Required Software

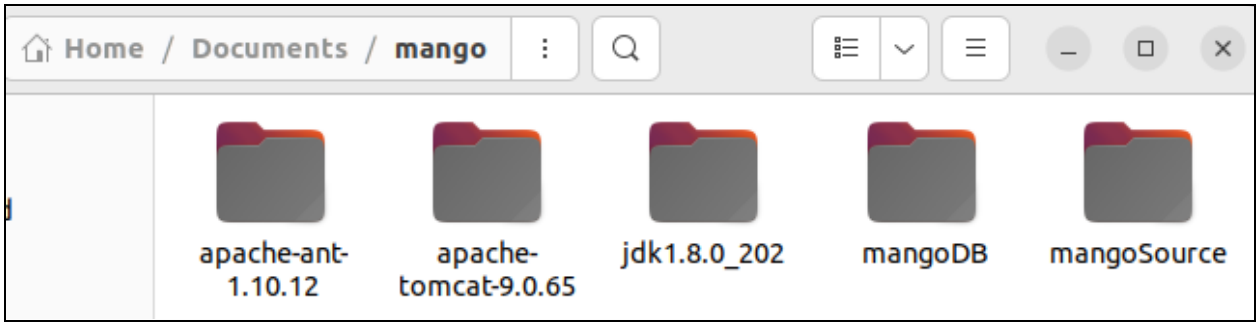
Apache Ant 1.10.x	- Java automated build + deployment tool
Apache tomcat 9.0.x	- Web Server
Java 8 JDK	- Java
mangoSource	- Mango source code
mangoDB	- Mango database

Linux Installation

The following instructions were performed on a clean installation of Ubuntu 22.04. There are no known compatibility issues on other distributions.

File Setup

Download the above required software for your distribution, and extract the directories into a folder. We will extract the files to ~/Documents/mango, but any directory will work.



Environment Variables

Now, we must set some environment variables. The easiest way to do this is by modifying out ~/.bashrc file. Note the absolute paths of ./apache-ant-1.10.12/bin and ./jdk1.8.0_202. In our case, these are /home/morgan/Documents/mango/apache-ant-1.10.12/bin and /home/morgan/Documents/mango/jdk1.8.0_202 respectively.

In our ~/.bashrc file, we must add ant's bin directory to our PATH, and set JAVA_HOME to be our java directory:

```
export PATH="/home/morgan/Documents/mango/apache-ant-1.10.12/bin":$PATH
export JAVA_HOME="/home/morgan/Documents/mango/jdk1.8.0_202"
```

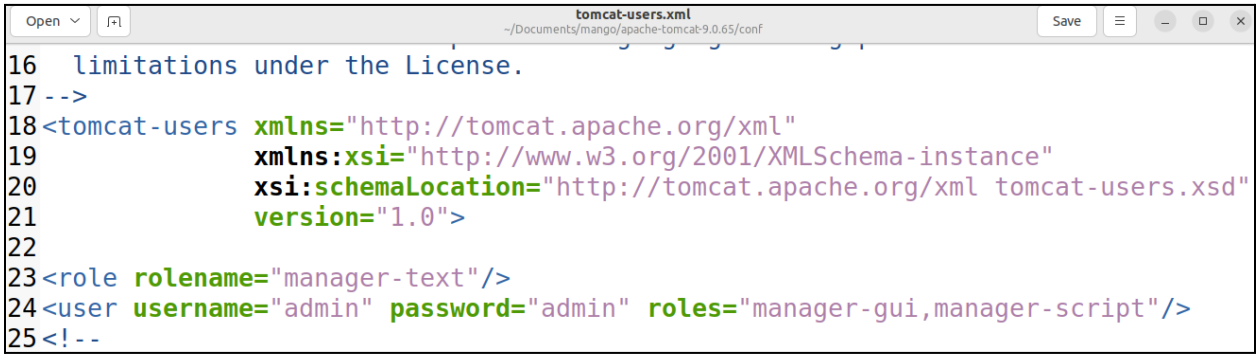
Make sure that **:$PATH** is appended to the PATH variable. Otherwise, your terminals will act strangely.

Tomcat Configuration

Next, we need to configure tomcat. Edit the conf/tomcat-users.xml file in the tomcat directory, and add the following lines:

```
<role rolename="manager-gui"/>
<user username="admin" password="admin"
        roles="manager-gui,manager-script"/>
```

Ex:

