

# Quick Start

2016 QuickStart

See <http://lists.apple.com/archives/webobjects-dev/2016/Jan/msg00045.html>

**Steps 0-2** (as elaborated below)

0. Install Java 8
1. Install Maven 3
2. Update `~/m2/settings.xml` to include the wocommunity repo

**Steps 3-6**

3. Install Eclipse 4.5 for Committers (<http://www.eclipse.org/downloads/>)
4. Install m2e via Help > Install New Software.... (See <http://www.eclipse.org/m2e/>)
5. Install [WOLips](#)
6. Preferences > Maven > Archetypes, Add Remote Catalog  
<http://maven.wocommunity.org/service/local/repositories/snapshots/content/archetype-catalog.xml>

Done. The WebObjects jars are in the wocommunity repo, so no need to do `winstaller.jar` hassle.

---

## Detailed Instructions...

### Step 0: Install latest Java

If you're on a Mac, it's using old and busted Java 6. Update that to Java 8 (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

### Step 1: Installing Maven

Downloading and installing the latest version of [Maven](#) is the first step to start using this tool via one of the following approaches.

(a) You can use MacPorts (<https://www.macports.org>) to install:

#### MacPorts usage

```
$ sudo port install maven3
```

(b) Or similarly via Homebrew (<http://brew.sh>):

#### Homebrew usage

```
$ brew install maven
```

(c) Or manually by downloading the binary tarball, copying it to `/usr/local/`, unpacking it, soft linking it and adding the path to your `bash_profile`:

#### Manual install

```
$ sudo su
$ cd /usr/local/
$ curl -O http://mirror.tcpcdiag.net/apache/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.tar.gz
$ tar -xzf apache-maven-3.3.3-bin.tar.gz
$ ln -s apache-maven-3.3.3 apache-maven
$ exit
$ cd ~/
$ vi .bash_profile
```

Then in your bash profile, ensure your path and java home are updated:

## ~/.bash\_profile

```
...
export PATH=/usr/local/apache-maven/bin:$PATH
export JAVA_HOME=$( /usr/libexec/java_home )
...
```

Whichever installation approach you've taken, you should now be able to do

```
$ mvn -version
```

**NOTE 1:** It is very helpful if you understand some Maven concepts before continuing with this tutorial. See [General Maven Documentation](#) for more information.

**NOTE 2:** You might also like to install bash completion utilities:

- See this superuser answer for bash-completion installation: (<http://superuser.com/questions/288438/bash-completion-for-commands-in-mac-os-x>).
- See also the following github repo for maven-bash-completion: (<https://github.com/juven/maven-bash-completion>).

## Step 2: Configuring your Maven Settings

Advanced users can skip this step, but it's highly recommended.

[Quick Start Maven Settings Configuration Guide](#)

**NOTE for those who know what to do:** You can download the complete and most recent settings.xml [here](#) and install it to ~/.m2/settings.xml.

Download the [archetype-catalog.xml](#) and copy it to ~/.m2/archetype-catalog.xml

See [Maven Settings Reference](#) if you want more information about additional options for the settings.xml file.

## Step 3: Installing WebObjects Libraries

Maven takes control of all your project's dependencies. To develop WebObjects applications, Maven has to reference the WebObjects jars. You have to install the WebObjects libraries into your local repository to accomplish this:

```
mvn wbootstrap:install -DwebObjectsLibFolder=/Library/WebObjects/lib -DwebObjectsVersion=5.4.3
```

**NOTE:** WebObjects must be installed.

See the [maven-wbootstrap-plugin](#) documentation for more information and options for specifying a specific version.

See [Maven Introduction to Repositories](#) if you want more information about Maven repositories.

Installing the right jars!

 Do **not** link to or install the WebObjects jars located inside /System/Library/Frameworks or /Library/Frameworks. If you're installing an older version of WebObjects you must obtain the jars that the WebObjects installer had previously installed into /Library/WebObjects/lib. For example, the jar /Library/WebObjects/lib/JavaWebObjects.jar has the complete set of resources and files included for the dependency JavaWebObjects whereas /System/Library/Frameworks/JavaWebObjects.framework/Resources/Java/javawebobjects.jar does not.

## Step 4: Creating a WebObjects or Wonder Project

Archetypes are the fast way to create a new project using Maven and WOProject provides various archetypes of its own to get you started with WebObjects, or better, Wonder-based applications. Execute the following command to generate a basic Wonder application project:

```
mvn archetype:generate -DarchetypeArtifactId=erxapplication-archetype \
    -DarchetypeGroupId=org.objectstyle.woproject.maven2 \
    -DarchetypeVersion=2.1
```

The maven-archetype-plugin will ask the required information to create the new project.

**NOTE:** You can use archetype catalogs to reduce the number of properties to set while creating a project. See the documentation of [woapplication-archetype](#).

**NOTE:** You can use archetypes within Eclipse as described [here](#).

**NOTE:** You can see the list of available archetypes in the following location: <http://wo-repository.moleque.com.br/nexus/content/groups/public/org/objectstyle/woproject/maven2/>.

## Step 5: Importing the Project into Eclipse

The new project is ready to be imported into Eclipse workspace. Use the Eclipse's import wizard (File > Import...) and choose the option "Existing Projects into Workspace". Select the root directory of your new project and finish the import wizard.

Warning



As Maven takes care of the dependency management, you must use a mechanism to add the jars to your Eclipse build path. There are 3 options:

1. Use the `mvn eclipse:eclipse` goal to update your `.classpath` file.
2. Use the `m2eclipse` plug-in for Eclipse.
3. Use the `q4e` plug-in for Eclipse.

After adding the project dependencies to your Eclipse build path, WOLips can build the project and you will be able to start developing and running the application.

**NOTE:** This tutorial assumes you have chosen the option 2. You can find more information about IDE integration [here](#).

## Step 6: Building WebObjects Applications with Maven

Now, building your project with Maven is easy. Go to the project folder on Terminal and execute:

```
mvn clean package
```

This goal will generate a WOA package inside the target folder of your project. It also generates two compressed packages: one for woapplication and other for the webserver resources.

To install your project into your local repository:

```
mvn clean install
```

This will install your project into `~/.m2/repository/your/project/groupId/artifactId/version/...`