases after 7.2.x, the default credentials were changed to `moab-admin:changeme!`. Use whatever credentials you have configured in `/opt/mws/etc/mws-config.groovy`.

- b. Merge customizations from `/opt/moab/etc/moab.cfg` and `/opt/moab/etc/moab.d/*` into `/opt/moab/etc/moab.hpc.cfg`.

Although there are several ways to configure and merge changes into the `/opt/moab/etc/moab.cfg` file, the following instructions outline the recommended best practices. *Deviations from these best practices may result in unexpected behavior or added difficulty in future upgrades.*

It is best to use the new default configuration file (`/opt/moab/etc/moab.hpc.cfg`) and merge changes from previous files into that one. You will notice that content from the `/opt/moab/etc/moab.d/` directory has been merged into `/opt/moab/etc/moab.hpc.cfg`. Ensure that custom configuration options in all files located in `/opt/moab/etc/moab.d/` directory get merged in to `/opt/moab/etc/moab.hpc.cfg`.

You should avoid `#include` configurations.

Although the upgrade should have created a backup of the `moab.cfg` file (in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`), it is best to

create your own backup until you can confirm the updated configuration behaves as expected.

```
[root]# mv /opt/moab/etc/moab.cfg /opt/moab/etc/moab.hpc.cfg.bak
```

Once the changes have been merged to `/opt/moab/etc/moab.hpc.cfg`, configure Moab to use the new file. The recommended configuration is to use a symlink called `/opt/moab/etc/moab.cfg` that points to `/opt/moab/etc/moab.hpc.cfg`.

```
[root]# ln -s /opt/moab/etc/moab.hpc.cfg /opt/moab/etc/moab.cfg
```

c. Merge the `/opt/mws/etc/mws-config.groovy` file.

Merge the `/opt/mws/etc/mws-config.groovy.rpmnew` file with the old `/opt/mws/etc/mws-config.groovy` file by editing `/opt/mws/etc/mws-config.groovy`. (Note the addition of the "auditAppender" in the default logging configuration of `/opt/mws/etc/mws-config.groovy.rpmnew`.)

```

moab.messageQueue.port = 5570

log4j = {
  // Configure an appender for the events log.
  def eventAppender = new org.apache.log4j.rolling.RollingFileAppender(
    name: 'events', layout: pattern(conversionPattern: "%m%n"))
  def rollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
    fileNamePattern: '/opt/mws/log/events.%d{yyyy-MM-dd}',
    activeFileName: '/opt/mws/log/events.log')
  rollingPolicy.activateOptions()
  eventAppender.setRollingPolicy(rollingPolicy)

  // Configure an appender for the audit log.
  def auditAppender = new org.apache.log4j.rolling.RollingFileAppender(
    name: 'audit',
    layout: new com.ace.mws.logging.ACPatternLayout("%j\t\t\t%c{1}
\t\t\t%m%n"))
  def auditRollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
    fileNamePattern: '/opt/mws/log/audit.%d{yyyy-MM-dd}',
    activeFileName: '/opt/mws/log/audit.log')
  auditRollingPolicy.activateOptions()
  auditAppender.setRollingPolicy(auditRollingPolicy)

  appenders {
    rollingFile name: 'stacktrace',
      file: '/opt/mws/log/stacktrace.log',
      maxFileSize: '100MB'
    rollingFile name: 'rootLog',
      file: '/opt/mws/log/mws.log',
      maxFileSize: '100MB', //The maximum file size for a single log file
      maxBackupIndex: 10, //Retain only the 10 most recent log files, delete
older logs to save space
      layout: pattern(conversionPattern: '%d %p %c %m%n'), //Configures the
output format of each log entry
      layout: new com.ace.mws.logging.ACPatternLayout(), //Configures the
output format of each log entry
      threshold: org.apache.log4j.Level.ERROR //Ignore any logging entries
less verbose than this threshold

    appender eventAppender
    appender auditAppender
  }

  // NOTE: This definition is a catch-all for any logger not defined below
  root {
    error 'rootLog'
  }

  // Individual logger configurations
  ...

  // Logs event information to the events log, not the rootLog
  trace additivity:false, events:'com.ace.mws.events.EventFlatFileWriter'
  // Logs audit information to the audit log, not the rootLog
  trace additivity:false, audit:'mws.audit'
}

```

*Additions are noted in red. Removed content is stricken out.*

Note that the **mws.suite** parameter and the **mam.\*** parameters have been moved to a suite-specific file in `/opt/mws/etc/mws.d/` and do not need to exist in `/opt/mws/etc/mws-config.groovy`.

