


# Manually Installing the JAMF Software Server v8.5 or Later

Red Hat Enterprise Linux 6, Ubuntu Server LTS, and  
Windows Server 2008 R2

September 2013



 JAMF Software, LLC  
© 2013 JAMF Software, LLC. All rights reserved.

JAMF Software has made all efforts to ensure that this guide is accurate.

JAMF Software  
301 4th Ave S Suite 1075  
Minneapolis, MN 55415-1039  
(612) 605-6625

Apache Tomcat and Tomcat are trademarks of the Apache Software Foundation.

Apple, the Apple logo, and Mac OS X are a trademark of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the United States and other countries.

The Casper Suite, JAMF Software, the JAMF Software logo, and the JAMF Software Server (JSS) are trademarks of JAMF Software, LLC, registered in the United States and other countries.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

All other product and service names mentioned are the trademarks of their respective companies.

# Contents

- Page 4**     **Red Hat Enterprise Linux 6**  
Requirements  
Installation and Configuration
  
- Page 8**     **Ubuntu Server LTS**  
Requirements  
Installation and Configuration
  
- Page 12**    **Windows Server 2008 R2**  
Requirements  
Installation and Configuration

# Red Hat Enterprise Linux 6

The instructions in this section explain how to manually install and configure the JAMF Software Server (JSS) on Red Hat Enterprise Linux® (RHEL) 6.

**Note:** If you do not want to manually install the JSS, you can use the JSS Installer for Linux. To obtain the JSS Installer for Linux along with its installation instructions, log in to JAMF Nation and click **Show alternative downloads** below the Casper Suite DMG on the following page:

<https://jamfnation.jamfsoftware.com/myAssets.html>

## Requirements

To complete the instructions in this section, you need:

- A computer with a fresh installation of Red Hat Enterprise Linux 6 or later
- The JSS web application (ROOT .war)

To obtain the JSS web application, see the introductory email that you received from JAMF Software or contact your JAMF Software Representative.

## Installation and Configuration

Follow the step-by-step instructions in this section to install and configure the JSS on RHEL.

### Step 1: Create a /tmp/JSSSetup Directory

First, create a /tmp/JSSSetup directory in which to store downloads, tools, and scripts for the JSS.

1. Create a /tmp/JSSSetup directory on the RHEL server by executing:

```
mkdir /tmp/JSSSetup
```

2. Copy the ROOT .war file to the JSSSetup directory that you just created.

## Step 2: Create the jamfsoftware Database

Create a MySQL database in which the JSS can store its data and a MySQL user that can access it. Name the database “jamfsoftware” and give the MySQL user the following credentials:

- Username: jamfsoftware
- Password: jamfsw03

**Note:** If you customize the database name, username, or password, the Database Connection Properties pane will appear the first time you try to access the JSS in a web browser. You will need follow the instructions on this pane and enter your custom settings before you can access the JSS.

1. Install the MySQL Server with yum by executing:

```
sudo yum install mysql-server
```

2. Enable MySQL to start at boot by executing:

```
sudo chkconfig --levels 345 mysqld on
```

3. Start MySQL by executing:

```
sudo /etc/init.d/mysqld start
```

4. At the prompt, create a database named “jamfsoftware” by executing:

```
CREATE DATABASE jamfsoftware;
```

5. Grant permissions to a MySQL user named “jamfsoftware” so that it can access the jamfsoftware database:

```
GRANT ALL ON jamfsoftware.* TO jamfsoftware@localhost IDENTIFIED  
BY 'jamfsw03';
```

**Note:** If you choose to enter a username other than “jamfsoftware,” it is recommended that you do not use “root”.

## Step 3: Install and Configure Tomcat

Apache Tomcat is the web application server that runs the JSS. In this section, you will install Tomcat and manually configure it to use a self-signed SSL certificate on port 8443.