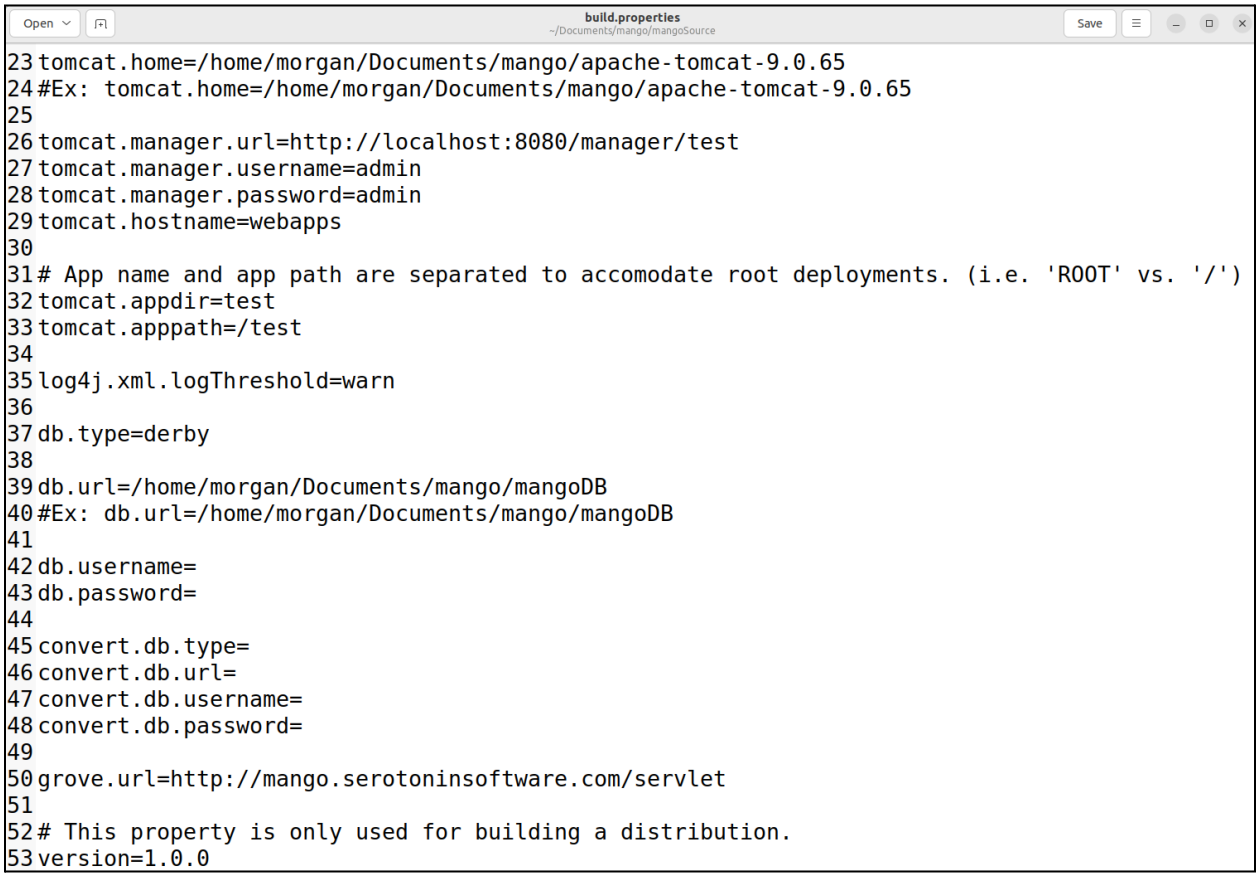


We also need to make the scripts in tomcat's bin directory executable. Navigating here and running **`sudo chmod +x *.sh`** is sufficient.

```
morgan@computer:~/Documents/mango/apache-tomcat-9.0.65/bin$ sudo chmod +x *.sh
morgan@computer:~/Documents/mango/apache-tomcat-9.0.65/bin$ ls
bootstrap.jar                daemon.sh          startup.bat
catalina.bat                 derby.log          startup.sh
catalina.sh                  digest.bat         tomcat-juli.jar
catalina-tasks.xml           digest.sh          tomcat-native.tar.gz
ciphers.bat                  makebase.bat       tool-wrapper.bat
ciphers.sh                   makebase.sh        tool-wrapper.sh
commons-daemon.jar           setclasspath.bat   version.bat
commons-daemon-native.tar.gz setclasspath.sh    version.sh
configtest.bat               shutdown.bat
configtest.sh                shutdown.sh
```

Configuring mango's build.properties

The final configuration step we must perform is modifying mangoSource/build.properties. The **tomcat.home** and **db.url** must be set to the absolute paths of the apache tomcat and mangoDB directories. In our case, our build.properties file is as follows:



Quickstart

To run mango, first navigate to apache tomcat's bin directory and run the startup.sh script. Then, navigate to mangoSource and run the following commands to compile mango and reload tomcat respectively:

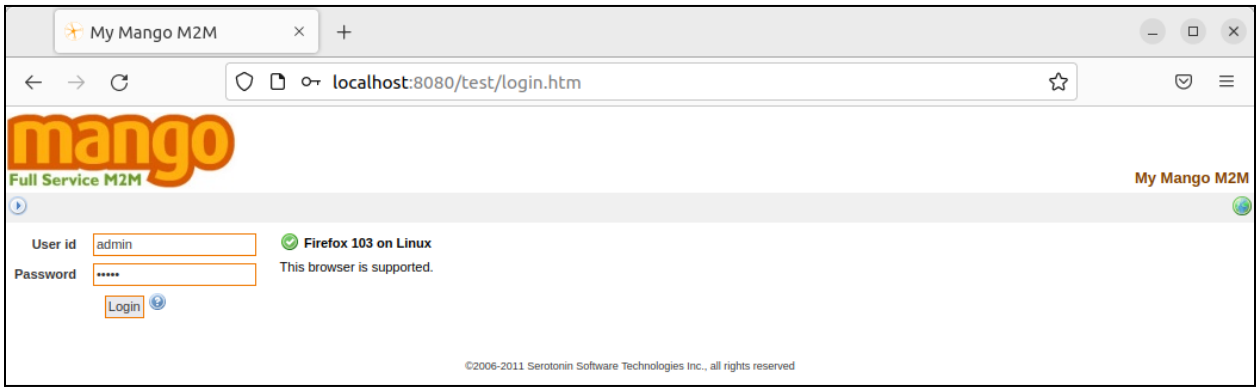
`ant fullDeploy`
`ant reload`

Once done, you should be able to navigate to http://localhost:8080/test in a browser to access mango. Note that once tomcat's startup.sh script has been run, enacting any code changes made to mango only requires running the two ant commands.

Example of all commands to launch mango:

```
morgan@computer:~/Documents/mango$ ./apache-tomcat-9.0.65/bin/startup.sh
Using CATALINA_BASE:   /home/morgan/Documents/mango/apache-tomcat-9.0.65
Using CATALINA_HOME:   /home/morgan/Documents/mango/apache-tomcat-9.0.65
Using CATALINA_TMPDIR: /home/morgan/Documents/mango/apache-tomcat-9.0.65/temp
Using JRE_HOME:        /home/morgan/Documents/mango/jdk1.8.0_202
Using CLASSPATH:       /home/morgan/Documents/mango/apache-tomcat-9.0.65/bin/bootstrap.jar:/home/morg
an/Documents/mango/apache-tomcat-9.0.65/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
morgan@computer:~/Documents/mango$ cd mangoSource/
morgan@computer:~/Documents/mango/mangoSource$ ant fullDeploy
Buildfile: /home/morgan/Documents/mango/mangoSource/build.xml

compile:
    [javac] Compiling 1 source file to /home/morgan/Documents/mango/mangoSource/build/WEB-INF/classes

static:

build:

createConfigFiles:
     [copy] Copying 1 file to /home/morgan/Documents/mango/mangoSource/build/WEB-INF/classes
     [copy] Copying 1 file to /home/morgan/Documents/mango/mangoSource/build/WEB-INF/classes

fullBuild:

copy:
     [copy] Copying 3 files to /home/morgan/Documents/mango/apache-tomcat-9.0.65/webapps/test

fullDeploy:

BUILD SUCCESSFUL
Total time: 0 seconds
morgan@computer:~/Documents/mango/mangoSource$ ant reload
Buildfile: /home/morgan/Documents/mango/mangoSource/build.xml

reload:
   [reload] OK - Reloaded application at context path [/test]

BUILD SUCCESSFUL
Total time: 48 seconds
morgan@computer:~/Documents/mango/mangoSource$
```

Mango homepage (the username and password are both **admin**):



Testing

The only extra pieces of software that are required for testing are a browser and its corresponding driver. This tutorial uses firefox and its gecko browser driver, other browsers such as chrome should work similarly. We download and extract the gecko driver from its github repo, make sure it's executable, and move it to /usr/local/bin (any directory in your PATH will work).

