

Zero Footprint Portable JEE Eclipse Development Environment

- [Introduction](#)
- [Eclipse Setup](#)
 - [Select Location](#)
 - [Setup Package Java](#)
 - [Modify eclipse.ini](#)
- [Launch Eclipse](#)
 - [Clicking to Add to GateKeeper Rules](#)
 - [Manually Adding Eclipse to GateKeeper Rules](#)
- [Setup Tomcat](#)
- [Configure Eclipse to Use Tomcat](#)

Introduction

Having a a portable environment to easily share among developers using the [zero footprint service](#) approach if possible.

This article is being pulled from the [Getting Started](#) and [Manual Eclipse Java Tomcat Setup](#) documents from www.tinframework.homeip.net.

There is a working implementation of this for SiteMesh at the [SiteMesh Wiki](#).

The BonsaiFramework uses a portable Java EE development environment that is composed of the following components,

- Eclipse
- Java
- Tomcat

To get quickly started download the package that matches your operating system.

Package	Operating Systems
tbd	Mac OS X
tbd	Windows 7 32-bit
tbd	Windows 7 64-bit

Let's try and do Eclipse using Java 64 but then have Tomcat 32-bit use Java 32-bit.

Eclipse Setup

You want to download the portal (tar.gz for Linux, zip for Windows) version of Eclipse and not the installer.

Following the portability principle of the Bonsai Framework there is no installer and the software is isolated from the operating system.

The remainder of this document provides details about how to build/adjust the portable environment yourself. These same instructions should also work for the [SpringSource Tool Suite](#).

Download [Eclipse IDE for Java EE Developers](#).

Select Location

Select a drive and directory your computer where you will be keeping everything. The Bonsai Framework uses,

- Windows = C:\apps\
- Linux or Unix = /home/[your home directory]/apps for example, /home/tpham/apps/
- Mac OS X = same as Linux

Mac OS X has a special file system for applications. You can not just browse to it normally. Go to your "apps" directory and you should see Eclipse.app,

1. Right click
2. Select "Show Package Contents"
3. Go into the "Eclipse" folder

Setup Package Java

Download Oracle's JRE (Java Runtime Environment) and follow the [manual Java setup instructions](#).

Move the Java folder into your eclipse directory. For example,

- Windows = C:\apps\eclipse\jre1.6.0_18
- Linux or Unix = /home/tpham/apps/eclipse/opt/jre1.6.0_18
- Mac OS X = same as Linux

Modify eclipse.ini

Modify the eclipse.ini to point to your JRE with by adding -vm and a reference to Java.

See [Specifying the JVM at the Eclipse Wiki](#) to understand why.

Specifically with Mac OS, getting to eclipse.ini takes a few extra steps,

1. Right click the Eclipse package.
2. Select, **Show Package Contents**.
3. Expand folders, Contents, Eclipse
4. You should see eclipse.ini.

Edit with [solid text editor](#).

If you are using STS on Mac OS X, first you need to use the JDK (Java Development Kit) not the JRE (Java Runtime Environment)

1. Go to your STS installation, and right-click on STS Application
2. Select 'Show Package Contents'
3. A new Finder window will open and show the content of the application bundle.
4. In Contents/MacOS is STS.ini which is equivalent of eclipse.ini

Two key thing to keep in mind,

1. You can **not** mix 32-bit and 64-bit. For example, running Eclipse 32-bit and Java 64-bit will not work.
2. Placement of the -vm option is very important. The -vm option must occur before the **-vmargs** option.
3. If everything appears on one line use a better free editor like [Notepad++](#).

Here is an example of what the **eclipse.ini** would look like for **Windows**,

```

-startup
plugins/org.eclipse.equinox.launcher_1.0.201.R35x_v20090715.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_1.0.200.v20090519
-product
org.eclipse.epp.package.jee.product
--launcher.XXMaxPermSize
256M
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
# -----
# Start Bonsai Framework
# -----
# Use local version of Java.
# Note, these lines must come just before -vmwargs
-vm
jre1.6.0_18\bin\javaw.exe
# -----
# End Bonsai Framework
# -----
-vmargs
-Dosgi.requiredJavaVersion=1.5
-Xms40m
-Xmx512m

```

Here is an example of **eclipse.ini** for Unix or Linux change the jre line to,

```

# -----
# Start Bonsai Framework
# -----
# Use local version of Java.
# Note, these lines must come just before -vmwargs
-vm
jre1.6.0_18/bin/java
# -----
# End Bonsai Framework
# -----

```

For Mac OS X the jre line is slightly different and is unique to your **own** home directory,

```
# -----  
# Start Bonsai Framework  
# -----  
# Use local version of Java.  
# Note, these lines must come just before -vmwargs  
-vm  
/Users/tin.pham/apps/eclipse/opt/java/jre1.8.0_121.jre/Contents/Home/bin/j  
ava  
# -----  
# End Bonsai Framework  
# -----
```

Launch Eclipse

Start Eclipse by running the Eclipse executable included with the release. In our example the directories would be,

- Windows = C:\apps\eclipse\eclipse.exe
- Linux or Unix = /home/tpham/apps/eclipse/eclipse
- Mac OS X = same as Linux

If using Mac OS X you will receive an error due to modification of the eclipse.ini file,

"Eclipse.app" is damaged and can't be opened. You should move it to the Trash.

There are two ways of fixing this.

Clicking to Add to GateKeeper Rules

By manually adding Eclipse, you can remove yourself in the future.

Open the mac terminal and type the following,

```
su - setup.admin # My normal Mac User does not have sudo privileges, yours  
might.  
cd ~/apps/  
sudo spctl --master-disable
```

Then the error message turns to,

"Eclipse.app" is an application downloaded from the Internet. Are you sure you want to open it?

Click "open". This will add Eclipse into the GateKeeper Rules and everything should start.

Then enable the protection again,

```
sudo spctl --master-enable
```

Now Eclipse will start properly.

When time permits, I will do a capture of the all registered rules to see exactly what the new rule fingerprint looks like and document the process here.

Manually Adding Eclipse to GateKeeper Rules

...

This would be the preferred method but I found it did not quite work right. I noticed a typo so need to try again on a fresh machine.

Open the mac terminal and type the following,

```
su - setup.admin # My normal Mac User does not have sudo privileges, yours
might.
spctl --add --label "eclipse" Eclipse.app # Eclipse.app is the Eclipse
package.
```

...

Look for the entry,

```
su - setup.admin # My normal Mac User does not have sudo privileges, yours
might.
spctl --list | grep -i "eclipse"
```

You can remove the Eclipse by typing,

```
su - setup.admin # My normal Mac User does not have sudo privileges, yours
might.
spctl --remove --label "eclipse" Eclipse.app # Eclipse.app is the Eclipse
package.
```

Setup Tomcat

Similar to Eclipse, Tomcat also is also packaged without an installer. For windows we download the x86 .zip and for Linux or Unix download the .tar.gz file. Following the Bonsai Framework standards, decompress to the following folder,

- C:\apps\eclipse-servers\apache-tomcat-6.0.24
- /home/tpam/apps/eclipse-servers/apache-tomcat-6.0.24

Configure Eclipse to Use Tomcat

...

Ok this is where we have a problem. Once the Tomcat instance is created there seems to be no way to dynamically specify the various classpath entries... double-click on the server, Overview, Open Launch configuration, Classpath.

There it is hard coded. As soon the eclipse directory changes, the server stops working. Tried various variable options with Garth without success.

Now that's it's 2014 maybe this changed... also try talking to Patrick.

...