1. Install Tomcat from the Red Hat repository by executing:

```
sudo yum install tomcat6
```

2. Enable Tomcat to start at server boot by executing:

```
sudo chkconfig --levels 345 tomcat6 on
```

3. Copy the JSS web application to the Tomcat webapps directory by executing a command similar to the following:

```
sudo cp /tmp/JSSSetup/ROOT.war /var/lib/tomcat6/webapps/
```

4. Generate a keystore to enable SSL for Tomcat by executing a command similar to the following:

```
sudo keytool -genkey -alias tomcat -keyalg RSA -keypass  
"changeit" -storepass "changeit" -dname "CN=jss.example.com,  
OU=JAMFSW, O=JAMF Software, L=Minneapolis, ST=MN, C=US"  
-keystore /etc/tomcat6/keystore -validity <numdays>
```

Make sure to enter the following attributes as appropriate to your site:

Attribute	Value	Example
CN=	Fully qualified domain name of the server	jss.company.com
OU=	Organizational unit	JAMFSW
O=	Organization	JAMF Software
L=	Location (city or office)	Minneapolis
ST=	State, province, or county	MN
C=	Country or region	US

5. Back up the Tomcat server.xml configuration file by executing:

```
sudo cp /etc/tomcat6/server.xml /etc/tomcat6/server.xml.bak
```

6. Open the server.xml configuration file in a text editor.
7. Locate the "SSL HTTP/1.1 Connector" entry.
8. Uncomment the Connector declaration and modify it to look something like this:

```
...  
<Connector port="8443" SSLEnabled="true"  
maxHttpHeaderSize="8192"  
maxPostSize="0"  
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"  
enableLookups="false" disableUploadTimeout="true"  
acceptCount="100" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS"  
keystoreFile="/etc/tomcat6/keystore"  
keystorePass="changeit" />  
...
```

9. Edit the firewall configuration to allow access to port 8443 by executing:

```
sudo system-config-firewall -tui
```

10. Choose **Other** or **Customize**, and then manually add port 8443 with TCP protocol. The option you choose depends on whether you have a GUI or shell-only interface.

## Step 4: Allocate Additional Memory to Tomcat

To accommodate a large number of computers in the JSS, you will have to allocate additional Java Virtual Machine (JVM) memory to Tomcat. Tomcat's default JVM maximum memory is set to 256Mb.

If there are other services running on your server, make sure to leave enough memory to accommodate them.

1. Edit the `/etc/tomcat6/tomcat6.conf` file.
2. Locate the `JAVA_OPTS` environment variable that defines memory and uncomment it if it is commented. This variable looks something like the following:

```
JAVA_OPTS=" -XmxSizeM -XX:MaxPermSize=256M"
```

3. Modify the `-Xmx` parameter to allocate additional memory to Tomcat. For example, if you need to allocate 1 GB of RAM, modify the variable to look something like this:

```
JAVA_OPTS=" -Xmx1024M -XX:MaxPermSize=256M"
```

4. Start Tomcat by executing:

```
sudo /etc/init.d/tomcat6 start
```

5. Verify that the JSS is running by opening a web browser and accessing the JSS over port 8443. The JSS URL should be something like this:

```
https://jss.mycompany.com:8443/
```

# Ubuntu Server LTS

The instructions in this section explain how to install and configure the JAMF Software Server (JSS) on Ubuntu 10.04 LTS or 12.04 LTS.

**Note:** If you do not want to manually install the JSS, you can use the JSS Installer for Linux. To obtain the JSS Installer for Linux along with its installation instructions, log in to JAMF Nation and click **Show alternative downloads** below the Casper Suite DMG on the following page:  
<https://jamfnation.jamfsoftware.com/myAssets.html>

## Requirements

To complete the instructions in this section, you need:

- A computer running a clean install of Ubuntu 10.04 LTS (64-bit) or 12.04 LTS Server (64-bit ) or later
- The JSS web application (ROOT . war)  
To obtain the JSS web application, see the introductory email that you received from JAMF Software or contact your JAMF Software Representative.

## Installation and Configuration

Follow the step-by-step instructions in this section to install and configure the JSS on Ubuntu 10.04 LTS or 12.04 LTS.

### Step 1: Create a /tmp/JSSSetup Directory

First, create a /tmp/JSSSetup directory in which to store downloads, tools, and scripts for the JSS.