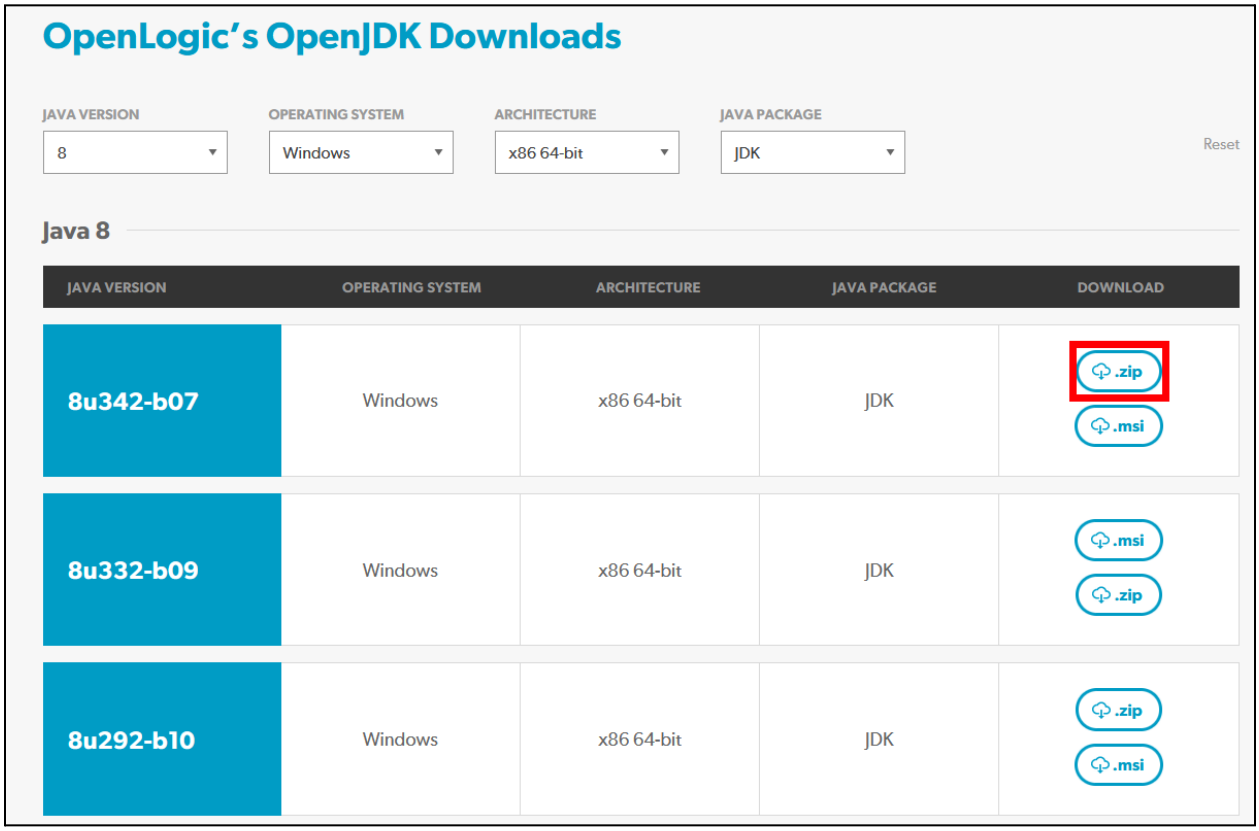
We then need to manually install firefox. On newer versions of Ubuntu, firefox is installed via snap when using the package manager. However, due to the way that snap installs its packages, the gecko driver will be unable to communicate with firefox if it's installed via snap. Fortunately, Mozilla has instructions for manually installing firefox on linux, which can be found here. After completing these instructions, it is recommended that you remove your snap installation of firefox by running **`sudo snap remove firefox`**.

To test mango, simply navigate to the mangoSource directory and run the command **`ant test`**. This will give a log4j error, but will give accurate results. The log4j error is caused by manually running mango classes outside of apache tomcat's context. To fix this, we can manually set the tomcat path in mangoSource/templates/log4j.xml, and replace "${catalina.home}" with the path of our apache tomcat directory. In our case, this is /home/morgan/Documents/mango/apache-tomcat-9.0.65. Errors may still persist, as mango needs to be rebuilt and re-deployed. This can be done by running the command **`ant fullTest`**.

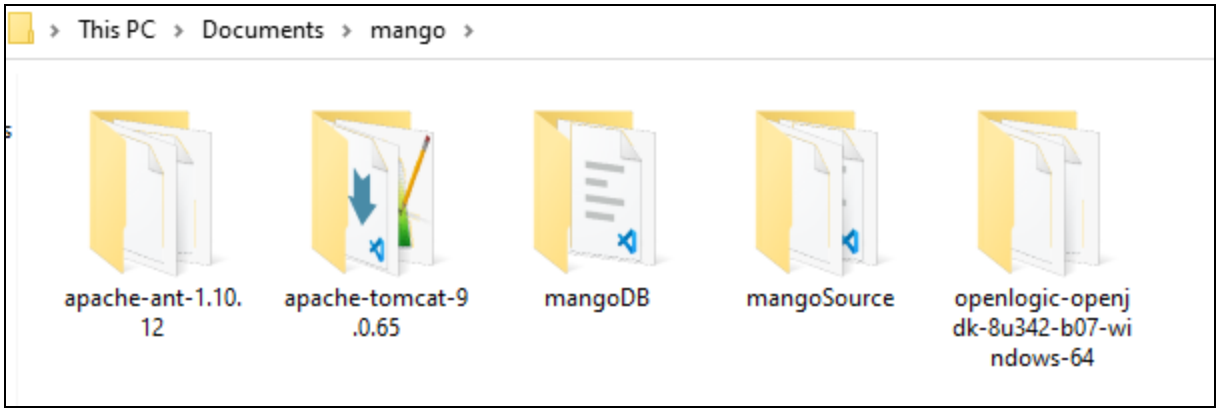
Windows Installation

File Setup

First, download the software listed in the **Required Software** section. Note that for Java 8, only the installation wizard is available on Oracle's website. This guide will show how to use Java 8 without installing it, as it may cause problems for your computer when running a program that requires a newer java version. To do this, you may download the raw java 8 installation files from here. Make sure to download the **.zip** version. If you are unconcerned with this, you may use Oracle's java 8 JDK installer instead.



