

# Moab HPC Suite - Enterprise Edition

## Installation and Configuration Guide 9.0.0

October 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</b>	<b>1</b>
<b>Chapter 1 Planning Your Installation</b>	<b>3</b>
Suite Topology	4
Identify The Manual Or RPM Installation Methods	9
Component Requirements	10
<b>Chapter 2 Manual Installation</b>	<b>19</b>
Installation	20
Preparing For Manual Installation	20
Installing Torque Resource Manager	22
Installing Moab Workload Manager	29
Installing Moab Accounting Manager	35
Installing Moab Web Services	49
Installing Moab Insight	60
Installing Moab Viewpoint	60
Additional Configuration	61
Configuring SSL In Tomcat	61
Setting Up OpenLDAP On CentOS 6	61
Moab Workload Manager Configuration Options	68
Moab Accounting Manager Configuration Options	69
Trusting Servers In Java	71
Upgrade	73
Preparing For Upgrade	73
Upgrading Torque	74
Upgrading Moab Workload Manager	80
Upgrading Moab Accounting Manager	82
Upgrading Moab Web Services	87
Migrating The MAM Database From MySQL To PostgreSQL	93
<b>Chapter 3 RPM Installation</b>	<b>97</b>
Installation	98
Preparing The Host For RPM Installations	98
Installing Torque Resource Manager	100
Installing Moab Workload Manager	104
Installing Moab Accounting Manager	107
Installing Moab Web Services	118
Installing Moab Insight	127
Installing Moab Viewpoint	139
Additional Configuration	150
Configuring SSL In Tomcat	150
Setting Up OpenLDAP On CentOS 6	150
Trusting Servers In Java	157
Upgrade	159

Upgrading The Moab HPC Suite RPMs .....	159
Upgrading From MongoDB 2.0 To 2.4.x .....	168
Migrating The MAM Database From MySQL To PostgreSQL .....	169
<b>Chapter 4 Troubleshooting .....</b>	<b>173</b>
General Issues .....	173
Moab Web Services Issues .....	176
<b>Component Documentation .....</b>	<b>181</b>

## Welcome

Welcome to the Moab HPC Suite – Enterprise Edition 9.0.0 Installation and Configuration Guide, 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 – Enterprise Edition. The Moab HPC Suite – Enterprise Edition 9.0.0 contains the following components:

- Moab Workload Manager 9.0.0
- Moab Web Services 9.0.0
- Moab Accounting Manager 9.0.0
- Moab Viewpoint 9.0.0
- Moab Insight 9.0.0
- Torque 6.0.0

Before commencing the installation or upgrade, please see [Chapter 1 Planning your Installation on page 3](#) to verify your system conforms to minimum prerequisites.



# Chapter 1 Planning your Installation



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 the Moab HPC Suite. Each environment has its own set of requirements and preferences. This chapter is intended to help an administrator understand how each of the Moab HPC Suite components interact, basic requirements and configuration information to prepare for the installation.



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.

In this chapter:

- [Installation Terminology on page 3](#)
- [Where to Start on page 4](#)
- [Suite Topology on page 4](#)
- [Identify the Manual or RPM Installation Methods on page 9](#)
- [Component Requirements on page 10](#)

## Installation Terminology

To aid in documentation clarity, Adaptive Computing uses the following terms in this Installation and Configuration Guide:

- **Components** – The different "products" included in the Moab HPC Suite. For example, Moab Workload Manager, Moab Web Services.

- Servers – Also known as components, but specifically relating to the actual services. For example, the Moab Workload Manager component is referred to as the Moab Server for non-client services.
- Host – The actual box where an Moab HPC Suite component (server or client) is installed.

**i** Previous documentation typically used Head Node to designate a host or a Server.

## Where to Start

You will need to plan your environment and determine how many hosts you will need and for which you components you will install using the Manual Installation or the RPM Installation method. The following are suggested steps to help you in your planning and installing process.

1. Determine whether you have a small, medium, High-Throughput or large environment; including an example, and required and recommended hardware requirements. See [Suite Topology on page 4](#).
2. Decide whether you will perform a Manual Installation or an RPM Installation for the various components. See [Identify the Manual or RPM Installation Methods on page 9](#).

**i** The Manual Installation and the RPM Installation chapters each have an "Additional Configuration" section that provides additional information and instructions for optional, but recommended configurations (for example, Configuring SSL in Tomcat).

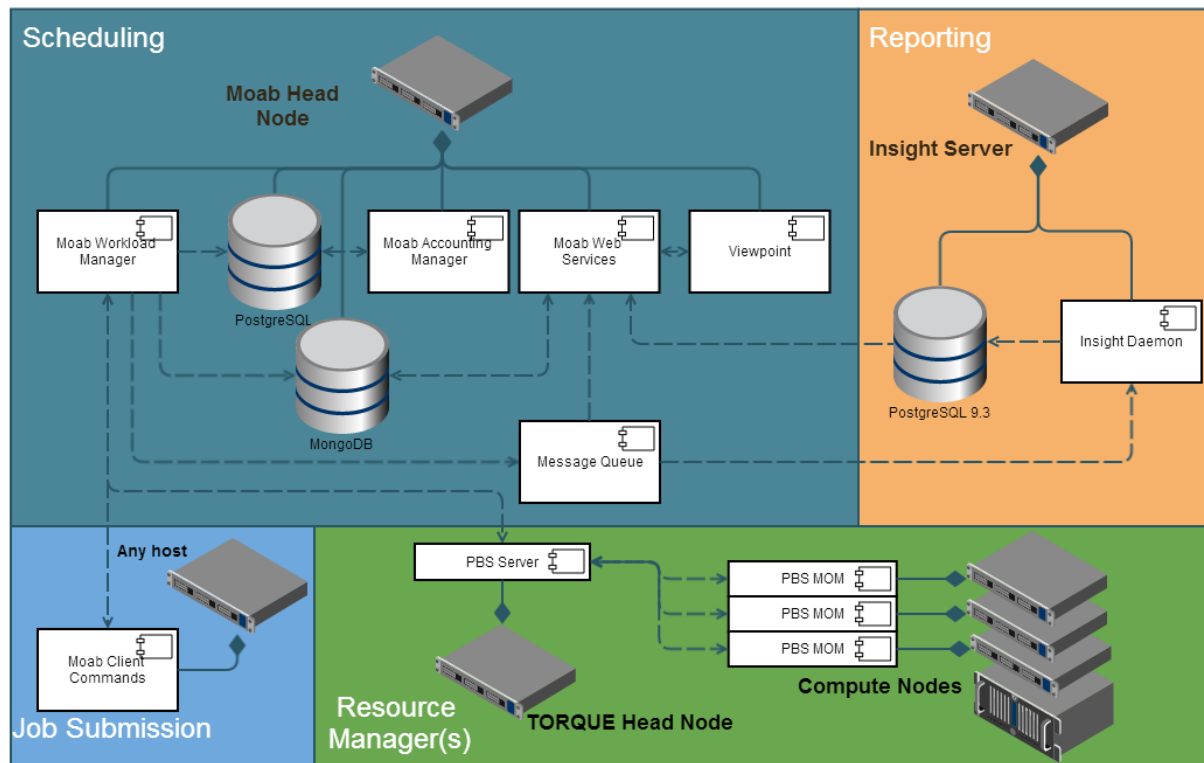
3. Review the software requirements for your components and set up your hosts accordingly. See [Component Requirements on page 10](#).
4. Install the individual components on their respective host(s). See [Preparing for Manual Installation on page 20](#) or [Preparing the Host for RPM Installations on page 98](#) as applicable.
5. Refer to [Chapter 4 Troubleshooting on page 173](#) for assistance in addressing common problems during installation and configuration.

## Suite Topology

The Moab HPC Suite is installed and configured differently for small, medium or large environment types. See [Environment Requirements on page 6](#).

The following diagram provides a general topology of the Moab HPC Suite for a medium (with high throughput) or a large environment.





Please note the following:

- Moab Accounting Manager is available only with the Moab HPC Enterprise Suite.
- Moab Insight is available only with the Moab HPC Enterprise Suite.
- Moab Viewpoint is available only with the Moab HPC Enterprise Suite.
- Smaller environments may elect to consolidate the Torque Server with the Moab Server on the same host, including PBS Server in the list of components installed on the same host.
- 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.
- Larger systems will require more dedicated resources for each component, in which case it may be necessary to move individual components from the Moab Host (i.e. databases, Moab Accounting Manager, and/or Viewpoint) 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.

## Environment Requirements

The following table identifies the minimum and recommended hardware requirements for the different environment types. Use this table as a guide when planing out your suite topology.

**i** Software requirements are listed per-component rather than suite-wide as the suite components reside on different hosts. See [Component Requirements on page 10](#)

Environment Type	# of Compute Nodes	Jobs/ Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Proof of Concept / Small Demo	50	<1k	<b>Moab+Torque Host</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> <b>Insight Host</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 256 GB dedicated disk space</li> </ul>	Same as minimum

Environment Type	# of Compute Nodes	Jobs/Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Medium	500	<100k	<b>Moab+Torque Host</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>Insight Host</b> <ul style="list-style-type: none"> <li>• 8 Intel/AMD x86-64 cores</li> <li>• At least 8 GB of RAM; a dedicated 1 Gbit channel between Insight and Moab</li> <li>• 128 GB local SSD for swapAt least 512 GB disk</li> </ul>	<b>Moab+Torque Host</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> <b>Insight Host</b> <ul style="list-style-type: none"> <li>• 8 Intel/AMD x86-64 cores</li> <li>• At least 8 GB of RAM; a dedicated 1 Gbit channel between Insight and Moab</li> <li>• 128 GB local SSD for swapAt least 512 GB disk</li> </ul>

Environment Type	# of Compute Nodes	Jobs/ Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Medium with High Throughput or Larger	>500	>100k	<b>Moab Host</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 Host</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>Insight Host</b> <ul style="list-style-type: none"> <li>8 Intel/AMD x86-64 cores</li> <li>At least 16 GB of RAM; a dedicated 1 Gbit channel between Insight and Moab</li> <li>128 GB local SSD for swap</li> <li>At least 512 GB disk</li> </ul>	<p>The Moab Server should <i>not</i> reside on the same host as the Torque Server.</p> <p>MWS Server <i>must</i> reside on the same host as the Moab Server (Moab Host).</p> <p>The MAM Server may reside on its own host, on the Moab Host (preferred), or another server's host (except for the Insight Host).</p> <p>The Viewpoint Server may reside on its own host, on the Moab Host (preferred), or another server's host (except for the Insight Host).</p> <p>Databases may also reside on the same or a different host than its server component.</p>

Please note the following:

- All requirements above (minimum and recommended) target a minimum number of management servers. Administrators are encouraged to separate the Torque Server and the Moab Server onto separate hosts 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.
- Due to the large amount of data Moab must send to Insight, Moab performs better without Insight enabled (for environments that do not require Viewpoint, or use Crystal Reporting).
- 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.

## Identify the Manual or RPM Installation Methods

Adaptive Computing provides two methods for installing the Moab HPC Suite components, Manual Installation and RPM Installation.

Depending on your environment and which components you are installing (and on which host), you may need to use a combination of Manual Installation and RPM Installation.

**i** Most components can be installed using either method. Please choose one method for each component.

### *Manual Installation*

This method provides advantages for administrators who want non-standard configure options.

- This method has more supported operating systems than the RPM Installation method.
- Insight or Viewpoint can not be installed using the Manual Installation method.

### *RPM Installation*

This method provides advantages for administrator who want a standard installation, with little customization.

- This method only supports CentOS, RHEL, or Scientific Linux operating systems.
- Whether you are installing RPMs on one host or on several hosts, each host must have the Adaptive Computing Package Repository enabled. See [Preparing the Host for RPM Installations on page 98](#)

# Component Requirements

This topic provides the various software requirements and dependencies for the suite components (servers).

In this topic:

- [Torque on page 10](#)
- [Moab Workload Manager on page 11](#)
- [Moab Accounting Manager on page 12](#)
- [Moab Web Services on page 13](#)
- [Moab Insight on page 14](#)
- [Moab Viewpoint on page 16](#)

## Torque



If you intend to use Torque 6.0.0 with Moab Workload Manager, you must run Moab version 9.0.0 or 8.0 or later. Torque 6.0.0 will not work with versions earlier than Moab 8.0.

In this section:

- [Supported Operating Systems on page 10](#)
- [Software Requirements on page 10](#)

## 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

- 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

- **cpusets and cgroups**
  - NUMA-awareness uses cgroups, which include cpusets. Red Hat systems must use libcgrouper version 0.40.rc1-16.el6 or later; SUSE systems need to use a comparative libcgrouper version.
  - cpusets: libhwloc 1.9.1 is the minimum supported, however NVIDIA K80 requires libhwloc 1.11.0. If you need to install libhwloc and the corresponding hwloc-devel package, see Linux Cpuset Support in the *Torque Resource Manager Administrator Guide*.

**i** If `--enable-cgroups` is specified, `--enable-cpuset` is ignored.

- if you build Torque from source (i.e. clone from github), the following additional software is required:
  - gcc
  - gcc-c++
  - posix-compatible version of make
  - libtool 1.5.22 or later
  - boost-devel 1.36.0 or later

**i** Red Hat 6-based systems come packaged with 1.41.0 and Red Hat 7-based systems 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

In this section:

- [Supported Operating Systems on page 11](#)
- [Software Requirements on page 12](#)
- [Supported Resource Managers on page 12](#)

### 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;



If installing MWS on the same host, SLES 11 is not supported. If your configuration includes Viewpoint, only Red Hat-based Oss are supported for the Moab Server and MWS Server shared host.

## Software Requirements

- 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.

In this topic:

- [Supported Operating Systems on page 12](#)
- [Software Requirements on page 13](#)
- [Depends On \(not necessarily on the same host\) on page 13](#)

## 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 9.0.0
- Perl modules; see [Installing Moab Accounting Manager on page 35](#) (Manual Installation) [Installing Moab Accounting Manager on page 107](#) (RPM Installation) for more details

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

MAM uses an RDBMS as a back end. Adaptive Computing recommends that the database used by MAM does *not* reside on the same host as the database used by Insight.

- PostgreSQL 7.2 or later

## Moab Web Services



MWS Server *must* reside same host as Moab Server (Moab Host). If using Viewpoint, the shared Moab Server and MWS Server must a Red-Hat based OS; regardless of whether Viewpoint is installed on that host.

In this topic:


- [Supported Operating Systems on page 13](#)
- [Software Requirements on page 14](#)
- [Depends On \(not necessarily on the same host\) on page 14](#)

## Supported Operating Systems

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

## Software Requirements

- Moab Workload Manager 9.0.0
- Oracle® Java® 8 Runtime Environment


 Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 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 host)

- OpenLDAP *or* PAM; see [Installing Moab Web Services on page 49](#) (Manual Installation) [Installing Moab Web Services on page 118](#) (RPM Installation) for more details

## Moab Insight

 Moab Workload Manager and Insight both tend to heavily consume system resources. The Insight Server and the Moab Server *must* run on separate hosts.

 Only an RPM-based installation is supported for installing Moab Viewpoint. A SLES OS is not supported.

In this section:

- [Supported Operating Systems on page 14](#)
- [Software Requirements on page 15](#)
- [Depends On \(not on the same host\) on page 15](#)
- [Performance Benchmarks on page 15](#)

## Supported Operating Systems

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x

## Software Requirements

- Oracle® Java® 8 Runtime Environment

**i** Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run Moab Web Services.

- PostgreSQL 9.3 or later
- MongoDB 2.4.x

## Depends On (not on the same host)

- Moab Workload Manager 9.0.0

## Performance Benchmarks

Adaptive Computing has tested and certified Insight's scale and performance under the following server configuration and load scenarios.

### Server Configuration

Host hardware: 8 core AMD Opteron 6320 2.8 GHz servers, with 32GB of ram and a 500GB WD Blue hard drive

Installed services: Moab Workload Manager, Moab Web Services, Moab Insight, Moab Viewpoint (all at version 9.0.0 and running on the same host)

**i** The benchmarks were ran with multiple services on a single host to benchmark Insight under very aggressive working conditions. Moab Insight must be installed on its own host.

### Load Scenarios

Jobs in queue	Avg Job Duration	Avg job Size (ppn)	Number of Nodes	Procs per Node	Avg Jobs per Week
1000	200	32	500	32	25200
1000	60	32	500	32	84000
1000	10	32	500	32	504000
1000	200	16	6384	16	321754

Jobs in queue	Avg Job Duration	Avg job Size (ppn)	Number of Nodes	Procs per Node	Avg Jobs per Week
1000	60	16	6384	16	1072512
1000	10	16	6384	16	6435072
10000	200	32	500	32	25200
10000	60	32	500	32	84000
10000	10	32	500	32	504000
10000	200	16	6384	16	321754
10000	60	16	6384	16	1072512
25000	200	32	500	32	25200
25000	60	32	500	32	84000
25000	10	32	500	32	504000

## Moab Viewpoint

**i** Only an RPM-based installation is supported for installing Moab Viewpoint. A SLES OS is not supported.

In this section:

- [Supported Operating Systems on page 16](#)
- [Software Requirements on page 17](#)
- [Depends On \(not necessarily on the same host\) on page 17](#)
- [Supported Browsers on page 17](#)

### Supported Operating Systems

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x

## Software Requirements

**i** The new user interface was built on Django, a forward-thinking web framework, which relies heavily on Python; thus, HPC administrators should install Viewpoint only on systems with standard system level Python installed. The system you select for Viewpoint should not have any modifications made to its default Python installation.

- httpd
- mod\_wsgi
- python-anyjson
- python-crypto2.6
- python-httplib2
- python-mako
- python-markupsafe
- python-paramiko
- python-pip
- python-requests
- python-simplejson
- python-six
- python-unittest2

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

- Moab Web Services 9.0.0
- Moab Insight 9.0.0

### Supported Browsers

- Mozilla Firefox 25+
- Internet Explorer 10+
- Chrome 35+



## Chapter 2 Manual Installation

This chapter provides installation, configuration, and upgrading information using the Manual Installation method.

**i** Manual Installation is not available for Insight or Viewpoint.

**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.

### Related Topics

- 
- [Preparing for Manual Installation on page 20](#)
- [Additional Configuration on page 61](#)
- [Upgrade on page 73](#)

# Installation

In this section:

- [Preparing for Manual Installation on page 20](#)
- [Installing Torque Resource Manager on page 22](#)
- [Installing Moab Workload Manager on page 29](#)
- [Installing Moab Accounting Manager on page 35](#)
- [Installing Moab Web Services on page 49](#)
- [Installing Moab Insight on page 60](#)
- [Installing Moab Viewpoint on page 60](#)

## 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 [Chapter 1 Planning your Installation 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.



## Set Up Proxies

If your site uses a proxy to connect to the internet, configure yum to use a proxy by editing the `/etc/yum.conf` file as follows:

```
proxy=http://<proxy_server_id>:<port>
```

## 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` and the `devel:languages:python` repos 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 addrepo
http://download.opensuse.org/repositories/devel:languages:python/SLE_
12/devel:languages:python.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 Resource Manager on page 22](#).
2. Install Moab Workload Manager. See [Installing Moab Workload Manager on page 29](#).
3. Install Moab Accounting Manager. See [Installing Moab Accounting Manager on page 35](#).
4. Install Moab Web Services. See [Installing Moab Web Services on page 49](#).

