



# Moab Access Portal

version 5.3

## Administrator's Guide

---

November 11, 2009

# Copyright

© 2009 Adaptive Computing Enterprises, Inc. All rights reserved.

# Trademarks

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

# Moab Access Portal<sup>®</sup> Administrator's Guide

Copyright © 2004-2009 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.

## Overview

The Moab Access Portal (MAP) is an end-user job submission portal that integrates with the Moab Workload Manager. It provides large-scale submission to the Moab Workload Manager, and associated resource managers, from any location where a web browser is available.

---

## Table of Contents

### 1.0 Installation and Initial Configuration

- [1.1 Installation Prerequisites](#)
- [1.2 Installing Moab Access Portal](#)
- [1.3 Initial Configuration](#)
- [1.4 Initial Testing](#)
- [1.5 Upgrading Access Portal](#)
- [1.6 Tomcat - Apache HTTP Integration](#)
  - [1.6.2 Enabling a SSL Connection](#)

### 2.0 Connectivity Configuration

- [2.1 SSH Public Keys](#)
- [2.2 Multiple Cluster Configuration](#)

### 3.0 Interface Customization

- [3.1 Stylesheets, Logos, etc.](#)

[Appendix A: Configuration Parameter Overview](#)

[Appendix B: Differences Guide](#)

[Appendix C: High Availability with Tomcat](#)

[Appendix D: Developer Resources](#)

[Appendix E: Troubleshooting](#)

# 1.1 Installation Prerequisites

**For an installation on a Linux variant operating system, the Moab Access Portal has no outside dependencies or requirements.**

If installing on an operating system other than Linux, however, Moab Access Portal requires the following:

- A Java 1.5 compatible JVM (Java Virtual Machine) installed and properly configured. You can check the version of your JVM by running "java --version" at a command line.

(Popular and freely available JVMs include [Sun's JDK/SDK](#) and [IBM's JDK](#).)

- A working JSP/servlet container that is compatible with at least the Servlet 2.2 and JSP 1.1 specifications and uses a Java 1.5 compliant SDK.

**NOTE:** Recommended containers include [Tomcat](#), Jetty, WebSphere 5.1, BEA WebLogic, etc. Most of these containers can be integrated or run parallel with existing web server software (such as Apache).

- A running version of the Moab Workload Manager<sup>®</sup> (5.2.0 or higher) running on the cluster head node(s).
- A Moab Workload Manager configuration that allows all non-privileged users to have ADMIN4 access rights. To do this place the line `ADMINCFG[4] USERS=ALL` in the `moab.cfg` file and restart Moab Workload Manager.
- A cluster head node that has SSH properly installed and configured along with users setup to use Moab Workload Manager (the users **must** have a correct environment and permissions to ensure they can access and execute all the Workload Manager client commands: `showq`, `mjobctl`, `mrsvctl`, etc.).
- Modern browsers (cookie support, stylesheets, and JavaScript) be used by end-users that wish to access the portal.

## 1.2 Installing Moab Access Portal

1. **NOTE: The following instructions pertain to installing the Moab Access Portal (MAP) on a Linux operating system. If you wish to install MAP on a different operating system, please contact Cluster Resources for more detailed instructions on how to proceed.**
2. If needed, go to the directory where you downloaded the MAP tarball.
3. Extract the contents of the compressed tarball at the location where you wish MAP to reside. The created directory will contain all of MAP's configuration and executable files.

**NOTE:** If necessary, please use an account that has proper file-system privileges before continuing with the installation.

Example:

```
[root@localhost opt]# tar -zxf map-5.3.0.tar.gz
[root@localhost opt]#
```

4. Change to the newly created directory.

Example:

```
[root@localhost opt]# cd map-5.3.0
[root@localhost map-5.3.0]#
```

5. Run the `./configure` script to begin the automated installation. A series of interactive steps and questions follows:
  - o **Cluster Headnode/Interactive Node Hostname:** Type the **hostname** or **IP address** of the cluster head node and hit **Enter**.
6. Execute the `./mapctl.sh` script to start/restart the servlet container and MAP.

Example:

```
[root@localhost map-5.3.0]# ./mapctl.sh start

Starting Moab Access Portal servlet engine...
Using CATALINA_BASE:   /tmp/map-5.3.0/tomcat
Using CATALINA_HOME:   /tmp/map-5.3.0/tomcat
Using CATALINA_TMPDIR: /tmp/map-5.3.0/tomcat/temp
Using JRE_HOME:        ./java

[root@localhost map-5.3.0]]#
```

7. Navigate to the MAP login page by visiting **`http://<portal-host>:8080/map/`** in a Web browser.

**NOTE:** Replace **`<portal-host:port>`** with the hostname or IP address of the machine where MAP was started (e.g., **`http://webserver.icluster.org:8080/map/`**). Note that the port 8080 is the default servlet engine port, but MAP can be configured to listen on any port.

8. Type in the username and password of a Moab ADMIN1 user such as "root" and click **Login**. If login is successful then Moab Access Portal has been correctly installed.

## 1.3 Initial Configuration

The automated setup script will correctly configure Moab Access Portal (MAP) for most sites; however, additional configuration options may be considered after installation:

- **Ensure Non-Privileged User Access:** To ensure that non-privileged users (those who are not given a specific ADMIN role level in Workload Manager's configuration) have proper access to all Access Portal functionality it may be necessary to enable default ADMIN4 as the default role. To do this place "ADMINCFG[4] USERS=ALL" in your moab.cfg file and restart Moab Workload Manager.
- **Changing Cluster Head Node Settings:** If your cluster head node's location or address changes you may need to change MAP's settings accordingly. This can be done by editing the "MAP-hosts" file located in MAP's configuration directory.
- **Configure Moab Access Portal for MULTIPLE Clusters:** If Access Portal will connect to multiple clusters refer to [Multiple Cluster Configuration](#).
- **Tweak Settings in the map.properties File:** Some default settings may need tweaking depending on the needs of your site.
  - UPLOADDIR - specify where uploaded scripts should be temporarily stored
  - SCRIPTDIR - specify where custom scripts should be temporarily stored
  - SSH-KEY-AUTH - specify whether or not you use SSH keys, instead of passwords, to authenticate users.
  - **NOTE: If you plan on using SSH-KEY-AUTH (SSH key authentication) also follow the [SSH Key Configuration](#) steps.**

(For full documentation on configuration and customization see the [configuration parameters](#).)

## 1.4 Initial Testing

One way to test Moab Access Portal (MAP) is to first submit jobs to the cluster, using Moab Workload Manager's console commands or by means of the Moab Cluster Manager (MCM). Once done, the same user can then log into MAP.

With jobs submitted both with and without MAP, a comparison of these jobs can be made to confirm that MAP is properly communicating with the Moab Workload Manager. One can also get more detailed information about each job or even submit jobs and then confirm on the console or in MCM that the jobs match.

## 1.5 Upgrading

Follow these steps to upgrade an existing Moab Access Portal (MAP) installation:

1. Acquire the MAP distribution file (it is packaged in tar.gz format.)
2. Stop MAP using `./mapctl.sh stop`.
3. Extract the contents of the compressed tarball over the existing MAP directory.
4. Run MAP's `./configure` script as a user with proper installation permissions.
5. Follow the same steps as outlined in the [Installation Instructions](#).

**NOTE:** Any of your customized configuration scripts will be preserved and not overwritten by the upgrade. If you wish to force an overwrite, use the command `./configure --force` when upgrading MAP.

6. Start MAP using `./mapctl.sh start`.

## 1.6 Tomcat-Apache HTTP Integration

The Apache Tomcat servlet container is one of the most popular and best supported open-source engines that Moab Access Portal® can be run in, and when combined with the powerful open-source Apache HTTP web server, provide greater efficiency, [security](#), and stability. Other proprietary offerings such as IBM WebSphere, BEA Weblogic, etc. provide even higher levels of support and equal performance gains. Although we recommend none of these solutions above another, this section was written to help those who wish to integrate Tomcat and HTTP in order to host the Access Portal.

1) First ensure that a recent JVM (Java Virtual Machine) is present on your system. Tomcat requires that a JVM be present in order to properly run. It is important that the Java version installed also includes tools to compile Java code, as this is used by Tomcat to compile JSP pages on the fly. The [Java 2 SDK 1.6.x](#) available from Sun Microsystems meets all of these requirements.

2) Next download both [Apache Tomcat](#) and [Apache HTTP](#) in preparation for installation. At time of writing we recommend Tomcat 5.0.x and Apache HTTP 2.0.x. Newer iterations are available, but at this time they are still in development and have not been extensively tested with Access Portal. Binary installation for Tomcat is almost always best, and the same goes with HTTP.

2) Next install the Tomcat binary files. Simply decompress the downloaded file to the location where you wish Tomcat to reside. (Will be referred to as **\$TOMCAT\_HOME** for the duration of this document.) The default file permissions should suffice for this stage in installation. (A comprehensive installation guide of Tomcat is beyond this document's scope. Please refer to Tomcat's own documentation for additional help and instruction.)

3) Prepare to start Tomcat by ensuring the environment variable **JAVA\_HOME** is set and points to the location of the installed JVM. (Running **which javac** may reveal the root directory of the JVM.) Now execute the **\$TOMCAT\_HOME/bin/startup.sh** file to start the Tomcat service. Open a browser and attempt to access Tomcat's built-in stand-alone web server at **http://hostname:8080/**. If you are welcomed by Tomcat's start web page your installation has been thus far successful.

4) Proceed by installing Access Portal. Detailed instructions are given in [Installing Access Portal](#).

5) Test Access Portal's installation by visiting, in a browser, **http://hostname:8080/map/**. A login screen for the Access Portal should now be visible. If not, check to ensure firewall settings are not restricting packet flow on port 8080.

6) Next install Apache HTTP web server. (A comprehensive installation guide of Apache HTTP is beyond this document's scope. Please refer to Apache HTTP's own documentation for additional help and instruction.)

7) Test the Apache installation by browsing to **http://hostname/** (defaults to port 80). You should see a page confirming you successfully installed Apache. If not, check firewall settings and ensure Apache was correctly installed. A vast community of support exists for the Apache HTTP server and may be able to further assist in exceptional installation issues.

8) Prepare Apache to communicate with Tomcat via the [Apache Tomcat Connector](#). The recommended connector version is JK-1.2.x, as it is the most stable and best supported. Use an already compiled binary that matches your OS and version of Apache HTTP most closely. Simply copy the **mod\_js.so** file found in the compressed installation file into the **\$HTTP\_CONF/modules/** directory, where **\$HTTP\_CONF** is usually **/etc/httpd/**. Ensure it has file permissions matching those of other module files in the same directory.

9) Next copy files found in the Access Portal **tools** directory to the **\$HTTP\_CONF/conf/** directory. The required files include **tomcat.conf** and **workers.properties**.

10) Next edit **tomcat.conf** in its new location and update each line following a **CUSTOMIZE** comment to

reflect the configuration on your system. Save the changes and exit the file.

11) Make a backup of **\$HTTP\_CONF/conf/httpd.conf** and then open it for editing. Append the line **Include \$HTTP\_CONF/conf/tomcat.conf** to the end of the file (where **\$HTTP\_CONF** is the actual directory location.)

12) Restart Apache HTTP. (On most systems, as root, run **/etc/init.d/httpd restart**.)

13) Verify that Apache is now communicating with Tomcat to service JSP/Servlets and Access Portal by visiting the URL **http://hostname/map/** (note that this is on port 80!) You should be directed to MAP's login screen as you were earlier when testing Tomcat's installation. Be aware it may take several seconds for Apache to create an initial connection with Tomcat.

14) Secure Tomcat by restricting access to port 8080 either with firewall settings, or by disabling the stand-alone web server built into Tomcat. You do this by commenting out the XML tag '**<Connector port="8080" ...**' in **\$TOMCAT\_HOME/conf/server.xml**

15) Restart Tomcat to load in configuration changes.

At this point Apache HTTP will serve all static content to users of Access Portal (html, graphics, etc.) while Tomcat will only service dynamic pages created by JSP/Servlet technologies.

## 1.6.2 Enabling a SSL Connection

It is often desirable to also add an additional layer of security between the end-user's browser and Access Portal by encrypting all data in a SSL connection. By default, Apache 2.x comes built with SSL support, and may already have an available secure connection. Check this by running '**netstat -tpan | grep 443**'. If no output is given, you will need to perform additional configuration. (A comprehensive installation guide of SSL is beyond this document's scope. Please refer to OpenSSL and mod\_ssl documentation for more complete instruction.) The next few steps give an overview of this process:

1) Ensure [OpenSSL](#) is installed on the web server machine. (Running **which openssl** as superuser should return a path.)

2) Ensure the Apache module [mod\\_ssl](#) is installed on the web server machine. (Look in **\$HTTP\_CONF/modules**.)

3) If there is already a **ssl.conf** file in the **\$HTTP\_CONF/conf** directory, simply include this into the **httpd.conf** file by appending the line **Include \$HTTP\_CONF/conf/ssl.conf** to the end of the file. If there is no **ssl.conf** file available, add the following to your **httpd.conf** file:

```
LoadModule ssl_module modules/mod_ssl.so
Listen 443
<VirtualHost _default_:443>
    ErrorLog logs/ssl_error_log
    TransferLog logs/ssl_access_log
    LogLevel warn
    SSLEngine on
    SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt
    SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.keyA
</VirtualHost>
```

4) Verify that the files **\$HTTP\_CONF/ssl.crt/server.crt** and **\$HTTP\_CONF/ssl.key/server.key** exist. If they do not exist, then a SSL key needs to be created and signed by a CA authority and copied into these locations. If you wish to create a self-signed certificate then you then execute the following commands:

```
openssl genrsa -out hostname.domain.key 1024
openssl req -new -key hostname.domain.key -out hostname.domain.csr
openssl x509 -days 730 -req -in hostname.domain.csr -signkey hostname.domain.key -out
hostname.domain.cert
```

```
cp hostname.domain.key $HTTP_CONF/ssl.key/server.key
cp hostname.domain.cert $HTTP_CONF/ssl.crt/server.crt
rm hostname.domain.*
```

- 5) Restart Apache HTTP server and verify that a SSL connection is now open on the server machine. (Be wary of firewall settings that may prohibit the binding of port 443!) You can use the '**netstat -tpan | grep 443**' to check if HTTP has activated a secure SSL connection.
- 6) Finally test the SSL connection in a browser by visiting **https://hostname/map/**.

## 2.1 SSH Public Key Configuration

Moab Access Portal® supports secure login by means of SSH key authentication, as opposed to password authentication. To enable SSH key authentication, set the **SSH-KEY-AUTH** parameter in `map.properties` to `TRUE`.

After activating this features, there are two options or methods of handling this type of authentication:

1. Run MAP's servlet engine as root so that MAP can access all users' homedirectories and their SSH keys in the `~/ .ssh/` directory. To run use this method you must enable the **SSH-HOMEDIR-ACCESS** parameter to tell MAP where it should look to find home directories. (See the `map.properties` file for comments related to this.) For example, most systems would use:

```
SSH-HOMEDIR-ACCESS=/home
```

as the location of their home directories.

2. Copy users' keys who will be using MAP to the `{ $CFGDIR } /map/ssh/` directory and rename each key as `<USER>_id` where `<USER>` is the username owning the key. When this is done, ensure that only the user running the servlet engine has permissions to read these files.

After these changes have been made, restart the servlet engine running Access Portal to load in the changes.

## 2.2 Multiple Cluster Configuration

Moab Access Portal<sup>®</sup> can be configured to support multiple clusters by allowing users to choose which cluster they would like to use as they login to the system.

To enable this feature:

1. Open the **\$CONFIG-DIR/MAP-hosts** file in any text editor.
2. Add any number of new cluster entries. The entry syntax is:

```
<Descriptive Name>=<Head Node Name>[:<Alternate SSH Port>][@<Moab's Bin Path>]
```

For example, if a new cluster serial job cluster is installed, with a head node at **headnode.domain.net** and accepting SSH connections through port **8888** then the cluster entry would be `serial=headnode.domain.net:8888`.

If the cluster also had to define where the Moab Workload Manager<sup>®</sup> commands are, set the path accordingly: `serial=headnode.domain.net:8888@/usr/local/moab/bin/`. This path will override, at least for this cluster, any globally configured paths defined by the [MOAB-COMMANDS-PATH](#) parameter.

3. Save changes and restart the servlet container.

## 3.1 Stylesheet and Logo Customization

The color scheme, graphics, and optional portal logos of Moab Access Portal<sup>®</sup> can be fully customized to accommodate any customer's needs. Only basic stylesheet skills and an understanding of URLs is required to fit Access Portal to your organization.

### Stylesheet (Color, Layout, etc.)

Access Portal uses a stylesheet to define the general look-and-feel of the portal. This stylesheet can be edited, or copied and then modified. The stylesheet is found at **webapps/map/include/default.css**. An alternate stylesheet is also provided at **webapps/map/include/spring.css** to demonstrate how simple it is to change color themes. If you make your own stylesheet file, you will need to update the **\$CONFIG-DIR/map.properties** file to reflect the changes. This is done by updating the [THEME](#) parameter to reference the newly created stylesheet. After changing the **map.properties** file you will need to restart the servlet container for changes to take effect.

### Company/Portal Title and Logos

Access Portal also allows for administrators to change the default portal title and to add company/portal logos to the interface. The **map.properties** parameters [PORTAL-TITLE](#), [PORTAL-LOGO](#), and [CLIENT-LOGO](#) change the main title of the portal, add a centered logo above the title, and insert a left-justified image to the title, respectively.

### Other Icons

Many other icons found throughout the portal (except for the left-hand navigational icons) can be changed. The icons' graphic files are defined in the **map.properties** file and can be modified by updating corresponding parameters. Refer to [Appendix A](#) for a complete list and description of each icon parameter.

## Appendix A: Configuration Parameters

These parameters are maintained in the `map.properties` file in a simple `<NAME>=<VALUE>` format. By default, Moab Access Portal® looks for this file in the `/etc/map/` directory, but one can specify an alternate location by passing in the Java system property `CONFIG-FILE=<ABSOLUTE PATH>`.

Name	Format	Default Value	Description	Example
<b>ALLOW-COMPLETED-VIEW</b>	<TRUE   FALSE>	FALSE	Specifies whether users have the option of viewing all jobs including jobs which have already completed.	ALLOW-COMPLETED-VIEW=TRUE
<b>ALLOW-SYSTEM-VIEW</b>	<TRUE   FALSE>	FALSE	specifies whether users have the option of viewing all other users' jobs, reservations, etc.	ALLOW-SYSTEM-VIEW=TRUE
<b>ATTRIBUTES-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the attributes icons (job submit)	ATTRIBUTES-ICON=custom/attrselect.gif
<b>BROWSE-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the file browse icon (job submit)	BROWSE-ICON=custom/filebrowse.gif
<b>CALENDAR-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the calednar icon (job submit)	CALENDAR-ICON=custom/calendar.gif
<b>CANCEL-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the cancellation icon (reservation list)	CANCEL-ICON=custom/cancel.gif
<b>CFGDIR</b>	<ABS PATH> (include / at end!)	<code>/etc/mapdir/</code>	an absolute path defining the directory where the portal's configuration file ( <code>map.properties</code> ) is located and where other config files will be stored	CFGDIR=/etc/mapdir/
<b>CLIENT-LOGO</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as a logo in the upper-right corner of the portal's header	CLIENT-LOGO=custom/computer_club.gif
<b>CLOCK-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the clock icon (job submit)	CLOCK-ICON=custom/clock.gif
<b>DEFAULT-REFRESH-INTERVAL</b>	<SECONDS>	120 seconds	specifies the default refresh or poll interval (in seconds) at which Access Portal will contact the cluster interactive node and refresh its information cache.	DEFAULT-REFRESH-INTERVAL=60
			specifies whether Access Portal	

<b>DEMO</b>	<TRUE   FALSE>	[NONE]	should run in demonstration mode or not.	DEMO=TRUE
<b>FAVICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a .ico file to be used as the browser FAVICON/SHORTCUT IMAGE	FAVICON=custom/favicon.ico
<b>FILE-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the file icon	FILE-ICON=custom/file.gif
<b>FOLDER-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the folder icon	FOLDER-ICON=custom/folder.gif
<b>HELP-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the "Help" navigation button. If this parameter is not specified (commented out) the help system will be disabled. At this time this is the default and recommended configuration.	HELP-ICON=custom/help.gif
<b>LOGDIR</b>	<ABS PATH>	/tmp	<b>deprecated</b> - an absolute path defining the directory where the portal's log configuration file (log4j.properties) is located	LOGDIR=/tmp
<b>LOGOUT-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the "Logout" navigation button	LOGOUT-ICON=custom/logout.gif
<b>MAXFILESIZE</b>	<SIZE>	3145728 bytes	the maximum size (in bytes) of a script/executable file that can be uploaded	MAXFILESIZE=2048
<b>MOAB-COMMANDS-PATH</b>	<ABS PATH>	user's PATH env variable on head node	the absolute path (on the cluster's head node) where Moab commands are located (i.e. showq, mjobctl, mdiag, etc.) INCLUDE TRAILING '/' !	MOAB-COMMANDS-PATH=/usr/local/moab/bin/
<b>PORTAL-LOGO</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as a logo for the main header	PORTAL-LOGO=custom/acme-computing.gif
<b>PORTAL-TITLE</b>	<HTML/TEXT>	[NONE]	HTML enriched text that will appear as the title in the main header - underneath the portal logo	PORTAL-TITLE=<b>ACME</b> High Performance Computing
<b>PORTAL-TITLE-SIZE</b>	<NUMBER>	1.5	Specify the font size of the PORTAL-TITLE. Allowable values are 1.0 to 3.0 and are in units of 'em'.	PORTAL-TITLE-SIZE=2.2
<b>REFRESH-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the	REFRESH-ICON=custom/refresh.gif

			"Refresh" navigation button	
<b>SCRIPTDIR</b>	<ABS PATH>	/tmp	an absolute path defining the directory where uploaded scripts/executables will be temporarily stored before being transferred to the cluster for staging	SCRIPTDIR=/tmp/scripts
<b>SHOW-WORKLOAD-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the "Show All/Only My" icon (used in many areas)	SHOW-WORKLOAD-ICON=custom/showall.gif
<b>SSH-KEY-AUTH</b>	<TRUE   FALSE>	FALSE	specifies whether Access Portal uses SSH keys (instead of passwords) for authentication with the cluster's interactive node	SSH-KEY-AUTH=TRUE
<b>SSH-HOMEDIR-ACCESS</b>	<ABS PATH>	[NONE]	if <b>SSH-KEY-AUTH</b> is set to TRUE this specifies the directory where users' home subdirectories are located. For example, on most default Linux systems these subdirectories are found in /home/. MAP will use this directory to find the users' SSH private keys it needs to login a user. It does this by looking for the file: <code>#{SSH-HOMEDIR-ACCESS}/&lt;USER&gt;/.ssh/&lt;USER&gt;_id,</code> where <USER> is the valid username.	SSH-HOMEDIR-ACCESS=/home
<b>THEME</b>	<REL PATH>	default.css	a relative path from the portal's include directory locating a .css stylesheet which is used to define the portal's "look-and-feel." (See the include/default.css file for details on configurable attributes.)	THEME=custom/showall.gif
<b>UPLOADDIR</b>	<ABS PATH>	/tmp	an absolute path defining the directory where uploaded scripts/executables will be temporarily stored before being transferred to the cluster for staging	UPLOADDIR=/tmp/uploads
<b>UPLOAD-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the upload icon (submit job)	UPLOAD-ICON=custom/upload.gif
<b>VIEW-JOB-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the view job icon (view jobs)	VIEW-JOB-ICON=custom/viewjobs.gif
			A text value that will be	

<b>WEB-TITLE</b>	<TEXT>	MAP	prepended to the title tag on all HTML pages and will show in the header of the web browser.	WEB-TITLE=ACME
<b>WRITE-SCRIPT-ICON</b>	<REL PATH>	[NONE]	a relative path from the portal's images directory locating a graphic file to be used as the write script icon (submit job)	WRITE-SCRIPT-ICON=custom/writescript.gif
<b>X11-FREE</b>	<TRUE   FALSE>	TRUE	specifies whether X11 (or any GUI) dependant Java infrastructure will be loaded in order to support GUI features. (No features at this time in Access Portal require this to be set to FALSE.) On servers, especially headless ones, we recommend TRUE in every case.	X11-FREE=TRUE

# Appendix B: Differences Guide

---

## Moab Access Portal 5.2.0

- **New Features**
  - New tab allows users view the standard output/error files of jobs they have submitted (only works with Moab 5.0.0 and above)
  - Popups on Visual Reservations now work in Internet Explorer
  - Job Templates on Job Submit Tab have been renamed to Job Outlines to correspond with the Job Outlines that exist in Graphical Moab Cluster Manager interface
  - Corrected use of Priority when submitting a job and MAP now recognizes ENABLEPOSUSERPRIORITY
  - Improved directory navigation when selecting directories and files

## Moab Access Portal 4.5.4

- **New Features**
  - Modified colors of charts in Reports section for better visibility
  - All pages now contain valid html markup according to HTML 4.01 Strict
  - When selecting files, an error message is now displayed when the user attempts to view a directory to which they do not have proper permissions
  - Many bug fixes (empty file name errors, incorrect priority range, many css errors, loading of job outlines when a command list is specified, incorrect parsing of job duration time)

## Moab Access Portal 4.5.0

- **New Features**
  - improved Moab Java API now requires Java 5 (1.5) or higher
  - significantly improved installation--Linux distribution now comes with Java and servlet engine embedded
  - improved submission screens that are more intuitive, allowing for easier creation of new scripts on-the-fly
  - added "Avg. Load" to available node information
  - improved the reading of large messages (using tool-tips) for jobs and nodes
  - improved XML parsing (now uses SAX engine) to reduce memory footprint and make job list more efficient
  - improved memory consumption for reports functions
  - eradicated a possible security hole
  - expanded logging verbosity to help diagnose future bugs/issues
  - fixed various minor bugs (incorrect sorting for nodes, time-zone issues, etc.)

## Moab Access Portal 2.2.2

- **New Features**
  - minor bug fixes
  - internationalization (i18n) capabilities which allow MAP to support different languages

## Moab Access Portal 1.2.2

- **New Features**

plug-in framework allowing for 3rd-party developers to create custom screens and modules to interact with MAP and Moab Workload Manager<sup>®</sup> API

- improved logging infrastructure
- improved fault-tolerance and error messages when network or other communication problems occur
- improved setup and installation (automated)
- uses latest Workload Manager Java API for smaller memory footprint and more efficient use of network bandwidth
- built-in support for SSH key authentication
- support for Workload Manager's advanced hosting center capabilities (querying of available resources, creation of virtual private clusters (VPC), VPC lists and management tools, etc.)
- criteria used for job submission and resource queries can now be saved as either personal/global templates that can be loaded for future use

## Moab Access Portal 1.2.0

- **Job Submission**
  - added ability to upload or create new scripts through web browser
  - added ability to browse remote filesystems for scripts, executables, input files, etc.
- **Personal Reservations**
  - added ability to create, list, and remove personal reservations according to specific criteria
- **End-User Information**
  - added ability to view cluster nodes in both list and visual modes
  - added estimated start-time matrix which tells end-users how long they must wait before submitting a theoretical job
  - added workload timeline to illustrate the current and future workload expected on the cluster
  - added ability to send e-mail notifications upon job execution, completion, or failure
  - added intelligible error messages, allowing users to better help themselves
  - added detailed user credential information which allows users to view their permissions on the cluster
  - added consumable credit progress bar which alerts user how many credits have been consumed and how many remain

## Appendix C: High Availability with Tomcat

It is possible to setup Moab Access Portal<sup>®</sup> to run in a high availability environment using the Apache Tomcat servlet engine. Tomcat provides a session replication (clustering) and load balancing feature to increase the availability of any web application. Session replication provides fail-over stability, so that if several users are using Access Portal on a server that fails, the users' session information is automatically replicated and sent to a backup server. The result is that the users do not even perceive any failure and are able to continue using the portal without incident.

The Apache Project provides detailed information on how to configure Tomcat to use these high availability features with web applications:

- [Clustering/Session Replication HOW-TO](#)
- [Load Balancer HOW-TO](#)

## Appendix D: Developer Resources

For customers of Cluster Resources, we provide links to a few resources in order to help web developers get better acquainted with the technologies Moab Access Portal is built with and works with. This can assist in customizing or expanding Moab Access Portal's capabilities.

### JSP/Servlet Resources

- [Official JavaServer Pages Technology Page](#)
- [Introduction to Developing JSP Web Applications](#)
- [Several JSP Tutorials](#)

### JSP/Servlet Engine Resources

- [Apache Tomcat](#)
- [Jetty Servlet Server](#)
- [IBM WebSphere](#)
- [BEA Weblogic](#)
- [JBoss Application Server](#)

# Appendix E: Troubleshooting

- [1. End users cannot see nodes or their credentials](#)
- [2. How to edit the notices page](#)
- [3. Where do I install Moab Access Portal](#)
- [4. Using SSL with tomcat](#)
- [5. "java.awt.HeadlessException" error](#)
- [6. ERROR: cannot migrate job to PBS - user 'root' does not exist on local system](#)
- [7. How do I view the XML output Moab Access Portal uses to send commands to Moab Workload Manager?](#)
- [8. How do I increase the Moab Access Portal Loglevel?](#)
- [9. How do I change aspects of pages in Moab Access Portal?](#)
- [10. My Web browser cannot find Moab Access Portal](#)
- [11. When a job is executed from within Access Portal, why are the results placed into the users root directory and not the directory specified within Access Portal?](#)

## 1. Problem:

### End users cannot see nodes or their credentials

When I log into Moab Access Portal as root, I see all the nodes and I can see credentials and other items. But if I log in as an end user, I can see that the user's job is running, but zero nodes are displayed. Also, the user has "no credentials set". This goes for any user I log in with. I do not see all running jobs when I am logged in as an ADMIN1 level account either.

### Solution:

- You'll need to enable two configuration options:

The first issue is mentioned in the [Initial Configuration](#) section and has to do with the non-privileged access. By default, Moab does not allow any normal user to view node information or their credential information. To enable this capability, insert the following into **moab.cfg**:

```
ADMINCFG[4] Name=map SERVICES=mcredctl,mdiag,mjobctl USERS=ALL
```

Restart Moab Workload Manager by typing **mschedctl -R**, and then you should instantly see node and credential information in Access Portal (if there is some available).

To allow root (and other users) to view all jobs, you will need to edit the **map.properties** file and make sure that **ALLOW-SYSTEM-VIEW=TRUE**. You will then need to restart Access Portal. At this time, there is no way to differentiate between users on a more fine-grained level in Access Portal (e.g. root can see all

jobs, but other users cannot). Even though users can see other's jobs, they cannot modify or cancel them.

## 2. Problem:

### How do I edit the notices page in Moab Access Portal?

#### Solution:

1. Go to the servlet engine's **webapps** directory and look for a directory called **map**. If it exists, enter it and you should also see a directory called **docs**. Inside the **docs** directory is a file called **notices.html**. Anything in this file will be included on the notices page and can be edited.

If there is no **map** in your servlet engine's **webapps** directory, shutdown the servlet engine, and then unzip the file **map.war** found in the **webapps** directory. (Either **jar -xf map.war** or **unzip map.war** will work.) A new directory named **map** should now exist. If it does, you can safely remove the **map.war** file and start up your servlet engine. You should now be able to navigate to the **map/docs/notices.html** file.

## 3. Problem:

### Where do I install Moab Access Portal

Do I need to install Moab Access Portal on the same machine as Moab Workload Manager?

#### Solution:

No, it can be installed on the head node or another node such as a Web server.

## 4. Problem:

### Using SSL with tomcat

How can I use SSL via Tomcat and Apache?

#### Solution:

See the [Enabling an SSL Connection](#) for more information.

## 5. Problem:

## "java.awt.HeadlessException" error

After entering my username and password, I receive the error: "**java.awt.HeadlessException**". This happens during authentication.

### Solution:

1. Make sure that in your **map.properties** file the following parameter is set:

The headless exception error points to a incorrectly configured environment variable, **-DCONFIG-FILE**. If you are starting Moab Access Portal with the **map-star.sh** script and have configured Access Portal to run with your servlet engine correctly then this variable is automatically applied.

- Run the Access Portal configuration again to make sure it is configured correctly with your servlet engine.
- Set the DCONFIG environment variable to the location of **map.properties**. An example for Tomcat is below:

```
export CATALINA_OPTS=-DCONFIG-FILE=/path/to/config/dir/map.properties
```

## 6. Problem:

### ERROR: cannot migrate job to PBS - user 'root' does not exist on local system

When I submit a job using **msub** through Moab Workload Manager or submitting through Cluster Manager or Access Portal, I receive the error: "**ERROR: cannot migrate job to PBS - user 'root' does not exist on local system.**"

### Solution:

1. Moab does not allow job submissions from root. Switch to another user and submit the job again.

## 7. Problem:

### How do I view the XML output Moab Access Portal uses to send commands to Moab Workload Manager?

### Solution:

1. You will need to change a parameter in the `log4jproperties.log` file.

**log4jproperties.log** is located in `/etc/` of the Access Portal, **map**, directory.

Open the file and find the lines **log4j.logger.com.sshertools=** and **log4j.logger.com.moab=**.

Change both settings to **debug**.

Go to `/map/etc/log` to view the xml output after a job is submitted through Access Portal.

## 8. Problem:

### How do I increase the Moab Access Portal Loglevel?

#### Solution:

1. The file that controls MAP's logging is located in the MAP etc directory and is named **log4j.properties**. It usually looks something like the below example:

```
log4j.rootLogger=INFO,ROLL
log4j.appender.ROLL=org.apache.log4j.RollingFileAppender
log4j.appender.ROLL.MaxFileSize=1000KB
log4j.appender.ROLL.MaxBackupIndex=1
log4j.appender.ROLL.layout=org.apache.log4j.PatternLayout
log4j.appender.ROLL.layout.ConversionPattern=%d{HH:mm:ss,SSS} [%t] %-
5p
log4j.logger.com.sshools=fatal
log4j.logger.com.moab=error
```

In order to activate "debug" logging, which aids in diagnostics of problems, we recommend changing the last line and adding the following two so the log file now looks like:

```
log4j.rootLogger=INFO,ROLL
log4j.appender.ROLL=org.apache.log4j.RollingFileAppender
log4j.appender.ROLL.MaxFileSize=1000KB
log4j.appender.ROLL.MaxBackupIndex=1
log4j.appender.ROLL.layout=org.apache.log4j.PatternLayout
log4j.appender.ROLL.layout.ConversionPattern=%d{HH:mm:ss,SSS} [%t] %-
5p
log4j.logger.com.sshools=fatal
log4j.logger.com.moab=debug
log4j.logger.com.moab.api=debug
log4j.logger.com.clusterresources.map=debug
```

This information is valuable in troubleshooting problems. If you need to contact Cluster Resources for further assistance, please make these changes and send the error messages to them.

## 9. Problem:

### How do I change aspects of pages in Moab Access Portal?

#### Solution:

1. [See Problem #2.](#)

## 10. Problem:

**My Web browser cannot find Moab Access Portal when I go to `http://[serverhost:port]/map`**

### **Solution:**

1. Make sure your servlet engine is running.
2. Make sure you can access your Servlet engine's test page.
3. Rerun the Moab Access Portal setup and make sure that the correct information is input into each step.

## 11. Problem:

**When a job is executed from within Access Portal, why are the results placed into the user's home directory and not the directory specified within Access Portal?**

### **Solution:**

1. The "Execution Directory" area in Moab Access Portal is sent to the resource manager. Each resource manager handles the execution directory slightly different. TORQUE uses this directory to submit the job. (See <http://www.clusterresources.com/torquedocs20/2.1jobsubmission.shtml>)

A job, when running under TORQUE, will always execute in the user's home directory, but there is a way to get around this. In the PBS script itself, before running any commands, always have the user put in '**cd \$PBS\_O\_WORKDIR**' which will change to the submission directory and provide the desired functionality.