Once everything has been downloaded, create a folder for your project and extract everything into it. In our case, we have created a folder in C:\Users\morga\OneDrive\Documents\mango.

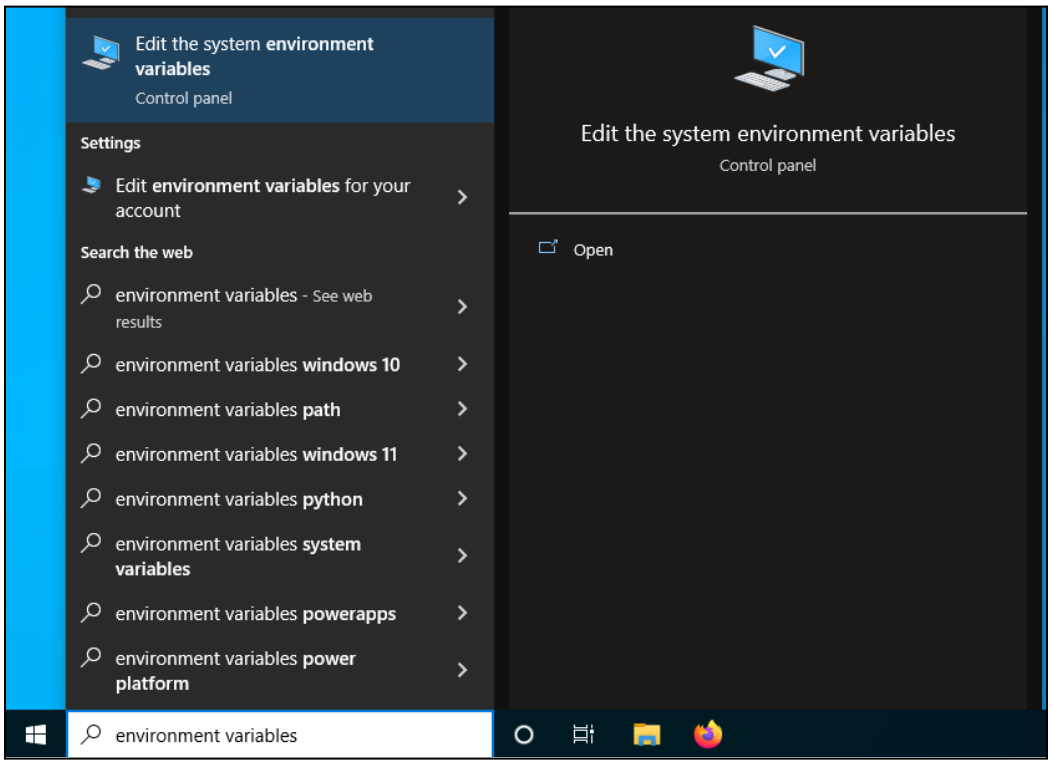


Environment Variables

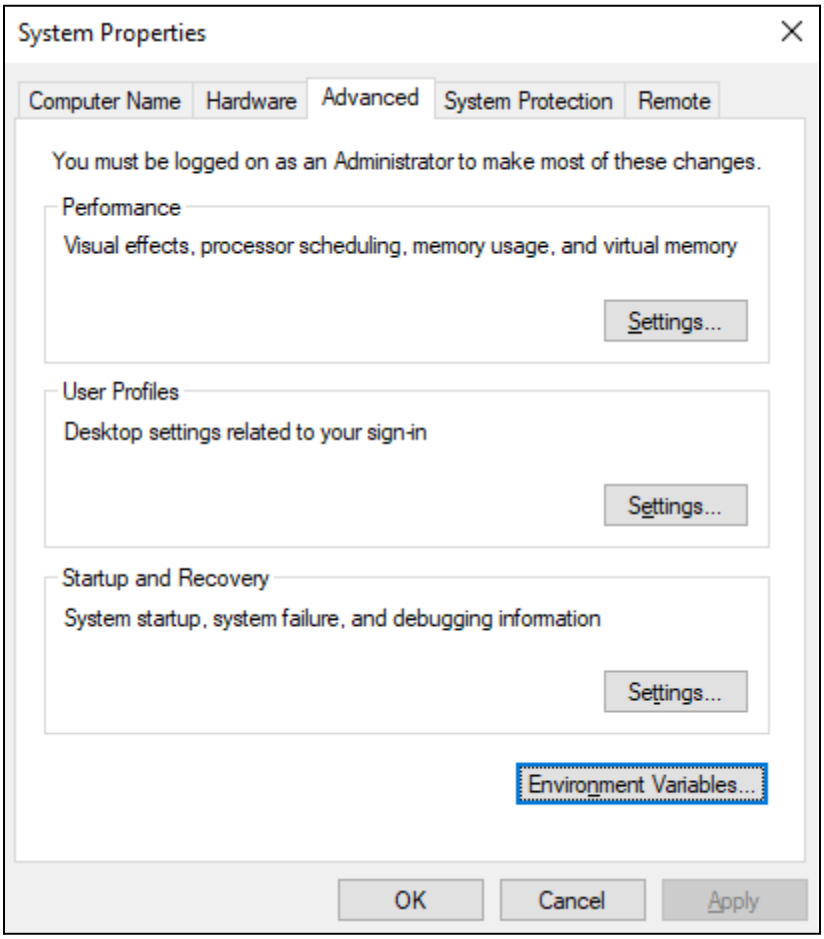
Make note of the absolute paths of ant's bin folder and the java JDK folder. In our case, these are:

C:\Users\morga\OneDrive\Documents\mango\apache-ant-1.10.12\bin
C:\Users\morga\OneDrive\Documents\mango\openlogic-openjdk-8u342-b07-windows-64

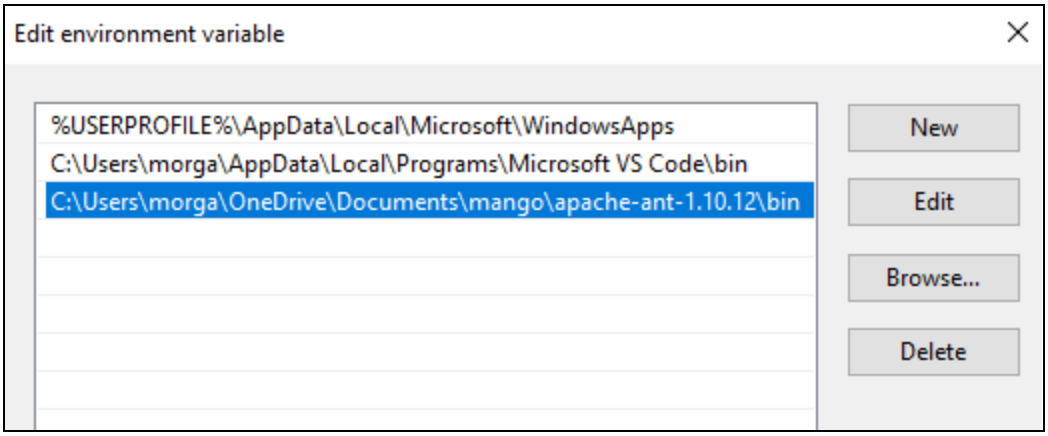
If you used Oracle's JDK installation wizard, you may skip any step in this section that involves the second path. First, we need to navigate to search "environment variables" in the windows search bar.



This should open a “System Properties” window.
Next, click the **Environment Variables…** button.



Finally, we can start editing the Environment Variables. Click on the **Path** entry in the top section (User variables section), and then click the **Edit** button. This should open a new window with several paths. Click **New** and paste your ant bin path into the new entry. Your Path entry should look similar to this:

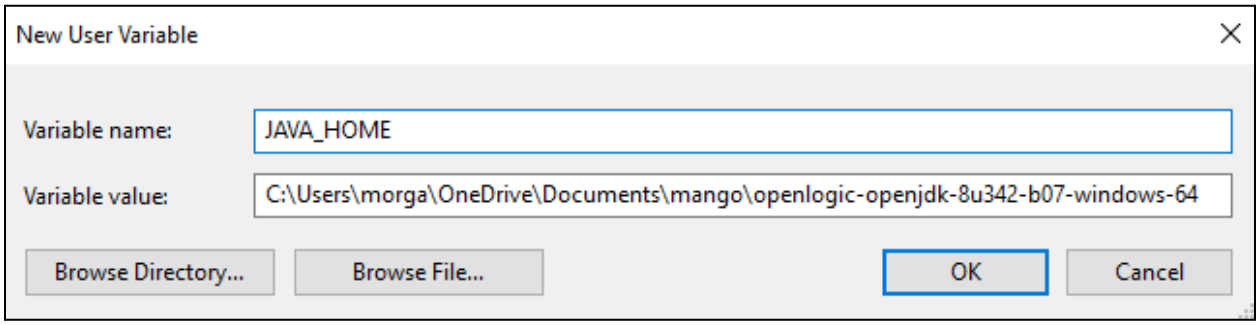


Click **OK**, then look for a **JAVA_HOME** entry in the lower section (System variables). If one already exists, delete it by clicking it, then clicking the **Delete** button.

Now, in the upper section (User variables):

- Click the lower **New…** button
- Enter **JAVA_HOME** into the **Variable name:** field
- Enter your java folder's path into the **Variable value:** field
- Click OK

Before you click OK, you should see something similar to the following:



You may now close everything.