1. Create a /tmp/JSSSetup directory on the Ubuntu server by executing:

```
mkdir /tmp/JSSSetup
```

2. Copy the ROOT . war file to the JSSSetup directory that you just created.



## Step 2: Create the jamfsoftware Database

Create a MySQL database in which the JSS can store its data and a MySQL user that can access it. Name the database “jamfsoftware” and give the MySQL user the following credentials:

- Username: jamfsoftware
- Password: jamfsw03

**Note:** If you customize the database name, username, or password, the Database Connection Properties pane will appear the first time you try to access the JSS in a web browser. You will need to follow the instructions on this pane and enter your custom settings before you can access the JSS.

1. Install the MySQL Server from the Ubuntu repository by executing:

```
sudo apt-get install mysql-server
```

2. (Optional) Create a root password for MySQL when prompted, and then press the Return key.
3. Access the MySQL command-line utility by executing:

```
mysql -u root -p
```

4. At the prompt, create a database named “jamfsoftware” by executing:

```
CREATE DATABASE jamfsoftware;
```

5. Grant permissions to a MySQL user named “jamfsoftware” so that it can access the jamfsoftware database:

```
GRANT ALL ON jamfsoftware.* TO jamfsoftware@localhost IDENTIFIED BY 'jamfsw03';
```

6. Exit the MySQL command-line utility by typing “exit”.

**Note:** If you choose to enter a username other than “jamfsoftware,” it is recommended that you do not use “root”.

## Step 3: Install and Configure Tomcat

Apache Tomcat is the web application server that runs the JSS. In this section, you will install Tomcat and manually configure it to use a self-signed SSL certificate on port 8443.

**Note:** If you are using Ubuntu 10.04 LTS (64-bit), you need to install Tomcat 6. Modify the commands in the following procedure appropriately.

1. Install Tomcat from the Ubuntu repository by executing a command similar to the following:

```
sudo apt-get install tomcat7
```

2. Rename and move the existing ROOT web application provided in the default installation:

```
sudo mv /var/lib/tomcat7/webapps/ROOT /var/lib/tomcat7/webapps/TOMCAT
```

3. Copy the JSS web application to the Tomcat webapps directory by executing a command similar to the following:

```
sudo cp /tmp/JSSSetup/ROOT.war /var/lib/tomcat7/webapps/
```

4. Generate a keystore to enable SSL for Tomcat by executing a command similar to the following:

```
sudo keytool -genkey -alias tomcat -keyalg RSA -keypass "changeit" -storepass "changeit" -dname "CN=jss.example.com, OU=JAMFSW, O=JAMF Software, L=Minneapolis, ST=MN, C=US" -keystore /etc/tomcat7/keystore -validity <numdays>
```

Make sure to change the following attributes as appropriate to your site:

Attribute	Value	Example
CN=	Fully qualified domain name of the server	jss.company.com
OU=	Organizational unit	JAMFSW
O=	Organization	JAMF Software
L=	Location (city or office)	Minneapolis
ST=	State, province, or county	MN
C=	Country or region	US

5. Back up the Tomcat server.xml configuration file by executing a command similar to the following:

```
sudo cp /etc/tomcat7/server.xml /etc/tomcat7/server.xml.bak
```

6. Open the server.xml configuration file in a text editor.
7. Locate the "SSL HTTP/1.1 Connector" entry.
8. Uncomment the Connector declaration and modify it to look something like this:

```
...  
<Connector port="8443" SSLEnabled="true"  
maxHttpHeaderSize="8192"  
maxPostSize="0"  
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"  
enableLookups="false" disableUploadTimeout="true"  
acceptCount="100" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS"  
keystoreFile="/etc/tomcat7/keystore"  
keystorePass="changeit" />  
...
```

- Restart Tomcat by executing:

```
sudo /etc/init.d/tomcat7 restart
```

## Step 4: Allocate Additional Memory to Tomcat

To accommodate a large number of computers in the JSS, you may have to allocate additional Java Virtual Machine (JVM) memory to Tomcat. Tomcat's default JVM maximum memory is 256Mb. If there are other services running on your server, make sure to leave enough memory to accommodate them.

**Note:** If you are using Tomcat 6, be sure to edit the appropriate file and modify the commands in the following procedure appropriately.