Also note the new **\*messageQueue** parameters in `/opt/mws/etc/mws-config.groovy.rpmnew`. These are required and the **moab.messageQueue.secretKey** value should match the value located in `/opt/moab/etc/moab-private.cfg`.

### 13. Start all Adaptive services.

```
[root]# service pbs_server start
[root]# service moab start
[root]# service tomcat6 start
[root]# service pbs_mom start      # if running pbs_mom on the head node
[root]# service trqauthd start
```

## Upgrading from MongoDB 2.0 to 2.4.x

Adaptive Computing recommends MongoDB version 2.4.x. Support for environments using 2.0 is now deprecated and will be removed in a future release.

1. Verify you can connect to the Mongo database.
  - a. Obtain the Mongo username and password.

```
[root]# grep grails.mongo /opt/mws/etc/mws-config.groovy
grails.mongo.username = "mws_user"
grails.mongo.password = "secret3"
```

- b. Using the Mongo username and password (in our example, username is "mws\_user" and password is "secret3"), confirm you can log in.

```
[root]# service mongod start
[root]# mongo -u mws_user -p secret3 mws
MongoDB shell version: 2.4.12
connecting to: mws
> show collections
event
mongeez
pluginInstance
...
```

2. Refer to [mongodb.org](http://docs.mongodb.org/manual/release-notes/2.4-upgrade/) (<http://docs.mongodb.org/manual/release-notes/2.4-upgrade/>) for instructions on how to upgrade MongoDB. Note that you must pay close attention to the information regarding instances with auth enabled (as this is the recommended setup for Moab HPC Suite).

### 3. Remove version 2.0 and install 2.4

```
[root]# service mongod stop
[root]# yum remove mongo20-10gen-server mongo20-10gen
[root]# yum install mongo-10gen-server
[root]# service mongod start
```

**i** Note that the settings in the `/etc/mongod.conf` file were saved in `/etc/mongod.conf.rpmsave` while removing MongoDB 2.0. You may need to restore any custom settings after MongoDB 2.4.x is installed in the new `/etc/mongod.conf` file (for example, "auth = true").

4. After upgrading from 2.0 to 2.4.x, you should verify that the MongoDB credentials were preserved. Refer to step 1.

## Migrating the MAM Database from MySQL to PostgreSQL

PostgreSQL is now the preferred DBMS for MAM. Customers who have already installed MySQL as the DBMS for MAM are not required to migrate their database to use PostgreSQL at this time. However, MySQL is considered deprecated and new installations will only use PostgreSQL.

**i** PostgreSQL does not provide a standard procedure for migrating an existing database from MySQL to PostgreSQL. Adaptive Computing has had success using the `py-mysql2pgsql` tools for migrating/converting/exporting data from MySQL to PostgreSQL. See <https://github.com/philipsoutham/py-mysql2pgsql> for additional details.

### To Migrate the MAM Database

This procedure was successfully tested on an actual customer MySQL database with millions of transactions on CentOS 6.4. It completed in less than an hour.

1. Make a backup copy of your MySQL mam database.

```
[root]# mysqldump mam > /archive/mam.mysql
```

2. Follow the instructions to Install PostgreSQL in the *Moab HPC Suite Installation and Configuration Guide*.
  - Manual Install - [Preparing for Manual Installation on page 12](#)
  - RPM Install - [Installing Moab HPC Suite RPM on page 62](#)
3. Install the prerequisite packages.

```
[root]# yum install git postgresql-devel gcc MySQL-python python-psycopg2 PyYAML
termcolor python-devel
```

#### 4. Install pg-mysql2pgsql (from source).

```
[root]# cd /software
[root]# git clone git://github.com/philipsoutham/py-mysql2pgsql.git
[root]# cd py-mysql2pgsql
[root]# python setup.py install
```

#### 5. Run pg-mysql2pgsql once to create a template yaml config file.

```
[root]# py-mysql2pgsql -v
```

#### 6. Edit the config file to specify the MySQL database connection information and a file to output the result.

```
[root]# vi mysql2pgsql.yml
```

```
mysql:
hostname: localhost
port: 3306
socket:
username: mam
password: changeme
database: mam
compress: false
destination:
# if file is given, output goes to file, else postgres
file: /archive/mam.pgsql
postgres:
hostname: localhost
port: 5432
username:
password:
database:
```

#### 7. Run the pg-mysql2pgsql program again to convert the database.

```
[root]# py-mysql2pgsql -v
```

#### 8. Create the mam database in PostgreSQL.

```
[root]# su - postgres
[postgres]$ psql
postgres=# create database "mam";
postgres=# create user mam with password 'changeme!';
postgres=# \q
[postgres]$ exit
```

#### 9. Import the converted data into the PostgreSQL database.

```
[root]# su - mam
[mam]$ psql mam < /archive/mam.pgsql
```

#### 10. Point MAM to use the new postgresql database.

```
[mam]$ cd /software/mam-latest
[mam]$ ./configure                # This will generate an etc/goldd.conf.dist
file
[mam]$ vi /opt/mam/etc/goldd.conf # Merge in the database.datasource from
etc/goldd.conf.dist
```

## 11. Restart Moab Accounting Manager.

```
[mam]$ goldd -r
```

## Chapter 4 Troubleshooting

This page details some common problems and general solutions. It contains these sections:

- [General Issues on page 105](#)
- [Moab Web Services Issues on page 109](#)

### General Issues

- [Moab error: "cannot determine local hostname" on page 105](#)
- [Moab error: "Moab will now exit due to license file not found" on page 106](#)
- [Other Moab issues on page 106](#)
- [Where do I change my passwords? on page 107](#)

#### Moab error: "cannot determine local hostname"

```
# service moab start
Starting moab: ERROR:    cannot determine local hostname - node is misconfigured
                        [FAILED]
```

If you encounter this error when starting Moab, check the `/opt/moab/etc/moab.cfg` file to make sure a valid host is configured. For example:

```
...
SCHEDCFG[Moab]                SERVER=<moab-hostname>:42559
...
```

Also check `/etc/hosts` to be sure the host name resolves, at least with `localhost`:

```
...
127.0.0.1    <moab-hostname> localhost localhost.localdomain localhost4
localhost4.localdomain4
...
```

### Moab error: "Moab will now exit due to license file not found"

```
# service moab start
Starting moab: Moab will now exit due to license file not found
Please contact Adaptive Computing (sales@adaptivecomputing.com) to get a license
for your system
[FAILED]
```

If you encounter this error when starting Moab, make sure your Moab license file is named **moab.lic** and is located in the `/opt/moab/etc/` directory.

Also make sure the license is not expired. The expiration date is listed in the license file. For example:

```
# cat /opt/moab/etc/moab.lic
...
# Expires after Tue Dec 31 10:43:46 2013
...
```

### Other Moab issues

Please see **Troubleshooting and System Maintenance** in the *Moab Workload Manager Administrator Guide*.

Where do I change my passwords?

**MWS super user username and password**

The default username and password for MWS are **moab-admin** and **changeme!** (respectively).

To change the username and/or the password for the MWS super user:

1. Stop the `tomcat6` and `moab` services.

```
[root]# service moab stop
[root]# service tomcat6 stop
```

2. Change the respective values in the following files:

- `/opt/mws/etc/mws-config.groovy`:

```
auth.defaultUser.username = "moab-admin"
auth.defaultUser.password = "changeme!"
```

- `/opt/moab/etc/moab-private.cfg`:

```
CLIENTCFG[RM:mws] USERNAME=moab-admin PASSWORD=changeme!
```

- `/opt/moab/etc/cloud.cfg`:

```
CONFIG[default]      MWS_USERNAME=moab-admin
CONFIG[default]      MWS_PASSWORD=changeme!
```

3. Start the `tomcat6` service.

```
[root]# service tomcat6 start
```

4. Start the `moab` service.

```
[root]# service moab start
```

**MongoDB passwords**

To change the passwords for MongoDB:

1. Stop the `tomcat6` and `moab` services.

```
[root]# service moab stop
[root]# service tomcat6 stop
```

2. Change the passwords for the MongoDB accounts (i.e., **moab\_user** and/or **mws\_user**). For instructions, see the [MongoDB documentation](http://docs.mongodb.org/manual/tutorial/change-user-password/) (<http://docs.mongodb.org/manual/tutorial/change-user-password/>).

3. Edit the password values in the following files:

- `/opt/moab/etc/moab-private.cfg`:

```
MONGouser          moab_user
MONGOPASSWORD      secret2
```

- `/opt/mws/etc/mws-config.groovy`:

```
// MongoDB configuration.
grails.mongo.username = "mws_user"
grails.mongo.password = "secret3"
```

4. Start the `tomcat6` service.

```
[root]# service tomcat6 start
```

5. Start the `moab` service.

```
[root]# service moab start
```

## Moab Web Services Issues

If something goes wrong during the MWS installation, look in the following files:

- The MWS log file. By default this is `/opt/mws/log/mws.log`.
- The Tomcat `catalina.out` file, usually in `/var/log/tomcat6` or `$CATALINA_HOME/logs`.

**i** If you remove the `log4j` configuration from `mws-config.groovy`, MWS writes its log files to `java.io.tmpdir`. For Tomcat, `java.io.tmpdir` is generally set to `$CATALINA_BASE/temp` or `CATALINA_TMPDIR`.

This section provides information on some common errors and their fixes.

- [MongoDB: Errors during MWS startup on page 110](#)
- [MongoDB: Out of semaphores to get db connection on page 113](#)
- [MongoDB: Connection wait timeout after 120000 ms on page 113](#)
- [java.lang.OutOfMemoryError: Java heap space on page 113](#)
- [java.lang.OutOfMemoryError: PermGen space on page 113](#)
- [SEVERE: Context \[/mws\] startup failed due to previous errors on page 114](#)
- [Moab Reached Maximum Number of Concurrent Client Connections on page 114](#)

MongoDB: Errors during MWS startup

If the application fails to start and gives error messages such as these:

```
Error creating bean with name 'mongoDatastore'
can't say something; nested exception is com.mongodb.MongoException
```

```
ERROR   grails.app.services.com.ace.mws.ErrorService    0
        Error encountered while attempting to authenticate account or query database;
        the MongoDB server is not available. Please verify connection to server
        '/127.0.0.1:27017' and that MongoDB is running.
```

MongoDB is most likely not running, or the MongoDB host and port are misconfigured.

In this case, there are a few things to verify:

- (Not relevant if MongoDB is installed on a separate server) **Is MongoDB installed?**

Run the following commands to assess whether MongoDB is installed on the current server.

```
$ mongo
-bash: mongo: command not found
```

To remedy, install MongoDB, start the `mongod` service and then restart the `tomcat6` service. See [Preparing for Manual Installation on page 12](#) or [Installing Moab HPC Suite RPM on page 62](#) for more information on how to install and configure MongoDB for Manual Installation or RPM Installation, respectively.

- (Only relevant if MongoDB is installed on a separate server) **Is MWS configured to connect to the remote MongoDB server?**

Run the following commands to assess whether MongoDB is installed on the current server.

```
[root]# cat /opt/mws/etc/mws-config.groovy | grep 'grails.mongo'
// grails.mongo.username = "mws_user"
// grails.mongo.password = "<ENTER-KEY-HERE>"
// grails.mongo.host = "127.0.0.1"
// grails.mongo.port = 27017
```

Make sure that the `grails.mongo.*` options are configured in `/opt/mws/etc/mws-config.groovy` for the remote MongoDB server and then restart the `tomcat6` service.

```
[root]# service tomcat6 restart
```

- **Is MWS configured to authenticate with MongoDB, and is MongoDB configured to enforce authentication?**

Run the following commands to assess the relevant MWS and MongoDB configurations.

```
[root]# cat /opt/mws/etc/mws-config.groovy | grep 'grails.mongo'  
// grails.mongo.username = "mws_user"  
// grails.mongo.password = "<ENTER-KEY-HERE>"  
  
[root]# cat /etc/mongod.conf | grep 'auth'  
#noauth = true  
auth = true
```

The configuration above is problematic because the `grails.mongo` credentials are commented out in the `/opt/mws/etc/mws-config.groovy` file while MongoDB is configured to enforce authentication ("`auth = true`"). Similar connection issues will exist if the `grails.mongo` parameters do not match the credentials configured for the "mws\_user" on both the `mws` and `moab` databases in MongoDB.

(For upgrade scenarios only) If the application fails to start and gives the following message in `/opt/mws/etc/log/mws.log`:

```
java.lang.Exception: The db-migrate.js script has not yet been run. Please see the  
upgrade section of the installation guide for instructions.
```

Then the `db-migrate.js` script must be run to update the schema of the `mws` database in MongoDB.

MongoDB: Out of semaphores to get db connection

To resolve this error, adjust the values of `connectionsPerHost` or `threadsAllowedToBlockForConnectionMultiplier` by adding them to `mws-config.groovy`. Example:

```
grails.mongo.options.connectionsPerHost = 60
grails.mongo.options.threadsAllowedToBlockForConnectionMultiplier = 10
```

For more information on these options, see:

- **Configuring Moab Web Services** in the *Moab Web Services Reference Guide* - Briefly discusses a few MongoDB driver options.
- [MongoOptions](http://api.mongodb.org/java/current/com/mongodb/MongoOptions.html) documentation (<http://api.mongodb.org/java/current/com/mongodb/MongoOptions.html>) - Contains full details on all MongoDB driver options.

**i** You must restart Tomcat after adding, removing, or changing **grails.mongo.options** parameters.

As shipped, `mws-config.groovy` does not contain any **grails.mongo.options** parameters. To adjust their values, you need to add them to `mws-config.groovy`.

The default value of **connectionsPerHost** is normally 10, but MWS sets it internally to 50.

The default value of **threadsAllowedToBlockForConnectionMultiplier** is 5.

Any of the options listed in `MongoOptions` can be specified in `mws-config.groovy`. Just use the prefix **grails.mongo.options** as shown above.

MongoDB: Connection wait timeout after 120000 ms

See [MongoDB: Out of semaphores to get db connection on page 113](#).

java.lang.OutOfMemoryError: Java heap space

Increase the size of the heap using JVM options **-Xms** and **-Xmx**. Here are the suggested values:

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m-Xmx3g -XX:MaxPermSize=384m"
```

- **-Xms**: Set initial Java heap size.
- **-Xmx**: Set maximum Java heap size.

java.lang.OutOfMemoryError: PermGen space

Increase the size of the permanent generation using JVM option **-XX:MaxPermSize**. Here are the suggested values:

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -XX:MaxPermSize=384m"
```

SEVERE: Context [/mws] startup failed due to previous errors

If `catalina.out` contains this error, look in `/opt/mws/log/mws.log` and `/opt/mws/log/stacktrace.log` for more details on the error.

Also ensure that the `/opt/mws/etc/mws-config.groovy` file can be read by the Tomcat user. The permissions should appear as follows:

```
$ ls -al /opt/mws/etc/mws-config.groovy
-r----- 1 tomcat tomcat 4056 Dec  4 12:07 mws-config.groovy
```

Moab Reached Maximum Number of Concurrent Client Connections

When this error message is encountered, simply add a new line to the `moab.cfg` file:

```
CLIENTMAXCONNECTIONS 256
```

This will change the Moab configuration when Moab is restarted. Run the following command to immediately use the new setting:

```
[root]# changeparam CLIENTMAXCONNECTIONS 256
```

**i** The number **256** above may be substituted for the desired maximum number of Moab client connections.

## Chapter 5 Component Documentation

The individual components of the suite have more options and allow for more configuration than can be contained in this guide. Refer to the individual component guides for more information.

### TORQUE

- TORQUE 5.1.1 Administrator Guide: [HTML](#)  - [PDF](#) 

### Moab Workload Manager

- Moab Workload Manager 8.1.1 Administrator Guide: [HTML](#)  - [PDF](#) 

### Moab Web Services

- Moab Web Services 8.1.1 Reference Guide: [HTML](#)  - [PDF](#) 

### Related Topics

[Requirements on page 3](#)

[Preparing for Manual Installation on page 12](#)

[Installing Moab HPC Suite RPM on page 62](#)