Tomcat Configuration

Next, we need to configure tomcat. Edit the conf/tomcat-users.xml file in the tomcat directory, and add the following lines:

```
<role rolename="manager-gui" />
<user username="admin" password="admin"
        roles="manager-gui,manager-script" />
```

Ex:

C: > Users > morga > OneDrive > Documents > mango > apache-tomcat-9.0.65 > conf > tomcat-users.xml

```
19                  xmlns:xsi= http://www.w3.org/2001/XMLSchema-instance
20                  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
21                  version="1.0">
22    <role rolename="manager-gui"/>
23    <user username="admin" password="admin" roles="manager-gui,manager-script"/>
```

Configuring mango's build.properties

Open mangoSource/build.properties, and change the tomcat.home and db.url fields to the corresponding paths. If you copied and pasted these paths, remove the **C:** prefix, and **make sure you change your forward slashes to backslashes.** If you use forward slashes (even if you escape the forward slashes by replacing them with "\\"), the database will fail to load. Your file should look something like this:

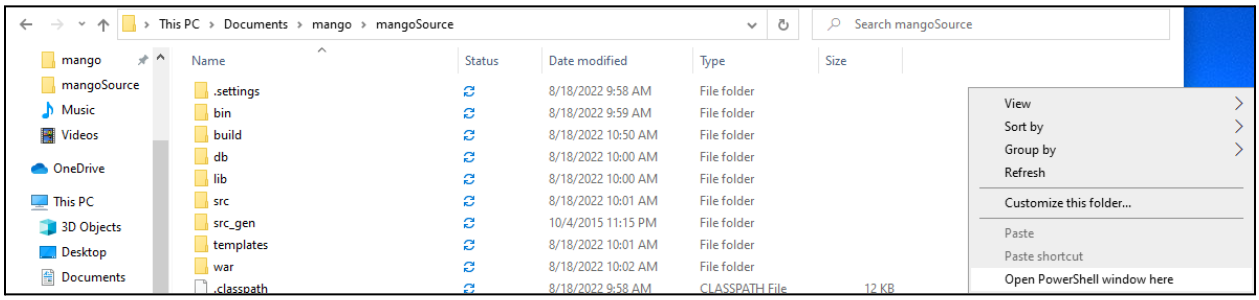
C: > Users > morga > OneDrive > Documents > mango > mangoSource > build.properties

```
22
23    tomcat.home=/Users/morga/OneDrive/Documents/mango/apache-tomcat-9.0.65
24    #Ex: tomcat.home=/home/morgan/Documents/mango/apache-tomcat-9.0.65
25
26    tomcat.manager.url=http://localhost:8080/manager/text
27    tomcat.manager.username=admin
28    tomcat.manager.password=admin
29    tomcat.hostname=webapps
30
31    # App name and app path are separated to accomodate root deployments. (
32    tomcat.appdir=test
33    tomcat.apppath=/test
34
35    log4j.xml.logThreshold=warn
36
37    db.type=derby
38
39    db.url=/Users/morga/OneDrive/Documents/mango/mangoDB
```

Quickstart

To run mango, first navigate to tomcat's bin folder and run the **startup.sh** file. There may be some "failed to load image" errors, but that doesn't mean that it isn't working. Next, navigate to

the mangoSource folder, hold the “Shift” key, and right-click on empty space as if you were creating a new folder. You should see an **Open PowerShell window here** option. Click it.



Run the following commands:
ant fullDeploy
ant reload.

```
PS C:\Users\morga\OneDrive\Documents\mango\mangoSource> ant fullDeploy
Buildfile: C:\Users\morga\OneDrive\Documents\mango\mangoSource\build.xml

compile:
    [javac] Compiling 1 source file to C:\Users\morga\OneDrive\Documents\mango\mangoSource\build\WEB-INF\classes

static:

build:

createConfigFiles:
     [copy] Copying 1 file to C:\Users\morga\OneDrive\Documents\mango\mangoSource\build\WEB-INF\classes
     [copy] Copying 1 file to C:\Users\morga\OneDrive\Documents\mango\mangoSource\build\WEB-INF\classes

fullBuild:

copy:
     [copy] Copying 3 files to C:\Users\morga\OneDrive\Documents\mango\apache-tomcat-9.0.65\webapps\test

fullDeploy:

BUILD SUCCESSFUL
Total time: 2 seconds
PS C:\Users\morga\OneDrive\Documents\mango\mangoSource> ant reload
Buildfile: C:\Users\morga\OneDrive\Documents\mango\mangoSource\build.xml

reload:
   [reload] OK - Reloaded application at context path [/test]

BUILD SUCCESSFUL
Total time: 8 minutes 6 seconds
PS C:\Users\morga\OneDrive\Documents\mango\mangoSource>
```