 Once you have finished installing the Moab HPC Suite and Moab is running, you can perform an RPM-based installation of the optional Insight and Viewpoint components.

## Installing Torque Resource Manager

 If you intend to use Torque Resource Manager 6.0.0 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque 6.0 functionality requires Moab 9.0 or later.

This topic contains instructions on how to install and start Torque Resource Manager (Torque).

In this topic:

- [Requirements on page 22](#)
- [Prerequisites on page 23](#)
- [Install Dependencies, Packages, or Clients on page 25](#)
- [Install Torque Server on page 25](#)
- [Install Torque MOMs on page 28](#)
- [Configure Data Management on page 29](#)

## Requirements

In this section:

- [Supported Operating Systems on page 23](#)
- [Software Requirements on page 23](#)

## 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

- 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
- cpusets and cgroups
  - NUMA-awareness uses cgroups, which include cpusets. Red Hat systems must use libcgroup version 0.40.rc1-16.el6 or later; SUSE systems need to use a comparative libcgroup version.
  - cpusets: libhwloc 1.9.1 is the minimum supported, however NVIDIA K80 requires libhwloc 1.11.0. If you need to install libhwloc and the corresponding hwloc-devel package, see Linux Cpuset Support in the *Torque Resource Manager Administrator Guide*.

**i** If using `--enable-cgroups` is specified, `--enable-cpuset` is ignored.

- if you build Torque from source (i.e. clone from github), the following additional software is required:
  - gcc
  - gcc-c++
  - posix-compatible version of make
  - libtool 1.5.22 or later
  - boost-devel 1.36.0 or later

**i** Red Hat 6-based systems come packaged with 1.41.0 and Red Hat 7-based systems 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* for more information.

## Prerequisites

In this section:

- [Open Necessary Ports on page 24](#)
- [Verify the hostname on page 25](#)

## 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* for more information.

If you have a firewall enabled, do the following:

### 1. On the Torque Server Host:

- Red Hat 6-based systems using iptables

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

# Add the following line immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"
-A INPUT -p tcp --dport 15001 -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 --reload
```

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

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

# Add the following port to the FW_SERVICES_EXT_TCP parameter
FW_SERVICES_EXT_TCP="15001"

[root]# service SuSEfirewall2_setup restart
```

## 2. On the Torque MOM Hosts (Compute Hosts):

- 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"
-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=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
FW_SERVICES_EXT_TCP="15002 15003"

[root]# service SuSEfirewall2_setup restart
```

## Verify the hostname

On the Torque Server Host, confirm 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, Packages, or Clients

### Install Packages

On the Torque Server Host, use the following commands to 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
```

## Install Torque Server

On the Torque Server Host, do the following:

1. Download the latest 6.0.0 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 6.0.0 6.0.0
[root]# cd 6.0.0
[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-6.0.0.tar.gz -O torque-6.0.0.tar.gz
[root]# tar -xzf torque-6.0.0.tar.gz
[root]# cd torque-6.0.0/
```

- SUSE-based systems

```
[root]# zypper install wget
[root]# wget http://www.adaptivecomputing.com/download/torque/torque-6.0.0.tar.gz -O torque-6.0.0.tar.gz
[root]# tar -xzf torque-6.0.0.tar.gz
[root]# cd torque-6.0.0/
```

2. Run each of the following commands in order.

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

See Customizing the Install in the *Torque Resource Manager Administrator Guide* for information on which options are available to customize the `./configure` command.

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

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

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. 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
```

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

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

7. Add nodes to the `/var/spool/torque/server_priv/nodes` file. See *Specifying Compute Nodes in the Torque Resource Manager Administrator Guide* for information on syntax and options for specifying compute nodes.
8. 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
```

## Install Torque MOMs

In most installations, you will install a Torque MOM on each of your compute nodes.

**i** See Specifying Compute Nodes or Configuring on Compute Nodes in the Torque Resource Manager Administrator Guide for more information.

Do the following:

1. On the Torque Server Host, do the following:

- a. Create the self-extracting packages that are copied and executed on your nodes.

```
[root]# make packages
Building ./torque-package-clients-linux-i686.sh ...
Building ./torque-package-mom-linux-i686.sh ...
Building ./torque-package-server-linux-i686.sh ...
Building ./torque-package-gui-linux-i686.sh ...
Building ./torque-package-devel-linux-i686.sh ...
Done.

The package files are self-extracting packages that can be copied and executed
on your production machines. Use --help for options.
```

- b. Copy the self-extracting packages to each Torque MOM Host.

Adaptive Computing recommends that you use a remote shell, such as SSH, to install packages on remote systems. Set up shared SSH keys if you do not want to supply a password for each Torque MOM Host.

**i** The only required package for the compute node is mom-linux. Additional packages are recommended so you can use client commands and submit jobs from compute nodes.

```
[root]# scp torque-package-mom-linux-i686.sh <mom-node>:
[root]# scp torque-package-clients-linux-i686.sh <mom-node>:
```

- c. Copy the pbs\_mom startup script to each Torque MOM Host.

- Red Hat 6-based systems

```
[root]# scp contrib/init.d/pbs_mom <mom-node>:/etc/init.d
```



- SUSE 11-based systems

```
[root]# scp contrib/init.d/suse.pbs_mom <mom-node>:/etc/init.d/pbs_mom
```

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

```
[root]# scp contrib/systemd/pbs_mom.service <mom-node>:/usr/lib/systemd/system/
```

## 2. On each Torque MOM Host, do the following:

- a. Install the self-extracting packages and run ldconfig.

```
[root]# ssh root@<mom-node>
[root]# ./torque-package-mom-linux-i686.sh --install
[root]# ./torque-package-clients-linux-i686.sh --install
[root]# ldconfig
```

- b. Configure pbs\_mom to start at system boot, and then start the daemon.

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

```
[root]# chkconfig --add pbs_mom
[root]# service pbs_mom start
```

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

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

## Configure Data Management

When a batch job completes, stdout and stderr files are generated and placed in the spool directory on the master Torque MOM Host for the job instead of the submit host. You can configure the Torque batch environment to copy the stdout and stderr files back to the submit host. See *Configuring Data Management* in the *Torque Resource Manager Administrator Guide* for more information.

### Related Topics

[Preparing for Manual Installation 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 30](#)
- [Install Dependencies, Packages, or Clients on page 30](#)

- [\(Optional\) Build a Custom RPM on page 31](#)
- [Install Moab Server on page 32](#)
- [Configure Torque to Trust Moab on page 34](#)
- [Verify the Installation on page 34](#)
- [\(Optional\) Install Moab Client on page 35](#)

## Open Necessary Ports

Moab 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 Host, or if you will be using Moab in a grid, and if you have a firewall enabled, you will need to configure the firewall to allow the server port.

On the Moab Server Host, 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 firewallld

```
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, Packages, or Clients

In this section:

- [Dependencies and Packages on page 31](#)
- [Torque Client on page 31](#)
- [Moab Accounting Manager Client on page 31](#)

## Dependencies and Packages

On the Moab Server Host, use the following commands to install the required Moab 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
```

## Torque Client

If you are using Torque and are installing the Torque Server on a separate host (Torque Server Host) from the Moab Server (Moab Server Host), you will need to install the Torque client on the Moab Server Host in order for Moab to interact with Torque.

Follow the instructions in [Installing Torque Resource Manager on page 22](#) with these exceptions:

- Use the configure options `--disable-server`, `--disable-mom` and `--disable-gui`
- Omit the step initializing the serverdb and all of the steps thereafter

## Moab Accounting Manager Client

If you are using Moab Accounting Manager and will be using the Native (custom script) accounting manager interface, and are installing Moab Accounting Manager on a separate host (MAM Server Host) from the Moab Server (Moab Server Host), you will need to install the Moab Accounting Manager client on the Moab Host in order for the custom scripts to use the MAM API.

Follow the instructions in [Installing Moab Accounting Manager on page 35](#) with the following exceptions:

- No need to install the database or database drivers
- Use the configure option `--without-init`
- Instead of running `make`, use `make clients-only`
- Instead of running `make install`, use `make install-clients-only`
- Omit the step to create the database and all of the steps thereafter

## (Optional) Build a Custom RPM

If you want to build 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 Server

On the Moab Server Host, do the following:

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

**i** The variable marked `<version>` indicates the build's version, revision, and changeset information. 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>
```

If Elastic Computing is part of your Moab Workload Manager configuration, install `deps/acpython-base*`.

```
# Red Hat-based systems
[root]# yum install deps/acpython-base*

# SUSE-based systems
[root]# zypper install deps/acpython-base*
```

3. Configure Moab. If you are installing Moab Accounting Manager, configure Moab with the `--with-am` option.

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

**i** See [Moab Workload Manager Configuration Options](#) on page 68 for a list of commonly used options or use `./configure --help` for a complete list of available 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. 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. If you are using Torque as a resource manager and you installed the Torque Server on a separate host (Torque Server Host), configure the RMCFG HOST parameter to tell Moab the host on which Torque Server is running.

```
RMCFG[torque] HOST=<torque_server_hostname>
```

7. Source the appropriate profile script to add the Moab executable directories to your current shell `$PATH` environment.

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

8. 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](#).

## 9. Start Moab.

- 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
```

## Configure Torque to Trust Moab

If you are using Torque as a resource manager and you installed the Torque Server on a separate host (Torque Server Host); recommended, do the following:

- On the Torque Server Host, add the name of the Moab Server Host (where Moab Server is installed) as a manager and as a submit host.

```
[root]# qmgr
Qmgr: set server managers += root@<moab_server_hostname>
Qmgr: set server submit_hosts += <moab_server_hostname>
Qmgr: exit
```

## Verify the Installation

If you have a resource manager configured, verify that the scheduler is able to schedule a job.

- 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
```

## (Optional) Install Moab Client

After you have installed Moab Server, you can create a client tarball to install just the Moab client commands on a login/client host. This tarball uses a single `tar` command to install the binary Moab client command files and their man pages. The tarball also contains a `moab.cfg` file configured with the Moab Server host name and port number so you do not have to manually configure this information on the login/client node.

**i** If your site needs secure communication and authentication between Moab Client Host and the Moab Server Host, create a site-specific key and place it in the same directory as your `moab.cfg` file. By default, this would be `$MOABHOMEDIR/etc/.moab.key`. When the Moab server and client commands detect the presence of those two files they will use the key in those files to authenticate and communicate, instead of the default key. See [Mauth Authentication](#) in the *Moab Workload Manager Administrator Guide* for more information.

Do the following:

1. On the Moab Server Host, create the client tarball.

```
[root]# make client-pkg
```

2. Copy the tarball to the root directory of the Moab Client Host.
3. On the Moab Client Host, run the tarball to install the Moab client commands.

```
[root]# tar xvf client.tgz
```

Related Topics

[Preparing for Manual Installation on page 20](#)

## Installing Moab Accounting Manager

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

Perform the following in order:

- [Plan Your Installation](#)
- [Open Necessary Ports](#)
- [Install and Initialize the PostgreSQL Server](#)
- [Install Dependencies, Packages, or Clients](#)
- [\(Optional\) Build a Custom RPM](#)
- [Install MAM Server](#)
- [Configure the MAM GUI](#)
- [Access the MAM GUI](#)
- [Configure Moab Workload Manager to use Moab Accounting Manager](#)
- [Initialize Moab Accounting Manager](#)

## Plan Your Installation

The first step is determining the number of separate hosts (physical machines) required for your MAM installation.

Your MAM installation includes:

- MAM Server
- MAM Database
- MAM GUI (optional)
- MAM Clients (possibly several hosts)

Each of these components can be installed on their own hosts (meaning the actual physical machine) or can be combined on same hosts. For example, the MAM Database can be installed on the same *host* as the MAM Server. Or the MAM Server may be installed on the same host you installed the Moab Server.

Once you have determined which components are installed on which hosts, complete the rest of the instructions for the MAM installation.

**i** The instructions that follow in this topic will use the term *Host* after each component to reflect installing on a host (again, meaning the physical machine). For example, MAM Server Host and MAM Database Host. Depending on your configuration, Host may refer to as installed on its own machine or installed on the same machine as another component.

## Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

Do the following as needed:



1. If you will be installing the MAM Server on a separate host than you installed the Moab Server *or* you will be installing the MAM Clients on other hosts, then on the MAM Server Host, open the MAM Server port (7112) in the firewall.

- 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"
-A INPUT -p tcp --dport 7112 -j ACCEPT
[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewallld

```
[root]# firewall-cmd --add-port=7112/tcp --permanent
[root]# firewall-cmd --reload
```

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

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="7112"
[root]# service SuSEfirewall2_setup restart
```

2. If using the MAM GUI, then on the MAM GUI Host, open the https port in the firewall for secure browser communication.

- 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"
-A INPUT -p tcp --dport 443 -j ACCEPT
[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewallld

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

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

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="443"
[root]# service SuSEfirewall2_setup restart
```

3. If you will be installing the MAM Database on a different host from the MAM Server, then on the MAM Database Host, open the postgres port (5432) in the firewall.

- 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"
-A INPUT -p tcp --dport 5432 -j ACCEPT
[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

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

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

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="5432"
[root]# service SuSEfirewall2_setup restart
```

## Install and Initialize the PostgreSQL Server

Moab Accounting Manager uses a database for transactions and data persistence. The PostgreSQL database may be installed on a separate host from the MAM Server; however, it is often convenient to install them on the same host.

On the MAM Database Host, do the following:

1. Install and initialize the PostgreSQL Server.

- 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
```

- SUSE 11-based and SUSE 12-based systems

```
[root]# zypper install postgresql-server
[root]# service postgresql start
```

## 2. Configure trusted connections.

Edit or add a "host" line in the `pg_hba.conf` file for the interface from which the MAM Server 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

# Replace 127.0.0.1 with the IP address of the MAM Server Host if the
# MAM PostgreSQL server is on a separate host from the MAM server.
host      all             all             127.0.0.1/32             md5
host      all             all             ::1/128                  md5
---
```

## 3. If the MAM Database Host is installed on a *separate* host from where you will install the MAM Server, configure PostgreSQL to accept connections from the MAM Server Host.

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

# Replace <mam-server-host> with the interface name from which the MAM server
# will be connecting to the database.
listen_addresses = '<mam-server-host>'
---
```

## 4. Start or restart the database.

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

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

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

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

## Install Dependencies, Packages, or Clients

Use the following instructions to install the required Moab Accounting Manager dependencies, packages, or clients.

**i** Depending on your configuration, the MAM Server Host and the MAM GUI Host may be installed on the same host. The MAM Client Host is automatically installed on the same host as the MAM Server Host; however, you can also install the MAM Client Host on any other hosts on which you want to have the MAM client commands available to users or administrators.

1. On the MAM Server Host, the MAM GUI Host, and the MAM Client Hosts, do the following:

- Red Hat-based systems

```
[root]# yum install gcc redhat-lsb-core perl rrdtool perl-Config-Tiny perl-
Crypt-CBC perl-Crypt-DES perl-Crypt-DES_EDE3 perl-Digest-HMAC perl-Error perl-
Log-Dispatch-FileRotate perl-Log-Log4perl perl-XML-LibXML
```

**i** If installing on RHEL, some packages may not be found in the standard RHEL distribution repositories.

- One way to overcome this problem is to install the missing dependencies from a CentOS repository using a reputable rpm download site (such as rpmfind.net or pbone.net). For example (for RHEL 7.1):

```
[root]# cat > /etc/yum.repos.d/centos.repo <<'EOF'
[centos]
name=CentOS $releasever - $basearch
baseurl=http://rpmfind.net/linux/centos/7.1.1503/os/$basearch/
enabled=0
gpgcheck=0
EOF

[root]# yum install --enablerepo=centos gcc redhat-lsb-core perl rrdtool
perl-Config-Tiny perl-Crypt-CBC perl-Crypt-DES perl-Crypt-DES_EDE3 perl-
Digest-HMAC perl-Error perl-Log-Dispatch-FileRotate perl-Log-Log4perl
perl-XML-LibXML
```

- *Alternatively*, you can install the available packages in the RHEL repository and then install the missing modules from CPAN.

```
[root]# yum install --skip-broken gcc redhat-lsb-core perl rrdtool perl-
Config-Tiny perl-Crypt-CBC perl-Crypt-DES perl-Crypt-DES_EDE3 perl-
Digest-HMAC perl-Error perl-Log-Dispatch-FileRotate perl-Log-Log4perl
perl-XML-LibXML perl-CPAN
[root]# cpan Config::Tiny Log::Log4perl Log::Dispatch::FileRotate
Compress::Zlib
```

You may need to run the cpan command more than once for it to complete successfully.

- SUSE-based systems

```
[root]# zypper install gcc lsb-release perl-Config-Tiny perl-Crypt-CBC perl-
Crypt-DES perl-Crypt-DES_EDE3 perl-Digest-HMAC perl-Error perl-Log-Log4perl
perl-XML-LibXML perl-Params-Validate perl-YAML perl-Log-Dispatch perl-Log-
Dispatch-FileRotate
```

## 2. On the MAM Server Host, do the following:

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

```
[root]# yum install postgresql postgresql-libs perl-DBD-Pg perl-Date-Manip perl-
Time-HiRes perl-DBI
```

- SUSE 11-based systems

```
[root]# zypper install postgresql postgresql-libs perl-DBD-Pg perl-Date-Manip
perl-DBI
```

- SUSE 12-based systems

```
[root]# zypper install postgresql93 postgresql93-devel libpq5 perl-DBD-Pg perl-
Date-Manip perl-DBI
```

## 3. On the MAM GUI Host, do the following:

- Red Hat-based systems

```
[root]# yum install httpd mod_ssl perl-CGI perl-CGI-Session
```

- SUSE-based systems

```
[root]# zypper install apache2 perl-CGI perl-CGI-Session
```

## 4. On each of the MAM Client Hosts (including the MAM Server Host), do the following:

- Red Hat 6-based systems

```
[root]# yum install perl-suidperl perl-Term-ReadLine-Gnu perl-TermReadKey
```

- Red Hat 7-based systems

```
[root]# yum install perl-CPAN openssl-devel readline-devel ncurses-devel perl-
TermReadKey
[root]# cpan Term::ReadLine::Gnu
```

- SUSE 11-based systems

```
[root]# zypper install libopenssl-devel perl-TermReadLine-Gnu perl-TermReadKey
[root]# chmod 4755 /usr/bin/sperl*
```

- SUSE 12-based systems

```
[root]# zypper install libopenssl-devel perl-TermReadLine-Gnu perl-Term-ReadKey
```

**i** If any of the Perl module packages fail to install or are unavailable for your system, you can install it from CPAN by running `cpan MODULENAME` where *MODULENAME* is the respective perl module name.

## (Optional) Build a Custom RPM

If you want to build a custom RPM, do the following:

1. Install `rpm-build`.

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

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

**i** The variable marked `<version>` is the desired version of the suite; for example, 9.0.0.

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

## Install MAM Server

On the MAM Server Host, do the following:

1. Create a user called `mam` and switch to that user.

```
[root]# useradd -m mam
[root]# su - mam
[mam]$ mkdir src
[mam]$ cd src
```

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

**i** The variable marked `<version>` indicates the build's version.

3. Untar the MAM tarball.

```
[mam]$ tar -zxvf mam-9.0.0.tar.gz
```

4. Navigate to `mam-9.0.0`.

```
[mam]$ cd mam-9.0.0
```

5. Configure the software. For a list of all the configuration options, see [Moab Accounting Manager Configuration Options on page 69](#).

```
[mam]$ ./configure
```

6. Compile the software.

```
[mam]$ make
```

**i** If you only need to install the clients on a particular system, replace `make` with `make clients-only`. If you only need to install the web GUI on a particular system, replace `make` with `make gui-only`.

7. Install the software.

```
[mam]$ exit
[root]# cd ~mam/src/mam-9.0.0
[root]# make install
```

**i** If you only need to install the clients on a particular system, replace `make install` with `make install-clients-only`. If you only need to install the web GUI on a particular system, replace `make install` with `make install-gui-only`.

8. As the database user, create a database called `mam` and grant database privileges to the `mam` user.

**i** PostgreSQL should have previously been installed using the instructions in [Preparing for Manual Installation on page 20](#).

```
[root]# su - postgres
[postgres]$ psql

create database mam;
create user mam with password 'changeme!';
\q

[postgres]$ exit
```

The password you define must be synchronized with the `database.password` value in `/opt/mam/etc/mam-server.conf`

```
[root]# vi /opt/mam/etc/mam-server.conf

database.password = changeme!
```

9. Run the `hpc.sql` script to populate the Moab Accounting Manager database with objects, actions, and attributes necessary to function as an Accounting Manager.

```
[root]# su - mam
[mam]$ cd src/mam-9.0.0
[mam]$ psql mam < hpc.sql
[mam]$ exit
```

10. Configure MAM to automatically start up at system boot; start the `mam` service.

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

```
[root]# chkconfig --add mam
[root]# service mam start
```

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

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

## Configure the MAM GUI

If you plan to use the web GUI, then on the MAM GUI Host, do the following:

1. As `root`, add or edit the SSL virtual host definition as appropriate for your environment. To do so, configure the `cgi-bin` directory in `ssl.conf`. Below the `cgi-bin` directory element, create an alias for `/cgi-bin` pointing to your `cgi-bin` directory. If you chose to install to a `cgi-bin` sub-directory, you might want to create an alias for that as well. Also, add `index.cgi` to the `DirectoryIndex` so you can use the shorter sub-directory name.

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

```
[root]# vi /etc/httpd/conf.d/ssl.conf

<Directory "/var/www/cgi-bin">
## Add these lines
Options ExecCGI
AddHandler cgi-script .cgi
AllowOverride All
Order allow,deny
Allow from all
</Directory>

# Aliases for /cgi-bin
Alias /cgi-bin/ /var/www/cgi-bin/
Alias /mam /var/www/cgi-bin/mam/

# Make shorter sub-dir name available
DirectoryIndex index.cgi
```

- SUSE 11-based systems



```
[root]# a2enflag SSL
[root]# cp /etc/apache2/vhosts.d/vhost-ssl.template /etc/apache2/vhosts.d/mam-ssl.conf
[root]# vi /etc/apache2/vhosts.d/mam-ssl.conf

<Directory "/srv/www/cgi-bin">
## Add these lines
    Options ExecCGI
    AddHandler cgi-script .cgi
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>

# Aliases for /cgi-bin
Alias /cgi-bin/ /srv/www/cgi-bin/
Alias /mam /srv/www/cgi-bin/mam/

# Make shorter sub-dir name available
DirectoryIndex index.cgi
```

## • SUSE 12-based systems

```
[root]# a2enflag SSL
[root]# cp /etc/apache2/vhosts.d/vhost-ssl.template /etc/apache2/vhosts.d/mam-ssl.conf
[root]# vi /etc/apache2/vhosts.d/mam-ssl.conf

<Directory "/srv/www/cgi-bin">
    Options ExecCGI
    AddHandler cgi-script .cgi
    AllowOverride All
    Require all granted
</Directory>

# Aliases for /cgi-bin
Alias /cgi-bin/ /srv/www/cgi-bin/
Alias /mam /srv/www/cgi-bin/mam/

# Make shorter sub-dir name available
DirectoryIndex index.cgi
```

2. For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow the web server to make network connections, use `setuid` for authentication, and write to the log file.

### a. Determine the current mode of SELinux.

```
[root]# getenforce

Enforcing
```

- If the command returns a mode of **Disabled** or **Permissive**, or if the `getenforce` command is not found, you can skip the rest of this step.
- If the command returns a mode of **Enforcing**, you can choose between options of customizing SELinux to allow the web GUI to perform its required functions or disabling SELinux on your system.

- b. If you choose to customize SELinux, do the following:

**i** SELinux can vary by version and architecture and that these instructions may not work in all possible environments.

**i** If you used the `--prefix=<prefix>` configuration option when you configured Moab Accounting Manager, you must replace references to `/opt/mam` in the example below with the `<prefix>` you specified. See [Moab Accounting Manager Configuration Options on page 69](#).

- Red Hat 6-based systems

```
[root]# cat > mamgui.te <<EOF
module mamgui 1.0;
require {
    type httpd_sys_script_t;
    type port_t;
    class capability setuid;
    class tcp_socket name_connect;
}
allow httpd_sys_script_t port_t:tcp_socket name_connect;
allow httpd_sys_script_t self:capability setuid;
EOF
[root]# checkmodule -M -m -o mamgui.mod mamgui.te
[root]# semodule_package -m mamgui.mod -o mamgui.pp
[root]# semodule -i mamgui.pp
[root]# setenforce 0
[root]# chcon -v -t httpd_sys_content_t /opt/mam/log
[root]# setenforce 1
```

- Red Hat 7-based systems

```
[root]# yum install checkpolicy policycoreutils-python
[root]# cat > mamgui.te <<EOF
module mamgui 1.0;
require {
    type httpd_sys_script_t;
    type unreserved_port_t;
    class tcp_socket name_connect;
}
allow httpd_sys_script_t unreserved_port_t:tcp_socket name_connect;
EOF
[root]# checkmodule -M -m -o mamgui.mod mamgui.te
[root]# semodule_package -m mamgui.mod -o mamgui.pp
[root]# semodule -i mamgui.pp
[root]# setenforce 0
[root]# chcon -v -t httpd_sys_rw_content_t /opt/mam/log
[root]# setenforce 1
```

3. For the highest security, it is recommended that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS 7 is documented at [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/System\\_Administrators\\_Guide/ch-Web\\_Servers.html#s2-apache-mod\\_ssl](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Web_Servers.html#s2-apache-mod_ssl).

Alternatively, if your network domain can be secured from man-in-the-middle attacks, you could use a self-signed certificate. Often this does not require any additional steps since in many distributions, such as Red Hat, the Apache SSL configuration provides self-signed certificates by default.

If your configuration uses self-signed certificates, do the following:

- Red Hat-based systems

No action required. RedHat ships with ready-made certificates.

- SUSE-based systems

```
[root]# cd /etc/apache2
[root]# openssl genrsa -out ssl.key/server.key 1024
[root]# openssl req -new -key ssl.key/server.key -x509 -out ssl.crt/server.crt
```

#### 4. Start or restart the HTTP server daemon.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

- SUSE 11-based systems

```
[root]# chkconfig apache2 on
[root]# service apache2 restart
```

- SUSE 12-based systems

```
[root]# systemctl enable apache2.service
[root]# systemctl restart apache2.service
```

## Access the MAM GUI

If you plan to use the web GUI, then on the MAM Server Host, do the following:

1. Create a password for the `mam` user to be used with the MAM Web GUI.

```
[root]# su - mam
[mam]$ mam-set-password
```

2. Verify the connection.

- a. Open a web browser and navigate to `https://<mam-server-host>/cgi-bin/mam`.
- b. Log in as the `mam` user with the password you set in step 1.

## Configure Moab Workload Manager to use Moab Accounting Manager

Do the following, where applicable:

1. On the *Moab* Server Host, edit the Moab configuration file.

```
[root]# vi /opt/moab/etc/moab.cfg
AMCFG[mam] TYPE=MAM HOST=<mam_server_host>
```

- a. Uncomment the AMCFG lines and customize as needed. See [Accounting, Charging, and Allocation Management](#) in the *Moab Workload Manager Administrator Guide*.
  - b. If the Moab Server and the MAM Server are on the *same* host, set HOST to 'localhost'; otherwise, set HOST to the host name for the MAM Server (MAM Server Host).
2. Configure Moab to authenticate with MAM using the MAM secret key.
    - a. On the *MAM* Server Host, copy the auto-generated secret key from the `token.value` value in the `/opt/mam/etc/mam-site.conf` file.
    - b. On the *Moab* Server Host, add the secret key to the `moab-private.cfg` file as the value of the CLIENTCFG KEY attribute.

```
[root]# vi /opt/moab/etc/moab-private.cfg
CLIENTCFG[AM:mam] KEY=<MAMSecretKey>
```

- c. Restart Moab.

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

```
service moab restart
```

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

```
systemctl restart moab.service
```

## Initialize Moab Accounting Manager

You will need to initialize Moab Accounting Manager to function in the way that is most applicable to the needs of your site. See [Initial Setup](#) in the *Moab Accounting Manager Administrator Guide* to set up Moab Accounting Manager for your desired accounting mode.

Related Topics

[Preparing for Manual Installation on page 20](#)

## Installing Moab Web Services



You must deploy Moab Web Services on the *same* host as Moab Server (Moab Server Host). If using Viewpoint, this shared host must have a Red Hat-based OS; regardless of whether Viewpoint is also installed on that host. For documentation clarity, these instructions refer to the shared host for Moab Server and MWS as the MWS Host.

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

In this topic:

- [Open Necessary Ports on page 49](#)
- [Install Dependencies, Packages, or Clients on page 51](#)
- [Install MWS Server on page 53](#)

### Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

In this section:

- [Open the Tomcat Port \(8080\) on page 49](#)
- [Open the MWS MongoDB Database Port \(27017\) on page 50](#)

### Open the Tomcat Port (8080)

On the MWS Server Host, 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"

-A INPUT -p tcp --dport 8080 -j ACCEPT

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

- Red Hat 7-based systems using firewallld

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

- SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="8080"
[root]# service SuSEfirewall2_setup restart
```

## Open the MWS MongoDB Database Port (27017)

**i** Depending on your system configuration, your MongoDB databases may not be installed on the same host as their corresponding component servers. For example, you may choose to install the MWS MongoDB database on the same host where you have installed other MongoDB databases instead of on the MWS Server Host.

Do the following, as needed:

- If you have chosen to install the MWS MongoDB database on the *same* host you installed other MongoDB databases (for example, the same host you installed the Moab MongoDB database), confirm the firewall port (27017) is already opened on that host.
- If you have chosen to install the MWS MongoDB database on a *separate* host from other MongoDB databases, you will need to open the MWS MongoDB database port in firewall for that host. To open the port in the firewall, do the following:

- Red Hat 6-based systems using iptables

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

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

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

- SUSE 11-based or SUSE 12-based systems using SuSEfirewall2 (SUSE 11 is not supported on the MWS Server Host)

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="27017"
[root]# service SuSEfirewall2_setup restart
```

## Install Dependencies, Packages, or Clients

In this section:

- [Install Java on page 51](#)
- [Install Tomcat on page 51](#)
- [Install MongoDB on page 51](#)

### Install Java

Install the Linux x64 RPM version of Oracle® Java® 8 Runtime Environment.

**i** Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run MWS.

On the MWS Host, do the following:

1. Install the Linux x64 RPM version of Oracle Java SE 8 JRE.
  - a. Go to the [Oracle Java download page](http://java.com/en/download/linux_manual.jsp) ([http://java.com/en/download/linux\\_manual.jsp](http://java.com/en/download/linux_manual.jsp)).
  - b. Copy the URL for the Linux x64 RPM version, and run the following command:

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

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

- SUSE 12-based systems

```
[root]# ln -s /usr/sbin/update-alternatives /usr/sbin/alternatives
[root]# rpm -Uh <URL>
```

### Install Tomcat

On the MWS Host, do the following:

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

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

- SUSE 12-based systems

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

### Install MongoDB

To install and enable MongoDB, on the MWS Host, 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-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 12-based systems

```
[root]# systemctl enable mongod.service
[root]# systemctl start mongod.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 12-based systems

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

## Install MWS Server

On the MWS Host, do the following:

1. Verify Moab Server is installed and configured as desired (for details, see [Installing Moab Workload Manager on page 29](#)).

2. Start Moab.

- Red Hat 6-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.

For more information, see [Configuration](#) in the *Moab Web Services Administrator 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. 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, 9.0.0.

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-9.0.0.tar.gz
```

6. Copy the extracted utility files to the utility directory created in the previous

step and give the tomcat user ownership of the directory.

```
[root]# cd /tmp/mws-install/mws-9.0.0/Utils
[root]# cp * /opt/mws/Utils
[root]# chown tomcat:tomcat /opt/mws/Utils/*
```

## 7. 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 on page 51](#).

```
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]# mdia -S

Mongo connection (localhost) is up (credentials are set)
```

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

```
[root]# systemctl restart moab.service
[root]# mdia -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. Generate a key and store the result in `/opt/moab/etc/.moab.key`.

- Red Hat 6-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
```

b. (Optional) Moab supports message queue security using AES. This feature requires a Base64-encoded 16-byte (128-bit) shared secret. Do the following:

i. Generate a key and append the result to `/opt/moab/etc/moab-private.cfg`

- Red Hat 6-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
```

**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.

ii. Verify that encryption is on for the ZeroMQ connection.

```
[root]# mdia -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 ownership.

```
[root]# cd /tmp/mws-install/mws-9.0.0
[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
```

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 you generated earlier.



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](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.

c. If you are using Moab Accounting Manager, change these settings in `/opt/mws/etc/mws.d/mws-config-hpc.groovy`:

- **mam.secretKey**: needs to match the MAM secret key in `/opt/mam/etc/mam-site.conf` on the MAM Server (as **token.value**)
- **mam.server**: set to the hostname of the MAM Server
- **mam.port**: set to the port of the MAM Server

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

mam.secretKey = "<ENTER-KEY-HERE>"
mam.server = "localhost"
mam.port = 7112
```

d. 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 the `mws-config.groovy` file:

```
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](#), 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](#).

**i** To see how to configure a secure connection to the LDAP server, see [Securing the LDAP Connection](#).

- 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](#).

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

- e. 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](#) section.

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

- f. Make the MWS configuration files read-only.

```
[root]# chmod 400 /opt/mws/etc/mws-config.groovy /opt/mws/etc/mws.d/mws-config-hpc.groovy
```

## 10. Configure Tomcat

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"
```

**i** MaxPermSize is ignored using Java 8; and therefore can be omitted.

## 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-9.0.0/mws.war /usr/share/tomcat/webapps
[root]# service tomcat 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-9.0.0/mws.war /usr/share/tomcat/webapps
[root]# systemctl start tomcat.service
```

12. Navigate to `http://<server>: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 176](#).

#### Related Topics

[Preparing for Manual Installation on page 20](#)

## Installing Moab Insight

Manual installation is not available for Insight.

See [Installing Moab Insight on page 127](#) for detailed instructions on installing the Insight RPM.

**i** Adaptive Computing *requires* the Insight Server and the Moab Workload Manager Server to run on separate hosts.

## Installing Moab Viewpoint

Manual installation is not available for Viewpoint.

See [Installing Moab Viewpoint on page 139](#) for detailed instructions on installing the Viewpoint RPM.



## Additional Configuration

In this section:

- [Configuring SSL in Tomcat on page 150](#)
- [Setting Up OpenLDAP on CentOS 6 on page 150](#)
- [Moab Workload Manager Configuration Options on page 68](#)
- [Moab Accounting Manager Configuration Options on page 69](#)
- [Trusting Servers in Java on page 157](#)

### Configuring SSL in Tomcat

To configure SSL in Tomcat, please refer to the Apache Tomcat [documentation](http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html) (<http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>).

### 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 61](#)
- [Adding an Organizational Unit \(OU\) on page 66](#)
- [Adding a User on page 66](#)
- [Adding a Group on page 67](#)
- [Adding a User to a Group on page 67](#)

 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}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
```

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}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

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}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```



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

**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.

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` earlier in this procedure, 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](http://wiki.centos.org/HowTos/Network/IPTables) (<http://wiki.centos.org/HowTos/Network/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 66](#).

## 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 66](#).

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
<b>--prefix</b>	Specifies the location of the binaries and libraries of the Moab install. The default location is <code>/opt/moab</code> .	<pre>[root]# ./configure --prefix=/usr/local</pre>
<b>--with-am</b>	Specifies that you want to configure Moab with Moab Accounting Manager. <div> 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.</div>	<pre>[root]# ./configure --with-am</pre>
<b>--with-flexlm</b>	Causes Moab to install the <code>license.-mon.flexLM.pl</code> script in the <code>/opt/moab/tools</code> directory. For more information about this script, see the <a href="#">Interfacing to FLEXlm</a> section in the Moab Administrator Guide.	<pre>[root]# ./configure --with-flexlm</pre>



Option	Description	Example
<b>--with-homedir</b>	<p>Specifies the location of the Moab configuration directory and the MOABHOMEDIR environment variable. The default location is <code>/opt/moab</code>.</p> <div>  <b>MOABHOMEDIR</b> is automatically set on some distributions during installation, when the <code>--with-profile</code> option is enabled. </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>
<b>--without-init</b>	<p>Disables the installation of a distribution-specific, Moab service startup file. By default, make install will install an <code>init.d</code> or <code>systemd</code> service startup file as appropriate for your distribution. The installed file (<code>/etc/init.d/moab</code> or <code>/usr/lib/systemd/system/moab.service</code>) may be customized to your needs. If you do not want this file to be installed, use this option to exclude it.</p>	<pre>[root]# ./configure --without-init</pre>
<b>--without-profile</b>	<p>Disables the installation of a distribution-specific shell profile for bash and C shell. By default, make install will install the Moab shell initialization scripts as appropriate for your operating system. These scripts help to establish the MOABHOMEDIR, PERL5LIB, PATH and MANPATH environment variables to specify where the new moab configuration, scripts, binaries and man pages reside. The installed scripts (<code>/etc/profile.d/moab.{csh,sh}</code>) may be customized to your needs. If you do not want these scripts to be installed, use this option to exclude them.</p>	<pre>[root]# ./configure --without-profile</pre>

## Moab Accounting Manager Configuration Options

The following table comprises commonly-used configure options.

Option	Description
<b>-h,--help</b>	Run <code>./configure --help</code> to see the list of configure options.

Option	Description
<b>--localstatedir=DIR</b>	Home directory where per-configuration subdirectories (such as <code>etc</code> , <code>log</code> , <code>data</code> ) will be installed (defaults to <code>PREFIX</code> ).
<b>--prefix=PREFIX</b>	Base installation directory where all subdirectories will be installed unless otherwise designated (defaults to <code>/opt/mam</code> ).
<b>--with-cgi-bin=DIR</b>	If you intend to use the web GUI, use <code>--with-cgi-bin</code> to specify the directory where you want the Moab Accounting Manager CGI files to reside (defaults to <code>/var/www/cgi-bin/mam</code> ).
<b>--with-db-name=NAME</b>	Name of the SQL database that the server will sync with (defaults to <code>mam</code> ).
<b>--with-legacy-links</b>	Creates symbolic links allowing the use of the old client and server command names (for example, <code>mam-list-users</code> would be created as symbolic link to <code>mam-list-users</code> ). When running a command under its old name, the command will issue a deprecation warning. This warning can be disabled by setting <code>client.deprecationwarning = false</code> in the <code>mam-client.conf</code> file. The default is not to install the legacy links.
<b>--with-mam-libs=local site</b>	Use <code>--with-mam-libs</code> to indicate whether you want to install the Perl MAM modules in a local directory ( <code>\${exec_prefix}/lib</code> ) or in the default system site-perl directory (defaults to <code>local</code> ).
<b>--with-promotion=mamauth suidperl</b>	Command-line clients and scripts using the API need to use a security promotion method to authenticate and encrypt the communication using the symmetric key. The default is <code>suidperl</code> if it is installed on the system, otherwise the default is <code>mamauth</code> . See the description for the <b>security.promotion</b> configuration parameter in the <a href="#">Client Configuration</a> section for more information about the two security promotion methods.
<b>--with-sha=SHA SHA1</b>	Allows you to override the auto-detected SHA Digest Perl (whether <code>Digest::SHA1</code> or <code>Digest::SHA</code> ) that should be used for your system.
<b>--with-user=USER</b>	Use <code>--with-user</code> to specify the accounting admin userid that the server will run under and who will have full administrative privileges (defaults to the user running the <code>configure</code> command). It is recommended that this be a non-privileged user for the highest security.
<b>--without-gui</b>	Specifies whether to install the CGI web GUI. If you do not intend to use the CGI web GUI, you can specify <code>--without-gui</code> to not install the CGI scripts. Otherwise, the default is to install the GUI CGI scripts.

Option	Description
<b>--without-init</b>	If you do not intend to use the <code>mam init.d</code> service, you can use <code>--without-init</code> to specify that Moab HPC Suite should not install the <code>mam init.d</code> script. Otherwise, the script is installed by default.
<b>--without-profile</b>	If you do not intend to use the <code>mam profile.d</code> environment scripts, you can use <code>--without-profile</code> to specify that Moab HPC Suite should not install the <code>mam profile.d</code> scripts. Otherwise, the scripts are installed by default.

## Trusting Servers in Java

In this topic:

[Prerequisites on page 71](#)

[Retrieve the Server's X.509 Public Certificate on page 71](#)

[Add the Server's Certificate to Java's Keystore on page 72](#)

### 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:

- [Preparing for Upgrade on page 73](#)
- [Upgrading Torque on page 74](#)
- [Upgrading Moab Workload Manager on page 80](#)
- [Upgrading Moab Accounting Manager on page 82](#)
- [Upgrading Moab Web Services on page 87](#)
- [Migrating the MAM Database from MySQL to PostgreSQL on page 169](#)

## 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](#)
2. Torque. See [Upgrading Torque](#)
3. Moab Workload Manager. See [Upgrading Moab Workload Manager](#)
4. Moab Accounting Manager. See [Upgrading Moab Accounting Manager](#)
5. Moab Web Services. See [Upgrading Moab Web Services](#)

## Related Topics

- [Chapter 1 Planning your Installation on page 3](#)

## Upgrading Torque

Torque 6.0.0 binaries are backward compatible with Torque 5.0 or later. However they are not backward compatible with Torque versions prior to 5.0. When you upgrade to Torque 6.0.0 from versions prior to 5.0, all MOM and server daemons must be upgraded at the same time.

The job format is compatible between 6.0.0 and previous versions of Torque and any queued jobs will upgrade to the new version. It is not recommended to upgrade Torque while jobs are in a running state.

This topic contains instructions on how to upgrade and start Torque Resource Manager (Torque).

**i** If you need to upgrade a Torque version prior to 4.0, contact Adaptive Computing.

### In this topic:

- [Before You Upgrade on page 74](#)
- [Stop Torque Services on page 75](#)
- [Upgrade the Torque Server on page 76](#)
- [Update the Torque MOMs on page 77](#)
- [Update the Torque Clients on page 78](#)
- [Start Torque Services on page 79](#)
- [Perform Status and Error Checks on page 79](#)

## Before You Upgrade

This section contains information of which you should be aware before upgrading.

### In this section:

- [serverdb on page 74](#)
- [Running Jobs on page 75](#)
- [Cray Systems on page 75](#)

### serverdb

The `pbs_server` configuration is saved in the file `TORQUE_HOME/server_priv/serverdb`. When running Torque 4.1 or later for the first time, this file

converts from a binary file to an XML-like format.

## Running 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 (using the "started" queue attribute). See `pbsnodes` or `Queue Attributes` in the *Torque Resource Manager Administrator Guide* for more information.

## Cray Systems

For upgrading Torque to 6.0.0 on a Cray system, refer to the [Installation Notes for Moab and Torque for Cray](#) in Appendix G of the *Moab Workload Manager Administrator Guide*.

## Stop Torque Services

Do the following:

1. On the Torque Server Host, shut down the Torque server.

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

```
[root]# service pbs_server stop
```

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

```
[root]# systemctl stop pbs_server.service
```

2. On each Torque MOM Host, shut down the Torque MOM service.



Confirm all jobs have completed before stopping `pbs_mom`. You can do this by typing "`momctl -d3`". If there are no jobs running, you will see the message "NOTE: no local jobs detected" towards the bottom of the output. If jobs are still running and the MOM is shutdown, you will only be able to track when the job completes and you will not be able to get completion codes or statistics.

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

```
[root]# service pbs_mom stop
```

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

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

3. On each Torque Client Host (including the Moab Server Host, the Torque Server Host, and the Torque MOM Hosts, if applicable), shut down the `trqauthd` service.

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

```
[root]# service trqauthd stop
```

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

```
[root]# systemctl stop trqauthd.service
```

## Upgrade the Torque Server

On the Torque Server Host, do the following:

1. Back up your `server_priv` directory.

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

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

- Red Hat-based systems

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

- SUSE-based systems

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

3. Download the latest 6.0.0 build from the [Adaptive Computing](#) website.
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. Update the `pbs_server` service startup script.

- a. Make a backup of your current service startup script.

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

```
[root]# cp /etc/init.d/pbs_server pbs_server.bak
```

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

```
[root]# cp /usr/lib/systemd/system/pbs_server.service pbs_server.service.bak
```

- b. Copy in the new stock service startup script.

- Red Hat 6-based systems

```
[root]# cp contrib/init.d/pbs_server /etc/init.d
```



- SUSE 11-based systems

```
[root]# cp contrib/init.d/suse.pbs_server /etc/init.d/pbs_server
```

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

```
[root]# cp contrib/systemd/pbs_server.service /usr/lib/systemd/system/
```

c. Merge in any customizations.

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

```
[root]# vi /etc/init.d/pbs_server
```

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

```
[root]# vi /usr/lib/systemd/system/pbs_server.service
```

## Update the Torque MOMs

Do the following:

1. On the Torque Server Host, do the following:

- a. Create the self-extracting packages that are copied and executed on your nodes.

```
[root]# make packages
Building ./torque-package-clients-linux-i686.sh ...
Building ./torque-package-mom-linux-i686.sh ...
Building ./torque-package-server-linux-i686.sh ...
Building ./torque-package-gui-linux-i686.sh ...
Building ./torque-package-devel-linux-i686.sh ...
Done.
```

The package files are self-extracting packages that can be copied and executed on your production machines. Use --help for options.

- b. Copy the self-extracting packages to each Torque MOM Host.

Adaptive Computing recommends that you use a remote shell, such as SSH, to install packages on remote systems. Set up shared SSH keys if you do not want to supply a password for each Torque MOM Host.

```
[root]# scp torque-package-mom-linux-i686.sh <torque-mom-host>:
```

- c. Copy the pbs\_mom startup script to each Torque MOM Host.

- Red Hat 6-based systems

```
[root]# scp contrib/init.d/pbs_mom <torque-mom-host>:/etc/init.d
```

- SUSE 11-based systems

```
[root]# scp contrib/init.d/suse.pbs_mom <torque-mom-host>:/etc/init.d/pbs_mom
```

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

```
[root]# scp contrib/systemd/pbs_mom.service <torque-mom-host>:/usr/lib/systemd/system/
```

2. On each Torque MOM Host, do the following:

**i** Many of these steps can be done from the Torque server from a remote shell, such as SSH. Set up shared SSH keys if you do not want to supply a password for each Torque MOM Host.

```
[root]# ./torque-package-mom-linux-x86_64.sh --install
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
```

## Update the Torque Clients

This section contains instructions on updating the Torque clients on the Torque Client Hosts (including the Moab Server Host and Torque Mom Hosts, if applicable).

1. On the Torque Server Host, do the following:

- a. Copy the self-extracting packages to each Torque Client Host.

Adaptive Computing recommends that you use a remote shell, such as SSH, to install packages on remote systems. Set up shared SSH keys if you do not want to supply a password for each Torque MOM Host.

```
[root]# scp torque-package-clients-linux-i686.sh <torque-client-host>:
```

- b. Copy the trqauthd startup script to each Torque Client Host.

- Red Hat 6-based systems

```
[root]# scp contrib/init.d/trqauthd <torque-client-host>:/etc/init.d
```

- SUSE 11-based systems

```
[root]# scp contrib/init.d/suse.trqauthd <torque-client-host>:/etc/init.d/trqauthd
```

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

```
[root]# scp contrib/systemd/trqauthd.service <torque-client-host>:/usr/lib/systemd/system/
```

2. On each Torque Client Host, do the following:

**i** Many of these steps can be done from the Torque server from a remote shell, such as SSH. Set up shared SSH keys if you do not want to supply a password for each Torque Client Host.

```
[root]# ./torque-package-clients-linux-x86_64.sh --install
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
```

## Start Torque Services

Do the following:

1. On each Torque Client Host (including the Moab Server Host, Torque Server Host and Torque MOM Hosts, if applicable), start up the trqauthd service.

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

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

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

```
[root]# systemctl daemon-reload
[root]# systemctl start trqauthd.service
```

2. On each Torque MOM Host, start up the Torque MOM service.

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

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

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

```
[root]# systemctl daemon-reload
[root]# systemctl start pbs_mom.service
```

3. On the Torque Server Host, start up the Torque server.

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

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

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

```
[root]# systemctl daemon-reload
[root]# systemctl start pbs_server.service
```

## Perform Status and Error Checks

On the Torque Server Host, do the following:

1. Check the status of the jobs in the queue.

```
[root]# qstat
```

2. Check for errors.

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

## Upgrading Moab Workload Manager

This topic provides instructions to upgrade Moab Workload Manager to the latest release version. Depending on which version of Moab you are presently running, upgrade instructions may vary.

Moab Workload Manager uses the standard configure, make, and make install steps for upgrades. This topic provides a number of sample steps referenced to a particular installation on a Linux platform using the bash shell. These steps indicate the user ID in brackets performing the step. The exact commands to be performed and the user that issues them will vary based on the platform, shell, installation preferences, and other factors.

It is highly recommended that you *first* perform upgrades in a *test environment*. See the warning in [Preparing for Upgrade on page 73](#). It is also recommend that you verify the policies, scripts, and queues work the way you want them to in this test environment. See [Testing New Releases and Policies](#) in the *Moab Workload Manager Administrator Guide*.

If you are also upgrading Torque from an older version (pre-4.0), contact Adaptive Computing.

**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.

## Upgrade Moab Workload Manager

On the Moab Server Host, do the following:

1. If you have not already done so, install extra packages from the add-on repositories. See [Enable Extra Packages for the Repository on page 21](#)
2. Download the latest Moab build (`moab-<version>-<OS>.tar.gz`) from the [Adaptive Computing](#) website.

**i** The variable marked `<version>` indicates the build's version, revision, and changeset information. The variable marked `<OS>` indicates the OS for which the build was designed.

3. Untar the distribution file. For example:

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

4. Change directory into the extracted directory.

## 5. 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`.

## 6. 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.

## 7. Back up your Moab Workload Manager home directory (`/opt/moab/` by default) before continuing.

## 8. 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.

**i** CPAN is installed on SUSE-based systems by default. If upgrading on a Red Hat-based system, you will need to install CPAN `[root]# yum install perl-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.

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

## 9. 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.

## 10. If you use ODBC, you must upgrade to the 9.0.0 schema. See [Migrating Your Database to Newer Versions of Moab](#) for more information.

## 11. Verify the version number is correct before starting the new server version.

```
[root]# moab --about

Defaults:  server=:42559  cfgdir=/opt/moab (env)  vardir=/opt/moab
Build dir:  /tmp/jenkins/workspace/MWM-9.0.0/label/build-sles12
Build host: us-devops-build10
Build date: Fri Oct 09 13:00:00 MST 2015
Build args: NA
Compiler Flags:  -D_M64 -D_BUILDDATETIME="2015100913" -DMUSEZEROMQ -
DMUSEWEBSERVICES -DMUSEMONGODB -DMMAX_GRES=512 -DMMAX_RANGE=2048 -DMMAX_TASK=32768
-fPIC -gdwarf-3 -Wall -Wextra -DVALGRIND -Og -x c++ -std=c++11 -DDMAX_PJOB=512 -D_
GNU_SOURCE
Compiled as little endian.
Version: moab server 9.0.0 (revision 2015100913, changeset
14dee972ebcee919207e48054e9f285db9f6a555)
```

12. If you are using Moab Accounting Manager with the native interface (**TYPE=***native*), remove all entries in `moab.cfg` with the form (`AMCFG[*]*URL=exec://*`), except for those that you have customized.

**i** In Moab Workload Manager 8.1 and after, Moab defaults to using a set of stock scripts that no longer need to be explicitly configured in the server configuration file.

13. Start Moab.

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

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

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

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

## Upgrading Moab Accounting Manager

This topic provides instructions to upgrade MAM to the latest release version. It includes instructions for migrating your database schema to a new version if necessary.

Moab Accounting Manager uses the standard `configure`, `make`, and `make install` steps for upgrades. This document provides a number of sample steps referenced to a particular installation on a Linux platform using the bash shell. These steps indicate the user ID in brackets performing the step. The exact commands to be performed and the user that issues them will vary based on the platform, shell, installation preferences, and other factors.

## Upgrade Moab Accounting Manager

On the MAM Server Host, do the following:

1. Determine the MAM Accounting admin user and change to that user.
  - If you are upgrading MAM from a version *prior* to 9.0, use `gluser`.

```
[root]# gluser | grep 'Accounting Admin'
mam      True
Accounting Admin
[root]# su - mam
```

- If you are upgrading MAM at or after 9.0, use `mam-list-users`.

```
[root]# mam-list-users | grep 'Accounting Admin'
mam      True
Accounting Admin
[root]# su - mam
```

2. Determine whether you need to migrate your database.

- a. Determine your database version.

- If you are upgrading MAM from a version *prior* to 9.0, run `goldsh System Query`.

```
[mam]$ goldsh System Query
```

- If you are upgrading MAM at or after 9.0, run `mam-shell System Query`.

```
[mam]$ mam-shell System Query
```

- b. If the current version is lower than 9.0, you must migrate your database. The steps required to do so are incorporated in the remaining steps for this topic.

3. Stop the server daemon.

- If you are upgrading MAM from a version *prior* to 9.0, run `goldd -k`.

```
[mam]$ goldd -k
```

- If you are upgrading MAM at or after 9.0, run `mam-server -k`.

```
[mam]$ mam-server -k
```

4. If you determined that you must migrate your database, create a database backup.

- PostgreSQL database.

```
[mam]$ pg_dump -U <mam_database_user> -W <old_database_name> > /tmp/<old_database_name>.sql
```

- MySQL database.

```
[mam]$ mysqldump -u <mam_database_user> -p <old_database_name> > /tmp/<old_database_name>.sql
```

5. Verify that each of the prerequisites listed in [Installing Moab Accounting Manager on page 35](#) have been satisfied.

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

**i** The variable marked `<version>` indicates the build's version.

- Unpack the tar archive and change directory into the top directory of the distribution.

```
[mam]$ tar -zxvf mam-9.0.0.tar.gz
[mam]$ cd mam-9.0.0
```

- Configure Moab Accounting Manager by running the `configure` script with the desired options.

It is recommended that you use the same configure options that were used in the previous installation. You can examine the `config.log` file where you unpacked your previous distribution to help determine the configuration options that were used to install the prior version of MAM.

In addition to your previous configuration options, it is recommended that you specify the prior symmetric key used between Moab Workload Manager and Moab Accounting Manager via the `--with-key` configure option so that you do not have to change the key in Moab. You can find this key in the prior `site.conf` or `mam-site.conf` file.

The examples in this guide demonstrate installing the new version of Moab Accounting Manager over the previous version by using the default directory, database name and port.

**i IMPORTANT:** client and server command names have changed beginning with 9.0. If you want to create symbolic links to enable you to continue to use the old client and server command names, use the `--with-legacy-links` option with `configure`. When running a command under its old name, the command will issue a deprecation warning. This warning can be disabled by setting `client.deprecationwarning = false` in the `mam-client.conf` file.

If you are migrating the database, you can install the new version of Moab Accounting Manager with a new installation directory, port and database name by specifying `--prefix`, `--with-port` and `--with-db-name` configuration options that are different from the old version. This would allow you to run both the new and old versions simultaneously.

```
[mam]$ ./configure --with-key=<MAMSecretKey>
```

- To compile the program, type `make`.

```
[mam]$ make
```



**i** If you only need to install the clients on a particular system, replace `make` with `make clients-only`. If you only need to install the web GUI on a particular system, replace `make` with `make gui-only`.

10. Run `make install` as root to install Moab Accounting Manager.

```
[mam]$ su -c "make install"
```

**i** If you only need to install the clients on a particular system, replace `"make install"` with `"make install-clients-only"`. If you only need to install the web GUI on a particular system, replace `"make install"` with `"make install-gui-only"`.

11. If you are migrating your database and used the `--with-db-name` option to specify a new database name that does not already exist, you must create and populate the database from the dump.

a. Create the new database.

- PostgreSQL database.

```
[postgres]$ psql
create database "<new_db_name>";
```

- MySQL database.

```
[root]# mysql
create database '<new_db_name>';
```

b. Import the old data into the new database.

- PostgreSQL database.

```
[mam]$ psql -U <mam_database_user> -W <new_db_name> < /tmp/<old_db_name>.sql
```

- MySQL database.

```
[mam]$ mysql -u <mam_database_user> -p <new_db_name> < /tmp/<old_db_name>.sql
```

12. Edit the configuration files as necessary. You may want to compare your existing configuration files with those distributed with the new release to determine if you want to merge and change any of the new options within your configuration files.

- If you are upgrading MAM from a version *prior* to 9.0, the install process will have saved your prior configuration files to `{goldd,gold,goldg}.conf.pre-9.0` and written new default server configuration file as `mam-{server,client,gui}.conf`. You will need to merge any non-default parameters from your prior config files to the new default config files.

```
[mam]$ diff /opt/mam/etc/goldd.conf.pre-9.0 /opt/mam/etc/mam-server.conf
[mam]$ vi /opt/mam/etc/mam-server.conf
[mam]$ diff /opt/mam/etc/gold.conf.pre-9.0 /opt/mam/etc/mam-client.conf
[mam]$ vi /opt/mam/etc/mam-client.conf
[mam]$ diff /opt/mam/etc/goldg.conf.pre-9.0 /opt/mam/etc/mam-gui.conf
[mam]$ vi /opt/mam/etc/mam-gui.conf
```

- If you are upgrading MAM at or after 9.0, merge and change any of the new options supplied in the new default configuration files (saved in `mam-{server,client,gui}.conf.dist`) into your existing configuration files (`mam-{server,client,gui}.conf`).

```
[mam]$ diff /opt/mam/etc/mam-server.conf /opt/mam/etc/mam-server.conf.dist
[mam]$ vi /opt/mam/etc/mam-server.conf
[mam]$ diff /opt/mam/etc/mam-client.conf /opt/mam/etc/mam-client.conf.dist
[mam]$ vi /opt/mam/etc/mam-client.conf
[mam]$ diff /opt/mam/etc/mam-gui.conf /opt/mam/etc/mam-gui.conf.dist
[mam]$ vi /opt/mam/etc/mam-gui.conf
```

Verify that your current path points to your newly installed clients and server.

```
[mam]$ which mam-server
/opt/mam/sbin/mam-server
```

### 13. Start the server daemon back up.

```
[mam]$ mam-server
```

14. If you are migrating your database to 9.0, you will do so by running one or more migration scripts. You must run every incremental migration script between the version you are currently using and the new version (9.0). These scripts are designed to be rerunnable, so if you encounter a failure, resolve the failure and rerun the migration script. If you are unable to resolve the failure and complete the migration, contact Support.

For example, if you are migrating from Moab Accounting Manager version 7.2, you must run five migration scripts: the first to migrate the database schema from 7.2 to 7.3, the second to migrate from 7.3 to 7.5, the third to migrate the database schema from 7.5 to 8.0, the fourth to migrate the database schema from 8.0 to 8.1, and the fifth to migrate the database schema from 8.1 to 9.0.

```
[mam]$ sbin/migrate_7.2-7.3.pl
[mam]$ sbin/migrate_7.3-7.5.pl
[mam]$ sbin/migrate_7.5-8.0.pl
[mam]$ sbin/migrate_8.0-8.1.pl
[mam]$ sbin/migrate_8.1-9.0.pl
```

### 15. Verify that the resulting database schema version is 9.0.

```
[mam]$ mam-shell System Query
```

Name	Version	Description
Moab Accounting Manager	9.0	Commercial Release

## 16. Verify that the executables have been upgraded to 9.0.0.

```
[mam]$ mam-server -v
Moab Accounting Manager version 9.0.0
```

## Upgrading Moab Web Services

This topic provides instructions to upgrade Moab Web Services to the latest release version. Depending on which version of MWS you are presently running, upgrade instructions may vary.

**i** You must deploy Moab Web Services on the *same* host as Moab Server (Moab Server Host). For documentation clarity, these instructions refer to the shared host for Moab Server and MWS as the MWS Host.

## Before You Upgrade

Before upgrading MWS, Adaptive Computing recommends you upgrade to Java 8 and MongoDB 2.4.x.

## Upgrade to Java 8

**i** Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run MWS.

If you wish to upgrade to Java 8, refer to the [Install Java on page 51](#) instructions.

## Upgrade the MongoDB

**!** 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) (<http://www.mongodb.org/display/DOCS/Backups>).

On the host where the MWS MongoDB database resides, do the following:

1. Check your MongoDB version.

```
[root]# mongo --version
```

2. If your MongoDB version is prior to 2.4, upgrade the database. When upgrading, you must add 'exclude=mongodb-org mongodb-org-server' to the `/etc/yum.repos.d/mongodb.repo` file to maintain 2.4.x. Depending on

your MongoDB version, the file path may be `/etc/yum.repos.d/10gen.repo`.

```
[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
```

## Upgrade Moab Web Services

**i** These instructions also include instructions for Tomcat. Depending on your OS, the Tomcat file or user names may not contain the version distinction (i.e. tomcat6).

On the MWS Server Host, do the following:

1. Create a directory for which you will extract the contents of the MWS download tarball. For example:

```
[root]# mkdir /tmp/mws-install
[root]# cd /tmp/mws-install
```

2. 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, 9.0.0.

3. In the directory you created earlier, extract the contents of the MWS download tarball and then change directory into the extracted directory. For example:

```
[root]# tar xvfz mws-9.0.0.tar.gz
[root]# cd /tmp/mws-install/mws-9.0.0
```

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

- Red Hat 6-based systems

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

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

```
[root]# systemctl stop tomcat.service
[root]# cp mws.war /usr/share/tomcat/webapps
[root]# rm -rf /usr/share/tomcat/webapps/mws
```

5. Create the MWS home directory and subdirectories. See [Configuration](#) in the *Moab Web Services Reference Guide* for more information.

**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/utls
[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/utls
```

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

```
[root]# cd utls
[root]# cp * /opt/mws/utls
[root]# chown tomcat:tomcat /opt/mws/utls/*
```

7. Merge the changes in the `/tmp/mws-install/mws-9.0.0/mws-config.groovy` file into your existing `/opt/mws/etc/mws-config.groovy`. Depending on your current MWS version, do the following as needed:
  - If Insight is part of your configuration, add the Insight configuration information (`insight.username`, `insight.password`, `insight.url`); prior version to 9.0
  - If Viewpoint is part of your configuration, register Viewpoint as client; prior to version 9.0
  - Change the `moab.messageQueue.port` to 5570; prior to version 8.0
  - Configure and appender for the audit log; prior to version 8.0
  - Change the layout to `"new com.ace.mws.logging.ACPatternLayout()"` for the output format of each log entry; prior to version 8.0
  - Remove the `mws.suite` parameter and the `mam.*` parameters (they have been moved to `/opt/mws/etc/mws.d/`); prior to version 8.0
  - Confirm the value for `moab.messageQueue.secretKey` matches 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](#).

The following is an example of the merged `mws-config.groovy` file for MWS 9.0:

```
// Any settings in this file may be overridden by any
// file in the mws.d directory.

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

// Moab Workload Manager configuration.
moab.secretKey = "<ENTER-KEY-HERE>"
moab.server = "localhost"
moab.port = 42559

// MongoDB configuration.
// grails.mongo.username = "mws user"
// grails.mongo.password = "<ENTER-KEY-HERE>"

// Insight configuration.
// dataSource_insight.username = "mws"
// dataSource_insight.password = "changeme!"
// dataSource_insight.url = "jdbc:postgresql://127.0.0.1:5432/moab_insight"

// Message bus configuration.
moab.messageQueue.port = 5570
// moab.messageQueue.secretKey = "<ENTER-KEY-HERE>"
mws.messageQueue.address = "*"
mws.messageQueue.port = 5564

// Sample OAuth Configuration
grails.plugin.springsecurity.oauthProvider.clients = [
    [
        clientId:"viewpoint",
        clientSecret:"<ENTER-CLIENTSECRET-HERE>",
        authorizedGrantTypes:["password"]
    ]
]

// Sample LDAP Configurations

// Sample OpenLDAP Configuration
//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"

// Sample Active Directory Configuration
//ldap.server = "192.168.0.5"
//ldap.port = 389
//ldap.baseDNs = ["CN=Users,DC=acme,DC=com", "OU=Europe,DC=acme,DC=com"]
//ldap.bindUser = "cn=Administrator,cn=Users,DC=acme,DC=com"
//ldap.password = "*****"
//ldap.directory.type = "Microsoft Active Directory"

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\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 the rest
    layout: new com.ace.mws.logging.ACPatternLayout(), //Configures the
format of each log entry
    threshold: org.apache.log4j.Level.ERROR //Ignore any logging entries more
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
debug    'com.ace.mws',
        'grails.app.conf.BootStrap',
        'grails.app.controllers.com.ace.mws',
        'grails.app.domain.com.ace.mws',
        'grails.app.filters.com.ace.mws',
        'grails.app.services.com.ace.mws',
        'grails.app.tagLib.com.ace.mws',
        'grails.app.jobs.com.ace.mws',
        'grails.app.gapiParser',
        'grails.app.gapiRequest',
        'grails.app.gapiSerializer',
        'grails.app.translator',
        'plugins' // MWS plugins

info     'com.ace.mws.gapi.Connection',
        'com.ace.mws.gapi.parsers',
        'grails.app.service.grails.plugins.reloadconfig',
        'com.ace.mws.gapi.serializers'

off      'org.codehaus.groovy.grails.web.errors'


// 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'
}

```

8. 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`.



9. Verify the Tomcat user has read access to the `/opt/mws/etc/mws-config.groovy` and `/opt/mws/etc/mws.d/mws-config-hpc.groovy` file.
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.

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.

11. Start Tomcat.

- Red Hat 6-based systems

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

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

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

12. 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.

13. 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`.)

 If you encounter problems, or if MWS does not seem to be running, see the steps in [Moab Web Services Issues on page 176](#).

## Migrating the MAM Database from MySQL to PostgreSQL

PostgreSQL is 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.

- **Manual Install** - [Installing Moab Web Services on page 49](#)
- **RPM Install** - [Installing Moab Web Services on page 118](#)

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/mam-
server.conf.dist file
[mam]$ vi /opt/mam/etc/mam-server.conf # Merge in the database.datasource from
etc/mam-server.conf.dist
```

11. Restart Moab Accounting Manager.

```
[mam]$ mam-server -r
```



## Chapter 3 RPM installation

This chapter provides installation, configuration, and upgrading information using the RPM Installation method.

**i** The RPM Installation method only supports installation on CentOS 6.x, 7.x, RHEL 6.x, 7.x, or Scientific Linux 6.x, 7.x. See [Chapter 2 Manual Installation on page 19](#) 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.

### Related Topics

- [Chapter 1 Planning your Installation on page 3](#)
- [Preparing the Host for RPM Installations on page 98](#)

# Installation

In this section:

- [Preparing the Host for RPM Installations on page 98](#)
- [Installing Torque Resource Manager on page 100](#)
- [Installing Moab Workload Manager on page 104](#)
- [Installing Moab Accounting Manager on page 107](#)
- [Installing Moab Web Services on page 118](#)
- [Installing Moab Insight on page 127](#)
- [Installing Moab Viewpoint on page 139](#)

## Preparing the Host for RPM Installations

Adaptive Computing provides RPMs to install the various component servers (such as Moab Server, MWS Server, Torque Server).

Depending on your configuration, you may install many servers on a single host, or a single server on its own host. In addition, you can install various clients and GUIs on the same host you installed the server or on another host. For example, you can have the Moab Server and the MWS Server on the same host but you install the Torque Server on a different host.

**i** Be aware that the same host may be called by different names. For example, if you installed the Moab Server and the MWS Server on the same host, the MWS instructions will call it the MWS Host, not the Moab Host.

## Set Up Proxies

If your site uses a proxy to connect to the internet, configure yum to use a proxy by editing the `/etc/yum.conf` file as follows:

```
proxy=http://<proxy_server_id>:<port>
```

In addition, if your site uses an external repository to install python dependencies (for example, the host where you install Viewpoint might need to download extra packages), you will need to set up pip to use a proxy. Do the following:

```
export http_proxy=http://<proxy_server_id>:<port>
export https_proxy=http://<proxy_server_id>:<port>
```

## Enable the Adaptive Computing Package Repository

**i** Whether you are installing RPMs on one host or on several hosts, each host (physical machine) on which a server is installed (Torque Server Host, Moab Server Host, etc) *must* have the Adaptive Computing Package Repository enabled. It is not necessary to enable the Adaptive Computing repository on the Torque MOM Hosts or client hosts.

Do the following:

1. Download the latest 9.0.0 RPM suite tarball (moab-hpc-enterprise-suite-*<version>*-*<timestamp>*-*<OS>*.tar.gz, for example) from the [Adaptive Computing](#) website.

2. Untar the downloaded package.

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

3. Change directories into the untarred directory.

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

4. 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: 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, 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.

## 5. 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-accounting-manager.x86_64 : Moab Accounting Manager for Moab HPC Suite
moab-hpc-enterprise-suite.noarch : Moab HPC Enterprise Suite virtual package
moab-insight.x86_64 : Moab Insight
moab-perl-RRDs.noarch : Moab RRDs
moab-tomcat-config.x86_64 : Tomcat Configuration 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
...
```

## Related Topics

- [Chapter 1 Planning your Installation on page 3](#)
- [Chapter 3 RPM installation on page 97](#)

# Installing Torque Resource Manager



If you intend to use Torque Resource Manager 6.0.0 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque 6.0 functionality requires Moab 9.0 or later.

This topic contains instructions on how to install, configure, and start Torque Resource Manager (Torque).

In this topic:

- [Prerequisites on page 101](#)
- [Install Torque Server on page 102](#)



- [Install Torque MOMs on page 103](#)
- [Configure Data Management on page 104](#)

## Prerequisites

In this section:

- [Open Necessary Ports on page 101](#)
- [Verify the hostname on page 102](#)

## 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* for more information.

If you have a firewall enabled, do the following:

### 1. On the Torque Server Host:

- Red Hat 6-based systems using iptables

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

# Add the following line immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"
-A INPUT -p tcp --dport 15001 -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 --reload
```

## 2. On each Torque MOM Host (Compute Hosts):

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

## Verify the hostname

On the Torque Server Host, confirm 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 Torque Server

On the Torque Server Host, do the following:

1. If you are installing the Torque Server on its own host (recommend) and *not* on the same host where you installed another server (such as Moab Server), verify you completed the steps to prepare the host. See [Preparing the Host for RPM Installations on page 98](#) for more information.
2. Install the Torque Server RPM.

```
yum install moab-torque-server
```

3. Add the hostnames of your Torque MOMs (which is commonly all of your compute nodes) to the `/var/spool/torque/server_priv/nodes` file. You can remove the hostname entry for the Torque server node *unless* you will be running a Torque MOM daemon on this host. See *Managing Nodes in the Torque Resource Manager Administrator Guide* for information on syntax and options for specifying compute nodes.

Example:

```
vi /var/spool/torque/server_priv/nodes

node01 np=16
node02 np=16
...
```

#### 4. Start the Torque server.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

## Install Torque MOMs

In most installations, you will install a Torque MOM on each of your compute nodes.

Do the following:

1. From the Torque Server Host, copy the `moab-torque-common` and `moab-torque-mom` RPM files to each MOM node. It is also recommended that you install the `moab-torque-common` RPM so you can use client commands and submit jobs from compute nodes.

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

2. On each Torque MOM Host, install the RPMs; `moab-torque-common` is installed *first*.

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

3. On each Torque MOM Host, 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. On each Torque MOM Host, edit the `/var/spool/torque/mom_priv/config` file. This file is identical for all compute nodes and can be created on the Torque Server 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 MOM Host, 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
```

6. If you installed the Torque Client RPM on the MOMs, then on each Torque MOM Host, start the `trqauthd` daemon.

- Red Hat 6-based systems.

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

- Red Hat 7-based system.

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

## Configure Data Management

When a batch job completes, `stdout` and `stderr` files are generated and placed in the spool directory on the master Torque MOM Host for the job instead of the submit host. You can configure the Torque batch environment to copy the `stdout` and `stderr` files back to the submit host. See *Configuring Data Management* in the *Torque Resource Manager Administrator Guide* for more information.

Related Topics

[Chapter 3 RPM installation on page 97](#)

## Installing Moab Workload Manager

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

In this topic:

- [Open Necessary Ports on page 104](#)
- [Install Moab Server on page 105](#)
- [Configure Torque to Trust Moab on page 107](#)
- [Verify the Installation on page 107](#)

### Open Necessary Ports

Moab 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 Host, 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.

On the Moab Server Host, 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 firewallld

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

## Install Moab Server

On the *Moab* Server Host do the following:

1. If you have not already done so, complete the steps to prepare the Moab Server Host. See [Preparing the Host for RPM Installations on page 98](#).

2. Install RPM packages.

- a. Install the Moab Server RPMs.

```
[root]# yum install moab-workload-manager moab-workload-manager-hpc-configuration
```

- b. If you are using Torque as a resource manager and installed the Torque Server on a separate host (Torque Server Host; recommended) from the Moab Server (Moab Server Host), you will need to install the Torque client RPM on the Moab Server Host in order for Moab to interact with Torque.

```
[root]# yum install moab-torque-client
```

- c. If you are using Moab Accounting Manager and will be using the Native (custom script) accounting manager interface, and are installing the Moab Accounting Manager Server on a separate host from the Moab Server (Moab Server Host) you will need to the install Moab Accounting Manager client on the Moab Server Host in order for the custom scripts to use the MAM API.

```
[root]# yum install moab-accounting-manager
```

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

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

4. 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](#).

5. If you are using Torque as your resource manager and you installed the Torque Server on a separate host (Torque Server Host) from the Moab Server (Moab Server Host), do the following:

- a. 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
```

- b. Verify that the Torque Server hostname used is *exactly* the name returned by a reverse hostname lookup.

```
[root]# cat /var/spool/torque/server_name | perl -lpe '$_=(gethostbyname($_))[0]
```

If different, take the necessary steps to make them match. For example, it may be necessary to add the Torque Server hostname to the `/etc/hosts` file on the Moab Server Host.

```
[root]# vi /etc/hosts

<Torque_server_ip_address> <Torque_server_hostname> <Torque_server_FQDN>
```

## c. Start the trqauthd daemon.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

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

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

## Configure Torque to Trust Moab

If you are using Torque as a resource manager and you installed the Torque Server on a separate host (Torque Host); recommended, do the following:

- On the *Torque* Host, add the name of the Moab Server Host (where Moab Server is installed) as a manager, and submit the host.

```
[root]# qmgr
Qmgr: set server managers += root@<moab_server_hostname>
Qmgr: set server submit_hosts += <moab_server_hostname>
Qmgr: exit
```

## Verify the Installation

If you have a resource manager configured, verify that the scheduler is able to schedule a job.

- 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

[Chapter 3 RPM installation on page 97](#)

## Installing Moab Accounting Manager

This topic contains instructions on how to install, configure, and start Moab Accounting Manager (MAM).

Perform the following in order:

- [Plan Your Installation](#)
- [Confirm Requirements](#)
- [Open Necessary Ports](#)
- [Install Dependencies, Packages, or Clients](#)
- [Install MAM Server](#)
- [Configure the MAM GUI](#)
- [Access the MAM GUI](#)
- [Configure Moab Workload Manager to use Moab Accounting Manager](#)
- [Initialize Moab Accounting Manager](#)

## Plan Your Installation

The first step is determining the number of separate hosts (physical machines) required for your MAM installation.

Your MAM installation includes:

- MAM Server
- MAM Database
- MAM GUI (optional)
- MAM Clients (possibly several hosts)

Each of these components can be installed on their own hosts (meaning the actual physical machine) or can be combined on same hosts. For example, the MAM Database can be installed on the same *host* as the MAM Server. Or the MAM Server may be installed on the same host you installed the Moab Server.

Once you have determined which components are installed on which hosts, complete the rest of the instructions for the MAM installation.

**i** The instructions that follow in this topic will use the term Host after each component to reflect installing on a host (again, meaning the physical machine). For example, MAM Server Host and MAM Database Host. Depending on your configuration, Host may refer to as installed on its own machine or installed on the same machine as another component.

## Confirm Requirements

In this section:

- [Hardware Requirements on page 109](#)
- [Supported Operating Systems on page 109](#)
- [Supported Databases on page 109](#)



## Hardware Requirements

- Dual or Quad core Intel/AMD x86-64 processor
- At least 8 GB of RAM
- 1-2 TB disk space

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

## Supported Operating Systems

MAM has been tested on the following variants of Linux:

- CentOS (6.x, 7.x)
- RHEL (6.x, 7.x)
- Scientific Linux (6.x, 7.x)

## Supported Databases

MAM uses an RDBMS as a back end. If this is a new installation, use the following database:

- PostgreSQL 7.2 or higher

## Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

Do the following as needed.

1. If you will be installing the MAM Server on a separate host than you installed the Moab Server *or* you will be installing the MAM Clients on other hosts, then on the MAM Server Host, open the MAM Server port (7112) in the firewall.
  - 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"
-A INPUT -p tcp --dport 7112 -j ACCEPT
[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=7112/tcp --permanent
[root]# firewall-cmd --reload
```

2. If using the MAM GUI, then on the MAM GUI Host, open the https port in the firewall for secure browser communication.

- 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"
-A INPUT -p tcp --dport 443 -j ACCEPT

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

- Red Hat 7-based systems using firewalld

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

3. If you will be installing the MAM Database on a different host from the MAM Server, then on the MAM Database Host, open the postgres port (5432) in the firewall.

- 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"
-A INPUT -p tcp --dport 5432 -j ACCEPT

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

- Red Hat 7-based systems using firewalld

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

## Install Dependencies, Packages, or Clients

In this section:

- [Install and Initialize PostgreSQL Server on page 111](#)
- [Install Perl ReadLine \(Optional\) on page 112](#)

## Install and Initialize PostgreSQL Server

Moab Accounting Manager uses a database for transactions and data persistence. The PostgreSQL database may be installed on a separate host from the MAM Server; however, it is often convenient to install them on the same host.

On the MAM Database Host, do the following:

### 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.

Edit or add a "host" line in the pg\_hba.conf file for the interface from which the MAM Server 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

# Replace 127.0.0.1 with the IP address of the MAM Server Host if the
# MAM PostgreSQL server is on a separate host from the MAM server.
host    all             all             127.0.0.1/32          md5
host    all             all             ::1/128              md5
---
```

### 3. If the MAM Database Host is installed on a *separate* host from where you will install the MAM Server, configure PostgreSQL to accept connections from the MAM Server Host.

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

# Replace <mam-server-host> with the interface name from which the MAM server
# will be connecting to the database.
listen_addresses = '<mam-server-host>'
---
```

### 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
```

## Install Perl ReadLine (Optional)

Moab Accounting Manager can be optionally configured to provide command history editing functionality in the mam-shell command.

The perl-Term-ReadLine-Gnu package is required; however, depending on your Red Hat OS version, it may not be already installed as a prerequisite package. You can install the RPM from a reputable third-party rpm site, or install from CPAN.

- Red Hat 6-based systems

No action needed. perl-Term-ReadLine-Gnu is installed as a prerequisite package.

- Red Hat 7.0-based systems

```
[root]# cpan Term::ReadLine::Gnu
```

- Red Hat 7.1-based systems

```
[root]# yum install perl-Term-ReadLine-Gnu
```

## Install MAM Server

On the MAM Server Host do the following:

1. If you are installing the MAM Server on its own host and *not* on the same host where you installed another server (such as Moab Server), verify you completed the steps to prepare the host. See [Preparing the Host for RPM Installations on page 98](#).
2. Install the MAM Server RPM.

```
yum install moab-accounting-manager
```

**i** If installing on RHEL, some packages may not be found in the standard RHEL distribution repositories.

- One way to overcome this problem is to install the missing dependencies from a CentOS repository using a reputable rpm download site (such as rpmfind.net or pbone.net). For example (for RHEL 7.1):

```
[root]# cat > /etc/yum.repos.d/centos.repo <<'EOF'
[centos]
name=CentOS $releasever - $basearch
baseurl=http://rpmfind.net/linux/centos/7.1.1503/os/$basearch/
enabled=0
gpgcheck=0
EOF

[root]# yum install --enablerepo=centos moab-accounting-manager
```

- *Alternatively*, you can install the missing prerequisites from CPAN and then rerun the install.

```
[root]# yum install perl-CPAN
[root]# cpan Crypt::CBC Crypt::DES
[root]# yum install moab-accounting-manager
```

3. As the database user, create a database called `mam` and grant database privileges to the `mam` user.

**i** PostgreSQL was installed and initialized earlier in this topic. See [Install and Initialize PostgreSQL Server on page 111](#).

```
[root]# su - postgres
[postgres]$ psql

create database mam;
create user mam with password 'changeme!';
\q

[postgres]$ exit
```

The *password* you define must be synchronized with the `database.password` value in `/opt/mam/etc/mam-server.conf`.

```
[root]# vi /opt/mam/etc/mam-server.conf

database.password = changeme!
```

4. Edit the `mam-site.conf` file and set a secure secret key to authenticate MAM requests. The following example shows how to generate a random base64-encoded string to use as the secret key.

```
[root]# dd if=/dev/urandom count=1 bs=18 2>/dev/null | openssl enc -base64
[root]# vi /opt/mam/etc/mam-site.conf

token.value = A9eszfd06Kib1ntTdeEg7ldP
```

5. Run the `hpc.sql` script to populate the Moab Accounting Manager database with objects, actions, and attributes necessary to function as an Accounting Manager.

```
[root]# su - mam

[mam]$ psql mam < /usr/share/moab-accounting-manager/9.0.0/hpc.sql
[mam]$ exit
```

6. Start the `mam` service.

- Red Hat 6-based systems

```
[root]# chkconfig --add mam
[root]# service mam start
```

- Red Hat 7-based systems

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

## Configure the MAM GUI

If you plan to use the web GUI, then on the MAM GUI Host, do the following:

1. As `root`, add or edit the SSL virtual host definition as appropriate for your environment. To do so, configure the `cgi-bin` directory in `ssl.conf`. Below the `cgi-bin` directory element, create an alias for `/cgi-bin` pointing to your `cgi-bin` directory. If you chose to install to a `cgi-bin` sub-directory, you might want to create an alias for that as well. Also, add `index.cgi` to the `DirectoryIndex` so you can use the shorter sub-directory name.

```
[root]# vi /etc/httpd/conf.d/ssl.conf

<Directory "/var/www/cgi-bin">
## Add these lines
Options ExecCGI
AddHandler cgi-script .cgi
AllowOverride All
Order allow,deny
Allow from all
</Directory>

# Aliases for /cgi-bin
Alias /cgi-bin/ /var/www/cgi-bin/
Alias /mam /var/www/cgi-bin/mam/

# Make shorter sub-dir name available
DirectoryIndex index.cgi
```

2. If Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow the web server to make network connections,

use `setuid` for authentication, and write to the log file.

a. Determine the current mode of SELinux.

```
[root]# getenforce
Enforcing
```

- If the command returns a mode of **Disabled** or **Permissive**, or if the `getenforce` command is not found, you can skip the rest of this step.
- If the command returns a mode of **Enforcing**, you can choose between options of customizing SELinux to allow the web GUI to perform its required functions or disabling SELinux on your system.

b. If you choose to customize SELinux, do the following:

**i** SELinux can vary by version and architecture and that these instructions may not work in all possible environments.

**i** If you used the `--prefix=<prefix>` configuration option when you configured Moab Accounting Manager, you must replace references to `/opt/mam` in the example below with the `<prefix>` you specified. See [Moab Accounting Manager Configuration Options on page 69](#).

• Red Hat 6-based systems

```
[root]# cat > mamgui.te <<EOF
module mamgui 1.0;
require {
    type httpd_sys_script_t;
    type port_t;
    class capability setuid;
    class tcp_socket name_connect;
}
allow httpd_sys_script_t port_t:tcp_socket name_connect;
allow httpd_sys_script_t self:capability setuid;
EOF
[root]# checkmodule -M -m -o mamgui.mod mamgui.te
[root]# semodule_package -m mamgui.mod -o mamgui.pp
[root]# semodule -i mamgui.pp
[root]# setenforce 0
[root]# chcon -v -t httpd_sys_content_t /opt/mam/log
[root]# setenforce 1
```

- Red Hat 7-based systems

```
[root]# yum install checkpolicy policycoreutils-python
[root]# cat > mamgui.te <<EOF
module mamgui 1.0;
require {
    type httpd_sys_script_t;
    type unreserved_port_t;
    class tcp_socket name_connect;
}
allow httpd_sys_script_t unreserved_port_t:tcp_socket name_connect;
EOF
[root]# checkmodule -M -m -o mamgui.mod mamgui.te
[root]# semodule_package -m mamgui.mod -o mamgui.pp
[root]# semodule -i mamgui.pp
[root]# setenforce 0
[root]# chcon -v -t httpd_sys_rw_content_t /opt/mam/log
[root]# setenforce 1
```

3. For the highest security, it is recommended that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS 7 is documented at [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/System\\_Administrators\\_Guide/ch-Web\\_Servers.html#s2-apache-mod\\_ssl](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Web_Servers.html#s2-apache-mod_ssl).

Alternatively, if your network domain can be secured from man-in-the-middle attacks, you could use a self-signed certificate. Often this does not require any additional steps since in many distributions, such as Red Hat, the Apache SSL configuration provides self-signed certificates by default.

4. Start or restart the HTTP server daemon.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

## Access the MAM GUI

If you plan to use the web GUI, then on the MAM Server Host, do the following:

1. Create a password for the `mam` user to be used with the MAM Web GUI.

```
[root]# su - mam
[mam]$ mam-set-password
```

2. Verify the connection.

- a. Open a web browser and navigate to `https://<mam-server-host>/mam`.



- b. Log in as the `mam` user with the password you set in step 1.

## Configure Moab Workload Manager to use Moab Accounting Manager

Do the following, where applicable:

1. On the *Moab* Server Host, edit the Moab configuration file.

```
[root]# vi /opt/moab/etc/moab.cfg
AMCFG[mam] TYPE=MAM HOST=<mam_server_host>
```

- a. Uncomment the AMCFG lines and customize as needed. See [Accounting, Charging, and Allocation Management](#) in the *Moab Workload Manager Administrator Guide*.
  - b. If the Moab Server and the MAM Server are on the *same* host, set HOST to 'localhost'; otherwise, set HOST to the host name for the MAM Server (MAM Server Host).
2. Configure Moab to authenticate with MAM using the MAM secret key.
    - a. On the *MAM* Server Host, copy the auto-generated secret key from the `token.value` value in the `/opt/mam/etc/mam-site.conf` file.
    - b. On the *Moab* Server Host, add the secret key to the `moab-private.cfg` file as the value of the CLIENTCFG KEY attribute.

```
[root]# vi /opt/moab/etc/moab-private.cfg
CLIENTCFG[AM:mam] KEY=<MAMSecretKey>
```

3. Restart Moab

- Red Hat 6-based systems

```
[root]# service moab restart
```

- Red Hat 7-based systems

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

## Initialize Moab Accounting Manager

You will need to initialize Moab Accounting Manager to function in the way that is most applicable to the needs of your site. See [Initial Setup](#) in the *Moab Accounting Manager Administrator Guide* to set up Moab Accounting Manager for your desired accounting mode.

Related Topics

[Chapter 3 RPM installation on page 97](#)

## Installing Moab Web Services



You must deploy Moab Web Services on the *same* host as Moab Server (Moab Server Host). For documentation clarity, these instructions refer to the shared host for Moab Server and MWS Server as the MWS Server Host.

This topic contains instructions on how to install, configure, and start Moab Web Services (MWS).

In this topic:

- [Open Necessary Ports on page 118](#)
- [Install Dependencies, Packages, or Clients on page 120](#)
- [Install MWS Server on page 122](#)
- [Verify the Installation on page 126](#)

### Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

In this section:

- [Open the Tomcat Port \(8080\) on page 118](#)
- [Open the MWS MongoDB Database Port \(27017\) on page 119](#)

### Open the Tomcat Port (8080)

On the MWS Server Host, 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"

-A INPUT -p tcp --dport 8080 -j ACCEPT

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

- Red Hat 7-based systems using firewallld

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

- SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="8080"
[root]# service SuSEfirewall2_setup restart
```

## Open the MWS MongoDB Database Port (27017)

**i** Depending on your system configuration, your MongoDB databases may not be installed on the same host as their corresponding component servers. For example, you may choose to install the MWS MongoDB database on the same host where you have installed other MongoDB databases instead of on the MWS Server Host.

Do the following, as needed:

- If you have chosen to install the MWS MongoDB database on the *same* host you installed other MongoDB databases (for example, the same host you installed the Moab MongoDB database), confirm the firewall port (27017) is already opened on that host.
- If you have chosen to install the MWS MongoDB database on a *separate* host from other MongoDB databases, you will need to open the MWS MongoDB database port in firewall for that host. To open the port in the firewall, do the following:

- Red Hat 6-based systems using iptables

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

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

-A INPUT -p tcp --dport 27017 -j ACCEPT

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

- Red Hat 7-based systems using firewallld

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

- SUSE 11-based or SUSE 12-based systems using SuSEfirewall2 (SUSE 11 is not supported on the MWS Server Host)

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="27017"
[root]# service SuSEfirewall2_setup restart
```

## Install Dependencies, Packages, or Clients

In this section:

- [Install Java on page 120](#)
- [Install MongoDB on page 120](#)

### Install Java

Install the Linux x64 RPM version of Oracle® Java® 8 Runtime Environment.

**i** Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run MWS.

On the MWS Server Host, do the following:

1. Install the Linux x64 RPM version of Oracle Java SE 8 JRE.
  - a. Go to the [Oracle Java download page](http://java.com/en/download/linux_manual.jsp) ([http://java.com/en/download/linux\\_manual.jsp](http://java.com/en/download/linux_manual.jsp)).
  - b. Copy the URL for the Linux x64 RPM version, and run the following command:

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

### Install MongoDB

To install and enable MongoDB, on the MWS Host, do the following:

1. Install `mongo-10gen-server`.

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

2. Start MongoDB.

**i** There may be a short delay (approximately 3 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
```

### 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/) (<http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication/>) for more information.

## b. 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
```

## Install MWS Server

On the MWS Host, do the following:

## 1. Install the MWS RPMs.

```
[root]# yum install moab-web-services moab-web-services-hpc-configuration
```

## 2. 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 on the same host.

```
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 on page 120](#) earlier in this topic.

```
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]# mdia -S | grep Mongo

Mongo connection (localhost) is up (credentials are set)
```

## • Red Hat 7-based systems

```
[root]# systemctl restart moab.service
[root]# mdia -S | grep Mongo

Mongo connection (localhost) is up (credentials are set)
```

## 3. 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. Generate a key and store the result in `/opt/moab/etc/.moab.key`.

## • Red Hat 6-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 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
```

## b. (Optional) Moab supports message queue security using AES. This feature requires a Base64-encoded 16-byte (128-bit) shared secret.

a. Generate a key and append the result to `/opt/moab/etc/moab-private.cfg`.

## • Red Hat 6-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 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
```

**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.

- b. 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)
```

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

- a. 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 you generated earlier.

**!** 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.

- b. If you are using Moab Accounting Manager, change these settings in `/opt/mws/etc/mws.d/mws-config-hpc.groovy`:

- **mam.secretKey**: needs to match the MAM secret key in `/opt/mam/etc/mam-site.conf` on the MAM Server (as **token.value**)
- **mam.server**: set to the hostname of the MAM Server
- **mam.port**: set to the port of the MAM Server

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

mam.secretKey = "<ENTER-KEY-HERE>"
mam.server = "localhost"
mam.port = 7112
```

- c. Do *one* of the following:

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

```
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.*



If you followed the Adaptive Computing tutorial, [Setting Up OpenLDAP on CentOS 6 on page 150](#), 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](#).



To see how to configure a secure connection to the LDAP server, see [Securing the LDAP Connection](#).

- 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"
```

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



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



For more information about PAM configuration with MWS, see [PAM \(Pluggable Authentication Module\) Configuration Using mws-config.groovy](#).



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.

- 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.

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

## 5. 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
```

## Verify the Installation

1. Open a web browser.
2. Navigate to `http://<server>:8080/mws/`. You will see some sample queries and a few other actions.
3. 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 [Chapter 4 Troubleshooting on page 173](#).

#### Related Topics

- [Chapter 3 RPM installation on page 97](#)
- [Installing Moab Workload Manager on page 104](#)

## Installing Moab Insight

This topic contains instructions on how to install Moab Insight (Insight).

Because Insight accumulates data for one cluster at a time, one Insight Server (daemon) should service one Moab instance. However, you can configure PostgreSQL to aggregate data using database replication mechanisms if you desire cross-cluster data.

**i** Moab Workload Manager and Insight both tend to heavily consume system resources. Therefore, Adaptive Computing *requires* that the Insight Server and the Moab Workload Manager Server run on separate hosts. For these installation instructions, the "Moab Server Host" refers to one host and the "Insight Server Host" refers to another host.

#### In this topic:

- [Prerequisites on page 128](#)
- [Dependencies, Packages, or Client Installations on page 130](#)
- [Install Insight on page 135](#)

## Requirements

Removing this section as the information is listed here: [Component Requirements on page 10](#)

It is only still here for now to not throw off the numbering during the review process.

## Prerequisites

In this section:

- [Open Necessary Ports on page 128](#)
- [Verify the hostname on page 130](#)

## Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

In this section:

- [Open the Insight Server Port \(5568\)](#)
- [Open the MongoDB Database Port \(27017\)](#)
- [Open the Moab Server Ports \(5574 and 5575\)](#)

### *Open the Insight Server Port (5568)*

On the Insight Server Host, do the following:

- Red Hat 6-based systems using iptables

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

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

-A INPUT -p tcp --dport 5568 -j ACCEPT

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

- Red Hat 7-based systems using firewallld

```
[root]# firewall-cmd --add-port=5568/tcp --permanent
[root]# firewall-cmd --reload
```

### *Open the MongoDB Database Port (27017)*

**i** Insight also requires access to the Moab MongoDB database. Depending on your system configuration, your MongoDB databases may not be installed on the same host as their corresponding component servers. For example, you may choose to install the Insight MongoDB on the same host where you have installed other MongoDB databases instead of on the Insight Server Host.

Do the following, as needed:

- If you have chosen to install the Insight MongoDB database on the *same* host you installed the Moab MongoDB database, confirm the firewall port (27017) is already opened on that host.
- If you have chosen to install the Insight MongoDB database on a *separate* host from the Moab MongoDB database (for example if you installed the Insight MongoDB database on the Insight Server Host or on still a different host), you will need to make sure that *both* the Moab MongoDB database host and the Insight MongoDB database host have the firewall port (27017) open. To open the port in the firewall, 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"

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

- SUSE 11-based or SUSE 12-based systems using SuSEfirewall2 (not applicable if you choose to install the Insight MongoDB database on the Insight Server Host or Viewpoint Server Host; SUSE 11 is not supported on the shared Moab Server and MWS Server Host [MWS Server Host])

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="27017"

[root]# service SuSEfirewall2_setup restart
```

### Open the Moab Server Ports (5574 and 5575)

On the Moab Server Host, 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"

-A INPUT -p tcp --dport 5574 -j ACCEPT
-A INPUT -p tcp --dport 5575 -j ACCEPT

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

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=5574/tcp --permanent
[root]# firewall-cmd --add-port=5575/tcp --permanent
[root]# firewall-cmd --reload
```

- SUSE 11-based or SUSE 12-based systems using SuSEfirewall2 (SUSE 11 is not supported on the shared Moab Server and MWS Server Host [MWS Server Host])

```
[root]# vi /etc/sysconfig/SuSEfirewall2
FW_SERVICES_EXT_TCP="5574 5575"
[root]# service SuSEfirewall2_setup restart
```

## Verify the hostname

On the Insight Server Host, confirm 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.

## Dependencies, Packages, or Client Installations

In this section:

- [Install Java on page 130](#)
- [Install MongoDB on page 131](#)
- [Install PostgreSQL on page 133](#)

### Install Java

Install the Linux x64 RPM version of Oracle® Java® 8 Runtime Environment.

**i** Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 is also supported.. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run Insight.

On the Insight Server Host, do the following:

1. Install the Linux x64 RPM version of Oracle Java SE 8 JRE.
  - a. Go to the to the [Oracle Java download page](http://java.com/en/download/linux_manual.jsp) (http://java.com/en/download/linux\_manual.jsp).
  - b. Copy the URL for the Linux x64 RPM version and run the following command:

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

## Install MongoDB

**i** The Insight MongoDB may be installed on the Insight Server Host or on different host. If you will install on a different host, and your configuration uses firewalls, open the necessary port. See [Open Necessary Ports on page 128](#)

To install and enable MongoDB, do the following:

1. On the host you have chosen to install the Insight MongoDB database, do the following:

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

```
[root]# yum install mongo-10gen-server --exclude mongodb-org,mongodb-org-server
```

**i** Running `yum upgrade` will replace MongoDB 2.4.x with a more recent, and incompatible version. Consider using `yum version lock` to maintain MongoDB 2.4.x.

- b. Start MongoDB.

**i** There may be a short delay (approximately 3 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
```

2. Add the required MongoDB users to Insight MongoDB and Moab MongoDB; regardless of whether they share a host.

**i** These instructions show password examples (`secret1`, `secret2`, and `secret3`). Choose your own passwords for these users.

- Insight MongoDB

```
[root]# mongo
> use admin;
> db.addUser("admin_user", "secret1");
> db.auth("admin_user", "secret1");
> use insight;
> db.addUser("insight_user", "secret4");
> db.addUser("mws_user", "secret3", true);

> exit
```

- Moab MongoDB

```
[root]# mongo
> use admin;
> db.auth("admin_user", "secret1");
> use moab;
> db.addUser("insight_user", "secret4", true);

> 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/) (<http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication/>) for more information.

### 3. Enable the user authentications in MongoDB.

**i** If Insight MongoDB resides on a different host from Moab MongoDB, you will need to perform this procedure twice (once for each host).

- 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
```

## Install PostgreSQL

**i** The Insight PostgreSQL database may be installed on the Insight Server Host or on different host. If you will install on a different host, and your configuration uses firewalls, open the necessary port. See [Open Necessary Ports](#) on page 128.

On the host you have chosen to install the Insight PostgreSQL database, do the following:

**i** These instructions are for installing PostgreSQL 9.3; however Insight may support other versions. See [Component Requirements](#) for requirements information.

### 1. Install PostgreSQL.

- Red Hat 6-based systems

```
[root]# yum install postgresql93-server
[root]# service postgresql-9.3 initdb
```

- Red Hat 7-based systems

```
[root]# yum install postgresql93-server
[root]# /usr/pgsql-9.3/bin/postgresql93-setup initdb
```

2. Edit the PostgreSQL configuration file to listen for connections on all interfaces. See the documentation provided in the `postgresql.conf` file if you want to be more restrictive.

```
[root]# vi /var/lib/pgsql/9.3/data/postgresql.conf

# Uncomment the listen addresses line in the configuration:
listen_addresses = '*'
```

3. Edit the PostgreSQL host-based authentication (HBA) configuration file to enable TCP connections using encrypted passwords (change `ident` to `md5`) and add a line for each host that needs to connect to the database. See the "ADDRESS" documentation provided in `pg_hba.conf` for details.
  - If you are using MWS, add the IP address of the host on which MWS Server is installed. This is shown in the following example as `<MWS_host_address>`.

**i** Using "0.0.0.0/0" in place of "`<MWS_host_address>`" will allow connections from all hosts.

```
[root]# vi /var/lib/pgsql/9.3/data/pg_hba.conf

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer

# IPv4 local connections:
host    all             all             <MWS_host_address>/32   md5 # If
using MWS
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.

4. Start or restart the PostgreSQL database.

- Red Hat 6-based systems

```
[root]# chkconfig postgresql-9.3 on
[root]# service postgresql-9.3 start
```

- Red Hat 7-based systems

```
[root]# systemctl enable postgresql-9.3.service
[root]# systemctl restart postgresql-9.3.service
```

## Install Insight



These instructions contain steps to edit the `/opt/insight/etc/config.groovy` file. It is recommended that anytime you edit the `config.groovy` file that you first stop Insight, edit the file and then restart Insight.

1. If you have not already done so, complete the steps to prepare the Insight Server Host. See [Preparing the Host for RPM Installations on page 98](#).
2. On the Insight Server Host, install the Insight RPM.

```
[root]# yum install moab-insight
```



If the installation returns the following warning line:

```
warning: rpmts_HdrFromFdno: Header V4 RSA/SHA1
Signature, key ID 952741e1: NOKEY

Retrieving key from file:///opt/adaptive-rpm-
repository/key/GPG_ADAPTIVE_COMPUTING_INC_EL_6_KEY
Importing GPG key 0x952741E1:
Userid: "Adaptive Computing Enterprises, Inc. (EL 6 key)
<info@adaptivecomputing.com>"
From : /opt/adaptive-rpm-repository/key/GPG_ADAPTIVE_
COMPUTING_INC_EL_6_KEY

This is normal. You can safely input y and continue.
```

3. Create the Insight user and PostgreSQL database.



PostgreSQL was installed earlier in this topic. See [Install PostgreSQL on page 133](#).



This instructions show the default changeme! password. Change this password according to your password security process.

- a. Change to the postgres user.

```
su - postgres
```

## b. Do the following:

```
[postgres]$ psql
CREATE USER moab_insight WITH PASSWORD 'changeme!';
CREATE DATABASE moab_insight WITH OWNER=moab_insight;
CREATE DATABASE moab_insight_reference WITH OWNER=moab_insight;
\q
```

c. Initialize the `moab_insight` database. This sets up event triggers so that database schema validation works correctly in Insight.

```
[postgres]$ psql moab_insight -f /opt/insight/db/initialize.sql
```

## d. If you are also using MWS, create and grant permissions for the MWS user to query the database.

i. Log in to the `moab_insight` database.

```
[postgres]$ psql moab_insight
```

## ii. Create the MWS user and grant the MWS user select permissions on all tables in the public schema.

```
CREATE USER mws WITH PASSWORD 'changeme!';
GRANT SELECT ON ALL TABLES IN SCHEMA public TO mws;
```

iii. Connect to the `moab_insight` database as the `moab_insight` user.

```
\connect moab_insight moab_insight localhost
```

## iv. Grant the MWS user select permissions on all tables in the public schema.

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO mws;
\q
[postgres]$ exit
```

## 4. If you are using MWS, on the MWS Server Host:

a. Add or edit the `dataSource.insight.url` parameter in the `mws-config.groovy` file to specify the JDBC database connection information for the Insight Server. See [Configuration](#) in the *Moab Web Services Reference Guide* for more information on the MWS configuration properties.

**i** In the following example, `<insight-server_ip_address>` represents the DNS name for the Insight Server. This example also uses the default PostgreSQL port number (5432).

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

dataSource_insight.url = "jdbc:postgresql://<insight_server_ip_
address>:5432/moab_insight"
dataSource_insight.password = "<postgres_mws_user_password>"
```

## b. Restart Tomcat.

- Red Hat 6-based systems.

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

- Red Hat 7-based systems.

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

5. On the host you have chosen to install the Insight MongoDB database, configure Insight's connection to the MongoDB database. Add or edit the `/opt/insight/etc/config.groovy` as follows:

```
mongo.username="insight_user"
mongo.password="secret4"

moab.mongo.username="insight_user"
moab.mongo.password="secret4"
moab.mongo.host="<moab mongo host>"
```

**i** "secret4" is the password you specified when installing the mongoDB. See [Install MongoDB on page 131](#).

6. On the Insight Server Host, configure the `moab.host` and `messageQueue.secretKey` parameters in the Insight configuration file `/opt/insight/etc/config.groovy`.

```
moab.host = "<moab server>"
messageQueue.secretKey = "<secret key>"
```

The `<secret key>` must match the secret key configured in `moab-private.cfg` on the Moab server for the **MESSAGEQUEUESECREKEY** configuration parameter.

7. On the Insight Server Host, verify that Insight runs on startup.

- Red Hat 6-based systems.

```
[root]# chkconfig insight on
```

- Red Hat 7-based systems

```
[root]# systemctl enable insight.service
```

8. On the Moab Server Host, configure Moab's connection to Insight.
  - a. In `/opt/moab/etc/moab.cfg`, configure the **INSIGHTENDPOINT** parameter so that Moab can connect to Insight. See [Moab Parameters](#) in the *Moab Workload Manager Administrator Guide* for parameter information.

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

`<hostname>` is the server where Insight is located. `<hostname>` is required, `<port>` is optional.

- b. In `/opt/moab/etc/moab-private.cfg`, configure the **MESSAGEQUEUESECRETKEY** parameter so that Moab can connect to Insight. See [Secure communication using secret keys on page 123](#).

```
MESSAGEQUEUESECRETKEY <secret key>
```

The `<secret key>` is required when updating the Insight configuration file later in this procedure.

- c. Restart Moab in order for the new configuration parameters to take effect.

- Red Hat 6-based systems.

```
service moab restart
```

- Red Hat 7-based systems.

```
systemctl restart moab.service
```

- d. Verify that Moab is properly configured to connect to Insight.

```
mdiag -S | grep Insight
```

You should see something similar to the following:

```
[root]# mdiag -S | grep Insight
ZeroMQ Insight connection is bound on port 5574 (reliability port 5575) on host
* using Insight endpoint <the insight hostname displays here>:5568
encryption is on)
ZeroMQ Insight reliable message delivery is using store file(s) up to 1024 MB in
/opt/moab/spool/insight_store/
```

9. On the Insight Server Host, start Insight.

- Red Hat 6-based systems.

```
[root]# service insight start
```

- Red Hat 7-based systems.

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



The first time you start Insight it will take a minute or two to create the database schema. Although 'service insight start' will quickly return OK, it is not safe to terminate Insight while this initialization is taking place. Rebooting or terminating Insight during this initialization may cause the database to not be initialized correctly.

You will know it is safe to reboot or terminate Insight if you see the following line in /opt/insight/log/insight.log.

```
2014-12-11T18:36:08.059-0700    main    INFO
com.ace.insight.app.Application 0    Started Application in 89.502
seconds (JVM running for 89.882)
```

## Related Topics

[Chapter 3 RPM installation on page 97](#)

## Installing Moab Viewpoint



Only an RPM-based installation on a Red Hat-based system is supported for installing Moab Viewpoint. A SUSE-based system is not supported.

This topic contains instructions on how to install Moab Viewpoint (Viewpoint).

In this topic:

- [Prerequisites on page 140](#)
- [Install Viewpoint on page 143](#)
- [Set Up Viewpoint Access on page 146](#)



Viewpoint requires a connection to Moab Server and MWS installed on the shared host (with a Red Hat-based OS; Viewpoint may also be installed on that shared host or on a different host. For documentation clarity, the instructions refer to the shared Moab Server and MWS host as the Moab Server Host and the host on which you install Viewpoint as the Viewpoint Server Host.

## Requirements

Removing this section as the information is listed here: [Component Requirements on page 10](#)

It is only still here for now to not throw off the numbering during the review process.

## Prerequisites

In this section:

- [Security Enhanced Linux on page 140](#)
- [Open Necessary Ports on page 140](#)
- [Configure the ViewpointQueryHelper Plugin on page 141](#)

## Security Enhanced Linux

For systems where Security Enhanced Linux (SELinux) is enforced, you need to adjust SELinux to allow the web server to make network connections and create and write to the log file.

1. To determine the current mode of SELinux, run `getenforce`.

```
[root]# getenforce
```

2. If the command returns a mode of Disabled or Permissive, or if the `getenforce` command is not found, you can skip the rest of this procedure.
3. If the command returns a mode of Enforcing, you can choose between options of customizing SELinux to allow the web GUI to perform its required functions or disabling SELinux on your system.
  - If you choose to customize SELinux:

**i** SELinux can vary by version and architecture and that these instructions may not work in all possible environments.

```
[root]# yum install policycoreutils-python
[root]# semanage permissive -a httpd_t
```

- If you choose to disable SELinux:

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

SELINUX=disabled

[root]# setenforce 0
```

## Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

In this section:

- [Open the Apache Web Server Port \(8081\)](#)
- [Open the WebDav Server Port \(8443\)](#)



## Open the Apache Web Server Port (8081)

On the Viewpoint Server Host, 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"

-A INPUT -p tcp --dport 8081 -j ACCEPT

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

- Red Hat 7-based systems using firewallld

```
[root]# firewall-cmd --add-port=8081/tcp --permanent
[root]# firewall-cmd --reload
```

## Open the WebDav Server Port (8443)

Viewpoint requires you install the moab-viewpoint-filemanager package (WebDav service) on the Moab Server Host. If you will install the Viewpoint Server on a separate host from where you installed the Moab Server, then do the following on the Moab Server Host:

- 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"

-A INPUT -p tcp --dport 8443 -j ACCEPT

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

- Red Hat 7-based systems using firewallld

```
[root]# firewall-cmd --add-port=8443/tcp --permanent
[root]# firewall-cmd --reload
```

## Configure the ViewpointQueryHelper Plugin

You will need to configure the MWS ViewpointQueryHelper plugin to allow Viewpoint to query the Insight MongoDB (MongoDB host, database, port, and user information).

Do the following:

1. Using a web browser, navigate to your MWS instance (<http://<server>:8080/mws/>) and then log in as the MWS administrative

user (moab-admin, by default).

2. Select **Plugins** and then from the drop-down select **Plugins** to display the list of MWS plugins (displays Plugin List page).
3. Click the viewpoint-query-helper plugin to view this plugin's information (displays Show Plugin page).
4. Click **Edit** to modify the Configuration table fields (displays Edit Plugin page). The following is an example of the Edit Plugin page.

**Moab® Web Services** Welcome, moab-admin  
[Documentation](#) [Log Out](#)

[Home](#) [Plugins](#) [Admin](#)

### Edit Plugin

ID: **viewpoint-query-helper**

Plugin Type: **ViewpointQueryHelper** ([Open Documentation](#))

State: **Started**

Precedence:

Auto Start: ☒

Configuration:  [Add Entry](#)

Key	Value
* Host	<input type="text" value="localhost"/>
* Database	<input type="text" value="insight"/>
* Port	<input type="text" value="27017"/>
* User	<input type="text" value="mws_user"/>
* Password	<input type="password" value="*****"/>

[Update](#) [Cancel](#)

5. Modify the values as needed. The following table describes the required information.

Key	Value Description
host	Hostname or IP address of the host on which Insight MongoDB resides.
database	Name of the MongoDB database to which Insight writes.
port	Port number for Insight MongoDB (typically 27017).
user	User name with which MWS connects to Insight MongoDB.
password	Password used by the user listed in the value for the "user" key.

6. When finished, click **Update** to save your changes and close this page; otherwise click **Cancel** to reset all the changes.
7. When satisfied with the values, log out of your MWS instance and close the web browser.

See also [About Plugins](#) in the *Moab Web Services Reference Guide* for more information.

## Install Viewpoint

Do the following:

1. If you are installing Viewpoint on its own host *or* on a host that does not have another RPM installation, complete the steps to prepare the host. See [Preparing the Host for RPM Installations on page 98](#).
2. Set up PostgreSQL for Viewpoint.

**i** These instructions assume you will install the Viewpoint PostgreSQL database on the same host as the Insight PostgreSQL database (strongly recommended). Depending on your system confirmation, this may be the on the Insight Server Host or on the PostgreSQL Database Host.

On the host containing the Insight PostgreSQL, do the following:

```
[root]# su - postgres
[postgres]$ psql
CREATE USER moab_viewpoint WITH PASSWORD 'changeme!';
CREATE DATABASE moab_viewpoint WITH OWNER=moab_viewpoint;
\q
[postgres]$ exit
```

3. On the Moab Server Host, install the moab-viewpoint-filemanager package.
  - a. Install the package.

```
[root]# yum install moab-viewpoint-filemanager
```

- b. Using the instructions in `/opt/acfileman/utils/certs-handling/Readme.txt`, follow these steps:
      - Step 1. Create CA (Certificate Authority).
      - Step 2. Create server (WebDav server) certificate and key.
      - Step 3. Create client certificate and key.
      - Step 4. Configure WebDav server.
      - Step 5. Configure WebDav client (Viewpoint); skip for now.

For example:

```
[root]# cd /opt/acfileman/utils/certs-handling
[root]# ./ac-cert-tool.sh create-ca
[root]# ./ac-cert-tool.sh create-server-cert --altnames 127.0.0.1,localhost
<moab_host>
[root]# ./ac-cert-tool.sh create-client-cert
[root]# bash certs/servers/<moab_host>/install-server-certs.sh -u root:root -p
600 /opt/acfileman/etc/
[root]# vi /opt/acfileman/etc/uwsgi.ini
      [uwsgi]
      https = 0.0.0.0:8443,/opt/acfileman/etc/server-
cert.pem,/opt/acfileman/etc/server-
key.pem,HIGH:!DSS:!3DES:!SRP:!PSK:!DHE:!EDH:!ECDH:!aNULL,!/opt/acfileman/etc/ca-
cert.pem
```

- c. Configure the moab-viewpoint-filemanager package to start up at system boot and start the moab-viewpoint-filemanager.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

4. On the Moab Server Host, enable negative job priority.

- a. Set the ENABLENEGJOBPRIORITY parameter in /opt/moab/etc/moab.cfg.

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

**i** You must set this Moab parameter to support Viewpoint features that enable users to specify user priorities for their jobs. See *Advanced Settings in the Viewpoint Reference Guide* for more information on enabling user priorities for jobs.

- b. Restart Moab.

- Red Hat 6-based systems

```
[root]# service moab restart
```

- Red Hat 7-based systems

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

5. On the Moab Server Host, register Viewpoint as a client in MWS.

- a. Edit the `grails.plugin.springsecurity.oauthProvider.clients` array in `/opt/mws/etc/mws-config.groovy` and specify a client id and a client secret.

**i** The following is a suggested script for generating the client secret:

```
dd if=/dev/urandom count=24 bs=1 2>/dev/null | base64
```

```
[root]# vi /opt/mws/etc/mws-config.groovy
grails.plugin.springsecurity.oauthProvider.clients = [
    [
        clientId: "viewpoint",
        clientSecret: "<ENTER-CLIENTSECRET-HERE>",
        authorizedGrantTypes: ["password"]
    ]
]
```

- b. Restart Tomcat.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

6. On the Viewpoint Server Host, do the following:

- a. Install the `moab-viewpoint` package. This will also install Apache Server and several Python packages.

```
[root]# yum install moab-viewpoint
```

- b. (Optional) Configure virtual hosts. The `moab-viewpoint` package installs an `/etc/httpd/conf.d/viewpoint.conf` file for Apache. Virtual host configurations should be made within this file. See <http://httpd.apache.org/docs/2.2/vhosts/> for more information.

- c. Initialize Viewpoint's PostgreSQL database.

```
[root]# export IRIS_DATABASE_HOST=<PostgreSQL_host> #Host on which you
installed PostgreSQL for Viewpoint
[root]# export IRIS_DATABASE_PORT=5432
[root]# export IRIS_DATABASE_NAME=moab_viewpoint
[root]# export IRIS_DATABASE_USER=moab_viewpoint
[root]# export IRIS_DATABASE_PASSWORD='changeme!'
[root]# /opt/viewpoint/manage.py syncdb
```


- d. Configure Viewpoint to access its PostgreSQL database.

- Uncomment all the `IRIS_DATABASE_*` entries in `/etc/httpd/conf.d/viewpoint.conf`.

- Set the values so that they match the values specified when you initialized the PostgreSQL database (previous step)

```
[root]# vi /etc/httpd/conf.d/viewpoint.conf
SetEnv IRIS_DATABASE_HOST <PostgreSQL_host>
SetEnv IRIS_DATABASE_PORT 5432
SetEnv IRIS_DATABASE_NAME moab_viewpoint
SetEnv IRIS_DATABASE_USER moab_viewpoint
SetEnv IRIS_DATABASE_PASSWORD changeme!
```

- e. Set the Viewpoint Administrator password (viewpoint-admin, by default). In the `/opt/viewpoint/iris/settings.py` file, change the value of "VIEWPOINT\_ADMIN\_PASSWORD" to a strong password.

 The "VIEWPOINT\_ADMIN\_PASSWORD" is "changeme!", by default. If you do not change VIEWPOINT\_ADMIN\_PASSWORD, your Viewpoint instance will not be secure. For tips on choosing a good password, see <https://www.us-cert.gov/ncas/tips/ST04-002>.

- f. Start (or restart) the Apache service.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

## Set Up Viewpoint Access

After you have finished the Viewpoint installation, you will need to perform additional activities to enable Viewpoint to function correctly.

Do the following in order:

- [Configure Viewpoint to Access MWS](#)
- [Configure Viewpoint to Access the WebDav Service \(filemanager\)](#)
- [Configure Viewpoint to Grant Access to Users](#)

## Configure Viewpoint to Access MWS

You will need to set up the initial MWS connection information for Viewpoint.

Do the following:

1. Using a web browser, navigate to your Viewpoint instance. (`http://<viewpoint_host>:8081`; where `<viewpoint_host>` is the IP address or name of the Viewpoint Server Host)

2. Log in as the Viewpoint administrative user (viewpoint-admin, by default) using the password you set in the Viewpoint installation instructions.

**i** The Viewpoint administrative user has very limited rights: it has permission to set up the initial connection with MWS and to change the MWS connection information later if necessary.

The Configuration page displays.  
The following is an example of the Configuration page.

The screenshot shows the 'Basic Configuration' page with a blue sidebar on the left. The main content area is divided into three sections:


- MWS Configuration:** Contains input fields for Server (http://server:8080), Username (moab-admin), Password (masked with dots), Path (/mws/), Client Id (viewpoint), and Client Secret (masked with dots). A blue 'TEST' button is at the bottom right of this section.
- File Manager Configuration:** Contains input fields for Accessible Roots (/home:/tmp) and Maximum Upload Size (bytes) (-1).
- Misc Options:** Contains an input field for Node Names to Ignore (DEFAULT,GLOBAL) and a checkbox for 'Use Google Analytics to help improve this product' which is currently unchecked.

A blue 'SAVE' button is located at the bottom right of the entire configuration area.

3. In the MWS Configuration area, do the following:
  - a. In the Server field, enter the URL for MWS on the Moab Server Host. For example: http://127.0.0.1:8080

**i** If your configuration uses a secure connection between Viewpoint and MWS, the URL must contain "https" and the secure port.

- b. In the Username and Password fields, enter the MWS administrator credentials. You can find these credentials in `/opt/mws/etc/mws-config.groovy` on the Moab Server Host. Look for `auth.defaultUser.username` and `auth.defaultUser.password`.

- c. In the Path field, the default value (/mws/) is already filled in. Leave it as is unless you have installed MWS with a non-default path.
  - d. In the Client Id and Client Secret fields, enter the values that you set during the Viewpoint installation. Refer back to the step ([On the Moab Server Host, register Viewpoint as a client in MWS.](#)) earlier in this topic.
  - e. Click **TEST** to confirm the settings are correct. If successful,  appears outside the Server field.
4. In the File Manager Configuration area, do the following:
    - a. In the Accessible Roots field, enter the directories that you want users to be able to access from within Viewpoint. Separate directories with a colon (:).
    - b. In the Maximum Upload Size field, specify the maximum file size that can be uploaded. A value of '-1' means unlimited.
  5. In the Misc Options area, do the following:
    - a. In the Node Names to Ignore field, enter the nodes that you want Viewpoint to ignore. Separate node names with a comma (,).
    - b. Choose whether you wish to use Google Analytics to help improve this product.
  6. Click **SAVE** to submit the information. You will be logged out of Viewpoint.

## Configure Viewpoint to Access the WebDav Service (filemanager)

The WebDav service is provided by the moab-viewpoint-filemanager package. This section finishes the SSL authentication steps you began when you installed moab-viewpoint-filemanager -- that is, Step 5 of

`/opt/acfileman/utils/certs-handling/Readme.txt` that you skipped earlier.

Do the following:

1. On the Moab Server Host, do the following:

```
[root]# cd /opt/acfileman/utils/certs-handling/certs
[root]# scp ca/ca-cert.pem client/client-cert.pem client/client-key.pem
root@<viewpoint_host>:/opt/viewpoint/webdav_client
```

2. On the Viewpoint Server Host, do the following:

- a. Set the mode, owner, and group of the files you copied over.

```
[root]# cd /opt/viewpoint/webdav_client
[root]# chmod 600 ca-cert.pem client-key.pem client-cert.pem
[root]# chown apache:apache ca-cert.pem client-key.pem client-cert.pem
```

- b. Replace the WEBDAV section in `/opt/viewpoint/config/config.json` with the following excerpt, substituting the CLIENT\_HOST option value



with the Moab Server Hostname.

```
[root]# vi /opt/viewpoint/config/config.json
"WEBDAV": {
  "CLIENT_PROTOCOL": "https",
  "CLIENT_SSL_CERT": [
    "/opt/viewpoint/webdav_client/client-cert.pem",
    "/opt/viewpoint/webdav_client/client-key.pem"
  ],
  "CLIENT_VERIFY_SSL": "/opt/viewpoint/webdav_client/ca-cert.pem",
  "CLIENT_PORT": 8443,
  "CLIENT_ROOT_PATH": "/",
  "CLIENT_HOST": "<moab_server_host>"
},
```

## Configure Viewpoint to Grant Access to Users

For a user to be able to access Viewpoint, he or she must be a member of a principal.

Do the following:

1. Using a web browser, navigate to your Viewpoint instance.  
([http://<viewpoint\\_host>:8081](http://<viewpoint_host>:8081); where *<viewpoint\_host>* is the IP address or name of the Viewpoint Server Host)
2. Log in as the MWS administrative user (moab-admin, by default).
3. Click **Configuration** from the menu. The Basic Configuration page displays.
4. Click **Principals** from the left pane.
5. Create one or more principals. See *Creating or Editing Principals* in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

**i** Viewpoint comes configured with an admin and a user role that you can assign to the principals. You can also modify the default roles and create new roles as needed. See *About Roles* in the *Moab Viewpoint Reference Guide* for more information.

### Related Topics

[Chapter 3 RPM installation on page 97](#)

## Additional Configuration

In this section:

- [Configuring SSL in Tomcat on page 150](#)
- [Setting Up OpenLDAP on CentOS 6 on page 150](#)
- [Trusting Servers in Java on page 157](#)

### Configuring SSL in Tomcat

To configure SSL in Tomcat, please refer to the Apache Tomcat [documentation](http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html) (<http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>).

### 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 150](#)
- [Adding an Organizational Unit \(OU\) on page 154](#)
- [Adding a User on page 155](#)
- [Adding a Group on page 156](#)
- [Adding a User to a Group on page 156](#)

**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}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
```

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}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

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}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```



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



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.

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` earlier in this procedure, 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](http://wiki.centos.org/HowTos/Network/IPTables) (<http://wiki.centos.org/HowTos/Network/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 154](#).

## 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 155](#).

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

In this topic:

[Prerequisites on page 157](#)

[Retrieve the Server's X.509 Public Certificate on page 157](#)

[Add the Server's Certificate to Java's Keystore on page 157](#)

### 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 the Moab HPC Suite RPMs on page 159](#)
- [Upgrading from MongoDB 2.0 to 2.4.x on page 168](#)
- [Migrating the MAM Database from MySQL to PostgreSQL on page 169](#)

## Upgrading the Moab HPC Suite RPMs

This topic provides instructions to upgrade the Moab HPC Suite RPMs to the latest release version. It includes instructions for migrating your database schema to a new version if necessary.



If using Moab Accounting Manager, this upgrade will result in the creation of a new mam user. This new mam user will replace the moab user as the primary MAM Accounting Admin. Moab Accounting Manager files and directories will be changed to be owned by this new user as part of the upgrade process.



If upgrading Moab Web Services from a version prior to 8.0, 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.

## Upgrade the RPM Suite



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.

## 1. Shut down all Adaptive services.

### • Red Hat 6-based systems

```
[root]# service moab stop      # you can also run mschedctl -k
[root]# service mam stop      # If running mam on the same host
[root]# service tomcat6 stop
[root]# service pbs_server stop
[root]# service pbs_mom stop   # if running pbs_mom on the same host
[root]# service trqauthd stop
```

### • Red Hat 7-based systems

```
[root]# systemctl stop moab      # you can also run mschedctl -k
[root]# systemctl stop mam      # If running mam on the same host
[root]# systemctl stop tomcat6
[root]# systemctl stop pbs_server
[root]# systemctl stop pbs_mom   # if running pbs_mom on the same host
[root]# systemctl stop trqauthd
```

## 2. Download the latest 9.0.0 build executable (moab-hpc-enterprise-suite-*<version>*-*<timestamp>*-*<OS>*.tar.gz, for example) from the [Adaptive Computing](#) website.

## 3. Untar the package.

```
[root]# tar xzf moab-hpc-enterprise-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: 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, `[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.


- a. 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
```

- b. After making changes in the `/etc/yum.repos.d` directory, run the following command to update the `yum` cache.

```
[root]# yum clean all
```

## 7. Update the 9.0.0 suite packages.

 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`.

- a. If you are upgrading Torque, then on the Torque Server Host, each Torque MOM Host, and each Torque Client Host (including the Moab Server Host if applicable), do the following:

```
[root]# yum update moab-torque*
```

- b. If you are upgrading Moab Workload Manager, then on the Moab Server Host, do the following:

```
[root]# yum update moab-workload-manager*
```

- c. If you are upgrading Moab Accounting Manager, then on the MAM Server Host, do the following as needed:

**i** The MAM RPM name has changed between version 8.1 and 9.0. The RPM obsolescence process removes the old RPM and installs the new RPM separately; this results in removing the mam user and not preserving the customized configuration files. A special process must be followed when upgrading from an RPM version prior to 9.0.

- If you are upgrading MAM from an RPM version prior to 9.0, do the following:

```
for i in /opt/mam/etc/{goldd,site}.conf
do
cp -p ${i} ${i}.rpmsave
done
rpm -e --no-postun moab-hpc-accounting-manager
yum install moab-accounting-manager
for i in /opt/mam/etc/mam-*.conf
do
cp -p ${i} ${i}.rpmnew
done
\cp -f /opt/mam/etc/gold.conf.rpmsave /opt/mam/etc/mam-client.conf
\cp -f /opt/mam/etc/goldd.conf.rpmsave /opt/mam/etc/mam-server.conf
\cp -f /opt/mam/etc/goldg.conf.rpmsave /opt/mam/etc/mam-gui.conf
\cp -f /opt/mam/etc/site.conf.rpmsave /opt/mam/etc/mam-site.conf
```

- If you are upgrading MAM from an RPM version at or after 9.0, do the following:

```
[root]# yum update moab-accounting-manager*
```

- d. If you are upgrading Moab Web Services, then on the MWS Server Host, do the following:

```
[root]# yum update moab-web-services*
```

8. If you use ODBC, you must upgrade to the 9.0.0 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 168](#) 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) (<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
```



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.



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 the Linux x64 RPM version of Oracle® Java® 8 Runtime Environment.



Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 is also supported. All other versions of Java, including OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run MWS.

Do the following:

- a. Go to the [Oracle Java download page](http://java.com/en/download/linux_manual.jsp) ([http://java.com/en/download/linux\\_manual.jsp](http://java.com/en/download/linux_manual.jsp)).
- b. Copy the URL for the Linux x64 RPM version and run the following command:

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

12. Merge the configuration files.

**i** You will need to decide whether to start with the old configuration file and add newer configuration options (or vice versa). Typically it 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 9.0.0 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/moab/etc/moab.cfg /opt/moab/etc/moab.cfg.bak
```

- If you are upgrading from a version prior to 8.0, 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\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. If you are using Moab Accounting Manager, do the following:

- a. If the upgrade resulted in the creation of any `*.conf.rpmnew` files in `/opt/mam/etc` (e.g. `/opt/mam/etc/mam-server.conf.rpmnew`), compare your existing configuration files with those distributed with the new release to determine if you want to merge any of the new options into your configuration files.

- b. Add the new mam user as a MAM Accounting Admin.

```
[root]# su -c "mam-create-user -u mam -d \"Accounting Admin\"" moab
[root]# su -c "mam-modify-role -r SystemAdmin --add-user mam" moab
[root]# perl -p -i -e 's/moab/mam/ if /^super.user/' /opt/mam/etc/mam-server.conf
```

- c. Migrate the Moab Accounting Manager database from your current version to 9.0, running the migration script. The migration scripts are located in the `/usr/share/moab-accounting-manager/9.0.0/` directory.

**i** The migration script must be run as the user that is running the mam service.

- i. Start the mam service and determine the user running the mam-server process. User "mam" is running mam-server in the following example.

- Red Hat 6-based system

```
[root]# service mam start
[root]# ps -ef | grep mam-server
mam      25274      1  0 Jan12 ?           00:00:01 /usr/bin/perl -w
/opt/mam/sbin/mam-server
```

- Red Hat 7-based system

```
[root]# systemctl start mam
[root]# ps -ef | grep mam-server
mam      25274      1  0 Jan12 ?           00:00:01 /usr/bin/perl -w
/opt/mam/sbin/mam-server
```

- ii. Run the migration script as the user running the mam service.

```
[root]# su -c /usr/share/moab-accounting-manager/9.0.0/migrate_8.0-8.1.pl
mam
```

## 14. Start all Adaptive services.

**i** The MAM service was started earlier in these instructions.

- Red Hat 6-based systems

```
[root]# service pbs_server start
[root]# service moab start
[root]# service tomcat6 start
[root]# service pbs_mom start      # if running pbs_mom on the same host
[root]# service trqauthd start
```

- Red Hat 7-based systems

```
[root]# systemctl daemon-reload
[root]# systemctl start moab
[root]# systemctl start tomcat6
[root]# systemctl start pbs_server
[root]# systemctl start pbs_mom      # if running pbs_mom on the same host
[root]# systemctl start trqauthd
```

## 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 rails.mongo /opt/mws/etc/mws-config.groovy
rails.mongo.username = "mws_user"
rails.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](https://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.

## Migrating the MAM Database from MySQL to PostgreSQL

PostgreSQL is 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.
  - **Manual Install** - [Installing Moab Web Services on page 49](#)
  - **RPM Install** - [Installing Moab Web Services on page 118](#)
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/mam-
server.conf.dist file
[mam]$ vi /opt/mam/etc/mam-server.conf # Merge in the database.datasource from
etc/mam-server.conf.dist
```

## 11. Restart Moab Accounting Manager.

```
[mam]$ mam-server -r
```





## Chapter 4 Troubleshooting

This chapter details some common problems and general solutions. Additional troubleshooting may be found in the individual Moab HPC Suite component documentation.

In this chapter:

- [General Issues on page 173](#)
- [Moab Web Services Issues on page 176](#)

### General Issues

This topic details some common problems and general solutions.

In this topic:

- [Moaberror: "cannot determine local hostname" on page 173](#)
- [Moaberror: "Moab will now exit due to license file not found" on page 174](#)
- [Other Moab issues on page 174](#)
- [Where do I change my passwords? on page 174](#)

#### Moaberror: "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.localhostdomain4
...
```

## Moaberror: "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

See Troubleshooting and System Maintenance in the *Moab Workload Manager Administrator Guide*.

## Where do I change my passwords?

In this section:

- [Moab Super User Username and Password on page 174](#)
- [MongoDB Passwords on page 175](#)

## Moab Super User Username and Password

The default username and password for Moab are **moab-admin** and **changeme!** (respectively).

To change the username and/or the password for the Moab 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**). See the [MongoDB documentation](http://docs.mongodb.org/manual/tutorial/change-user-password/) (<http://docs.mongodb.org/manual/tutorial/change-user-password/>) for detailed instructions.

### 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

This topic details some common problems and general solutions for Moab Web Services.

If something goes wrong with MWS, 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`.

In this topic:

- [MongoDB: Errors during MWS startup on page 176](#)
- [MongoDB: Out of semaphores to get db connection on page 178](#)
- [MongoDB: Connection wait timeout after 120000 ms on page 178](#)
- [java.lang.OutOfMemoryError: Java heap space on page 178](#)
- [java.lang.OutOfMemoryError: PermGen space on page 179](#)
- [SEVERE: Context \[/mws\] startup failed due to previous errors on page 179](#)
- [MoabReached Maximum Number of Concurrent Client Connections on page 179](#)

### 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    rails.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 host) **Is MongoDB installed?**

Run the following commands to assess whether MongoDB is installed on the current host.

```
$ mongo
-bash: mongo: command not found
```

To remedy, install MongoDB, start the `mongod` service and then restart the `tomcat6` service. See [Installing Moab Workload Manager on page 29](#) (Manual Installation) or [Install MongoDB on page 120](#) (RPM Installation) for more information on how to install and configure MongoDB.

- (Only relevant if MongoDB is installed on a separate host) **Is MWS configured to connect to the remote MongoDB host?**

Run the following commands to assess whether MongoDB is installed on the current host.

```
[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`. For example:

```
grails.mongo.options.connectionsPerHost = 60
grails.mongo.options.threadsAllowedToBlockForConnectionMultiplier = 10
```

For more information on these options, refer to these documents:

- [Configuring Moab Web Services](#) in the *Moab Web Services Administrator Guide*, which briefly discusses a few MongoDB driver options.
- The [MongoOptions](#) documentation (<http://api.mongodb.org/java/current/com/mongodb/MongoOptions.html>), which 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](#) above.

## 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.

**i** Beginning with Java 8, the `MaxPermSize` option is ignored.

## java.lang.OutOfMemoryError: PermGen space

(Recommended) Upgrade to Java. Java 8 has completely removed PermGen space and the `MaxPermSize` option is ignored.

For Java version prior to 8, you can 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
```

## MoabReached 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.








## 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 6.0.0 Administrator Guide: [HTML](#)  – [PDF](#) 

### Moab Workload Manager

- Moab Workload Manager 9.0.0 Administrator Guide: [HTML](#) 

### Moab Accounting Manager

- Moab Accounting Manager 9.0.0 Administrator Guide: [HTML](#) 

### Moab Web Services

- Moab Web Services 9.0.0 Reference Guide: [HTML](#) 