- Edit the `/etc/init.d/tomcat7` file.
- Locate the `JAVA_OPTS` environment variable that defines memory and uncomment it if it is commented. This variable looks something like the following:

```
JAVA_OPTS=" -XmxSizeM -XX:MaxPermSize=256M"
```

- Modify the `-Xmx` parameter to allocate additional memory to Tomcat. For example, if you need to allocate 1 GB of RAM, modify the variable to look something like this:

```
JAVA_OPTS=" -Xmx1024M -XX:MaxPermSize=256M"
```

- Restart Tomcat by executing:

```
/etc/init.d/tomcat7 restart
```

- Verify that the JSS is running by opening a web browser and accessing the JSS over port 8443. The JSS URL should be something like this:

```
https://jss.mycompany.com:8443/
```

# Windows Server 2008 R2

The instructions in this section explain how to install and configure the JAMF Software Server (JSS) on Windows Server 2008 R2.

**Note:** If you do not want to manually install the JSS, you can use the JSS Installer for Windows. To obtain the JSS Installer for Windows along with its installation instructions, log in to JAMF Nation and click **Show alternative downloads** below the Casper Suite DMG on the following page:

<https://jamfnation.jamfsoftware.com/myAssets.html>

## Requirements

To complete the instructions in this section, you need:

- A computer running a clean install of Windows Server 2008 R2 Standard with security updates applied
- The JSS web application (ROOT . war)  
To obtain the JSS web application, see the introductory email that you received from JAMF Software or contact your JAMF Software Representative.
- MySQL MSI Installer-Essentials 5.5 or later, available at:  
<http://dev.mysql.com/downloads/mysql/5.5.html#downloads>
- Java SE Development Kit (JDK) 1.6 or 1.7 for Windows x64  
You can download the latest JDK from:  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
- Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 1.6 or 1.7  
You can download the latest JCE from:  
<http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>
- Apache Tomcat 7x Windows Service Installer, available at:  
<http://tomcat.apache.org/download-70.cgi>

# Installation and Configuration

Follow the step-by-step instructions in this section to install and configure the JSS on Windows Server 2008 R2.

## Step 1: Create a \temp\JSSSetup Directory

First, create a \temp\JSSSetup directory in which to store downloads, tools, and scripts for the JSS.

1. Create a temp directory called JSSSetup on Windows server. For example:  
C:\temp\JSSSetup
2. Copy the ROOT .war file to the JSSSetup directory that you just created.

## Step 2: Create the jamfsoftware Database

Create a MySQL database in which the JSS can store its data and a MySQL user that can access it. Name the database "jamfsoftware" and give the MySQL user the following credentials:

- Username: jamfsoftware
- Password: jamfsw03

**Note:** If you customize the database name, username, or password, the Database Connection Properties pane will appear the first time you try to access the JSS in a web browser. You will need follow the instructions on this pane and enter your custom settings before you can access the JSS.

1. Launch the MySQL Installer.
2. Follow the onscreen instructions to perform a Typical Install.  
If you want to change the installation path, perform a Custom Install instead.
3. When the MySQL Wizard appears, use the default settings to configure the installation.
4. When the installation is complete, leave the **Configure the MySQL Server now** checkbox selected and click **Finish**.
5. When the MySQL Server Instance Configuration Wizard appears, click **Next**.
6. Select **Standard Configuration** and click **Next**.
7. Select **Include the Bin Directory in the Windows PATH** and click **Next**.
8. Create a root password and click **Next**.
9. Click **Execute** when prompted.
10. Open the MySQL Command Line Client by choosing **Start > All Programs > MySQL > MySQL Server 5.x > MySQL Command Line Client** from the menu bar.

11. In the MySQL Command Line Client, enter the root password that you created in step 8, and then press the Return key.

12. At the prompt, create a database named "jamfsoftware" by executing:

```
CREATE DATABASE jamfsoftware;
```

13. Grant permissions to a MySQL user named "jamfsoftware" so that it can access the jamfsoftware database:

```
GRANT ALL ON jamfsoftware.* TO jamfsoftware@localhost IDENTIFIED BY 'jamfsw03';
```

**Note:** If you choose to enter a username other than "jamfsoftware," it is recommended that you do not use "root".

14. Type `exit` to close Command Prompt.

### Step 3: Install and Configure Java

Java is required to start Tomcat, the web application server that runs the JSS.

1. Launch the JDK Installer.
2. Follow the onscreen instructions, using the default settings to configure the installation.
3. Download the JCE if you haven't already. Then extract the downloaded file.
4. Copy the following files from the extracted folder to `C:\Program Files\Java\jre6 or 7\lib\security\`:
  - `local_policy.jar`
  - `US_export_policy.jar`

Replace the existing files when prompted.

### Step 4: Install and Configure Tomcat

Apache Tomcat is the web application server that runs the JSS. The following instructions explain how to manually configure Tomcat to use a self-signed SSL certificate on port 8443.

1. Launch the Apache Tomcat 7.x Windows Service Installer.
2. Follow the onscreen instructions and choose **Minimum** when prompted with installation options.
3. Expand the option for **Tomcat** and select the **Service Startup** option to customize the install. Selecting the **Service Startup** option automatically starts Tomcat when the computer starts.
4. Click **Next** and use the default settings to configure the rest of the installation.

- Open Command Prompt.
- Rename and move the existing ROOT web application provided in the default Tomcat installation:

```
move "C:\Program Files\Apache Software Foundation\Tomcat 7.0\
webapps\ROOT" "C:\Program Files\Apache Software Foundation\Tomcat
7.0\webapps\TOMCAT"
```

- Copy the JSS web application to the Tomcat webapps directory by executing a command similar to the following:

```
copy C:\temp\JSSSetup\ROOT.war "C:\Program Files\Apache Software
Foundation\Tomcat 7.0\webapps\"
```

- Generate a keystore to enable SSL for Tomcat by executing a command similar to the following:

```
"C:\Program Files\Java\jre6 or jre7\bin\keytool.exe" -genkey -alias
tomcat -keyalg RSA -keypass "changeit" -storepass "changeit"
-dname "CN=jss.example.com, OU=JAMFSW, O=JAMF Software,
L=Minneapolis, ST=MN, C=US" -keystore "C:\Program Files\Apache
Software Foundation\Tomcat 7.x\keystore" -validity
<numdays>
```

Make sure to change the following attributes as appropriate to your site:

Attribute	Value	Example
CN=	Fully qualified domain name of the server	jss.company.com
OU=	Organizational unit	JAMFSW
O=	Organization	JAMF Software
L=	Location (city or office)	Minneapolis
ST=	State, province, or county	MN
C=	Country or region	US

- Back up the Tomcat server.xml configuration file located at C:\Program Files\Apache Software Foundation\Tomcat 7.x\conf\.
- Open the server.xml configuration file.

**Note:** You can do this with Wordpad, as long as the file is saved as a plain text document named server.xml.

- Locate the "SSL HTTP/1.1 Connector" entry.

12. Uncomment the Connector declaration and modify it to look something like this:

```
...
<Connector port="8443" SSLEnabled="true"
  maxHttpHeaderSize="8192"
  maxPostSize="0"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS"
  keystoreFile="C:\Program Files\Apache Software
  Foundation\Tomcat 7.x\keystore"
  keystorePass="changeit" />
...
```

13. Go back to Command Prompt, and add a rule to the firewall configuration to allow access to port 8443 by executing:

```
netsh advfirewall firewall add rule name=JSS_Secure dir=in
  action=allow protocol=TCP localport=8443
```

## Step 5: Allocate Additional Memory to Tomcat

To accommodate a large number of computers in the JSS, you will have to allocate additional Java Virtual Machine (JVM) memory to Tomcat. Tomcat's default JVM maximum memory is set to 256Mb.

If there are other services running on your server, make sure to leave enough memory to accommodate them.

1. From the menu bar, choose **Start > All Programs > Apache Tomcat 7** and open the Configure Tomcat application.
2. Click the **Java** tab.
3. Enter the amount of memory you want to allocate in the **Maximum Memory Pool** field.
4. Click **Apply**.
5. Open Command Prompt and restart Tomcat by executing:

```
net stop Tomcat7
net start Tomcat7
```

6. Verify that the JSS is running by opening a web browser and accessing the JSS over port 8443. The JSS URL should be something like this:  
<https://jss.mycompany.com:8443/>