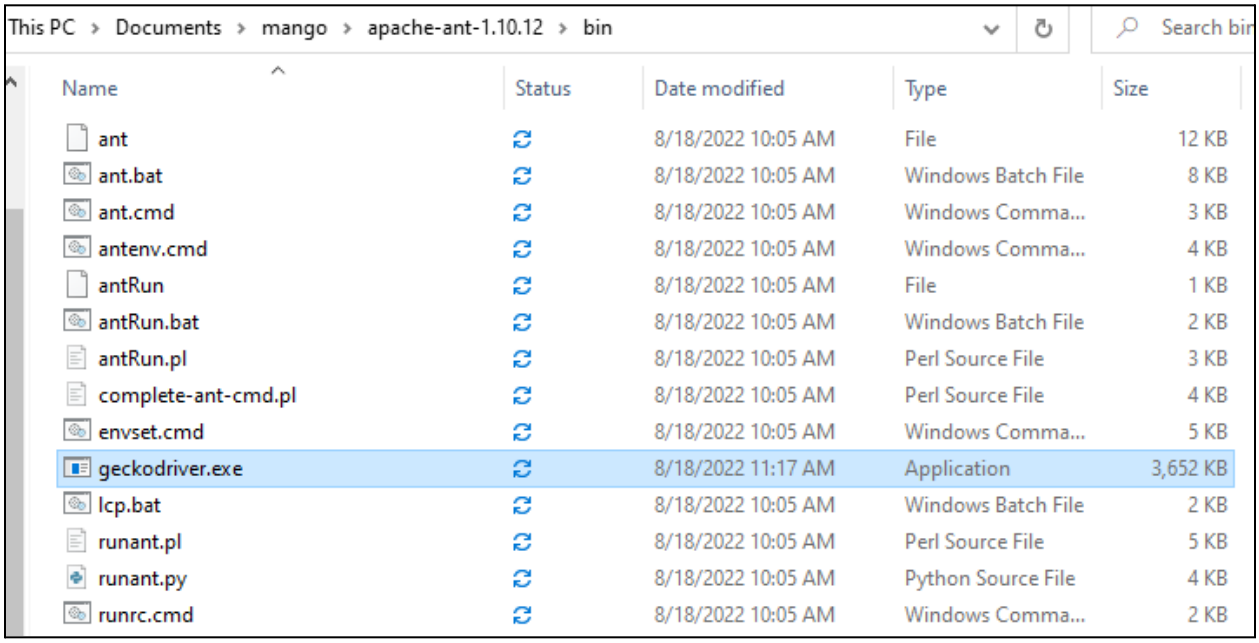
You should now be able to open a web browser and log into mango by navigating to http://localhost:8080/test. The username and password are both **admin**.

Note that once tomcat's startup.sh script has been run, enacting any code changes made to mango only requires running the two ant commands.

Testing

The only additional software required to test mango is a browser and its corresponding web driver. Most modern browsers should work, but we will be using **firefox** and its **gecko** browser driver.

This installation is mostly what you would expect. Simply download and install firefox from Mozilla's website. Installing geckodriver is a bit less straightforward. Download it from its github repo, and place the **geckodriver.exe** file into ant's bin folder.

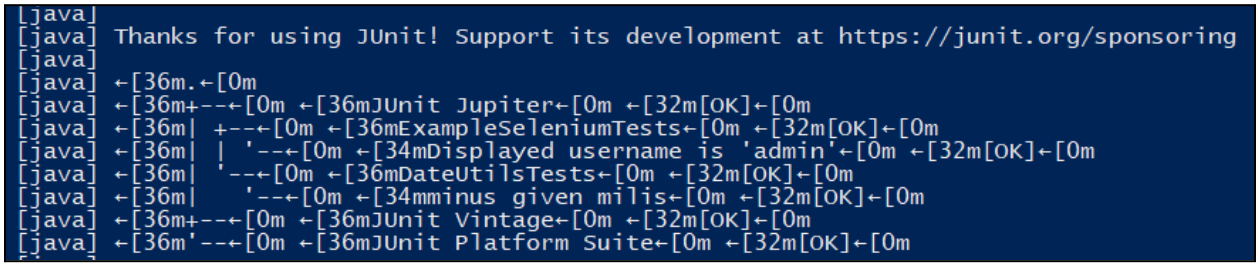


To test mango, simply navigate to the mangoSource directory and run the **`ant test`** command. This will give a log4j error, but will give accurate results. The log4j error is caused by manually running mango classes outside of apache tomcat's context. To fix this, we can manually set the tomcat path in mangoSource/templates/log4j.xml, and replace "${catalina.home}" with the path of our apache tomcat directory. In our case, this is /Users/morga/OneDrive/Documents/mango/apache-tomcat-9.0.65. Errors may still persist, as mango needs to be rebuilt and re-deployed. This can be done by running the **`ant fullTest`** command.

Terminal Recommendation

Note that the testing framework uses colored output by default using ANSI escape sequences. Windows PowerShell does not support this, but Windows Terminal does. If you plan on testing mango, we recommend that you install Windows Terminal from the Microsoft Store app. The main caveat is that opening Windows Terminal from a folder is not possible. This means that you will need to manually navigate to your mangoSource folder by entering the **cd [mangoSource path here]** command.

Test output using Powershell:



Test output using Windows Terminal: