

# Developing Jetspeed2

## Table of contents

1 Getting Started Developing Jetspeed .....	2
1.1 Introduction .....	2
1.2 Setting up a development environment .....	2
1.3 Setting up Eclipse .....	2
1.4 Setting up Tomcat .....	3
1.5 Checking out Jetspeed .....	3

##language:en

## 1. Getting Started Developing Jetspeed

### 1.1. Introduction

This section is for developers who want to work on Jetspeed 2 itself, not necessarily for portlet developers.

### 1.2. Setting up a development environment

There are many different ways to set up a development environment that will enable you to effectively develop Jetspeed 2. This example uses Eclipse, Tomcat 5, and MySQL.

#### 1.2.1. Prerequisites

Download and install these packages:

- [Java JDK 1.4](http://java.sun.com/) (http://java.sun.com/)
- [Ant 1.5](http://ant.apache.org/) (http://ant.apache.org/)
- [Maven 1.0](http://maven.apache.org/) (http://maven.apache.org/)

### 1.3. Setting up Eclipse

1. Download Eclipse from [eclipse.org](http://www.eclipse.org) (http://www.eclipse.org) . Any Latest Release version from 3.0 onward will probably be OK. If you have problems with any of the plugins, make sure they support the version of Eclipse you downloaded.

At a minimum you will need to install the Mevenide and Checkstyle plugins. Mevenide is used for working with Mavenized projects. Checkstyle is a Java programming style enforcer.

The easiest way to install plugins is through the Eclipse Update Manager. Choose **Help** > **Software Updates** > **Find and Install** . In the next dialog, choose "Search for new features to install" and then "Add Update Site".

Name: [Mevenide](http://mevenide.codehaus.org/mevenide-ui-eclipse/features.html) (http://mevenide.codehaus.org/mevenide-ui-eclipse/features.html) [[BR]]

Update Manager URL:

<http://mevenide.codehaus.org/release/eclipse/update/site.xml>

If you see a java.lang.NoSuchMethodError error with Mevenide during launch, use the following URL for the Update Manager[[BR]]

<http://mevenide.codehaus.org/update/3.0/site.xml>

Unfortunately, installing the Checkstyle plugin is a manual process. Start by downloading the

## Developing Jetspeed2

plugin from the [Checkstyle plugin site](http://eclipse-cs.sourceforge.net/) (<http://eclipse-cs.sourceforge.net/>) . Next, unzip the download and copy the directory to your eclipse/plugins directory. Restart Eclipse if it's running.

You may also want to try [Veloclipse](http://prosorter.sourceforge.net/veloeclipse) (<http://prosorter.sourceforge.net/veloeclipse>) , a plugin for editing Velocity templates. It's available through the Eclipse Update Manager. Use the steps described above for Mevenide, but use the following values.

Name: Veloclipse [[BR]] Update Manager URL:

<http://prosorter.sourceforge.net/veloeclipse> [[BR]]

Some developers also use [XMLBuddy](http://xmlbuddy.com/) (<http://xmlbuddy.com/>) .

The Sysdeo Tomcat plugin is also handy for controlling Tomcat from Eclipse <http://www.sysdeo.com/eclipse/tomcatPlugin.html>. This is a manual install.

There are more Eclipse plugins available at <http://www.eclipse-plugins.info/>.

**Note on the Eclipse workspace:** If you are using Windows, set your Eclipse workspace (in the annoying window that opens up every time you launch Eclipse until you finally notice the checkbox) such that the directory path does not contain any spaces. You will have trouble building if the path to your workspace contains spaces.

### 1.4. Setting up Tomcat

Download and install Tomcat from <http://jakarta.apache.org/tomcat/>.

Tomcat 4 and 5 are both supported. Versions of Tomcat older than 5.0.28 require a JAASRealm patch that will be automatically applied by Maven during the build. Refer to <http://issues.apache.org/jira/browse/JS2-55> for more information. Otherwise, Tomcat 4 or 5 are equally good choices.

### 1.5. Checking out Jetspeed

The first thing you need to do is add a CVS repository: **Window > Open Perspective > CVS Repository Exploring**

Make sure the "CVS Repositories" tab is selected and right-click (Control-click on the Mac) that pane. Choose **New > Repository Location...** in the context menu. Alternatively, you can click the "Add CVS Repository" button to the right of the tabs.

Enter the following information in the Add CVS Repository dialog: [[BR]] Host:

`cv.s.apache.org` [[BR]] Repository path: `/home/cvspublic` [[BR]] User: `anoncv`s  
[[BR]] Leave Password empty. [[BR]]

The connection type is `pserver` , using the default port.

You can then drill down to the `jakarta-jetspeed-2` module from `HEAD`.

Right-click (control-click on the Mac) on the `jakarta-jetspeed-2` module and choose **Check Out As...** from the context menu.

Select "Check out as a project in the workspace" and click the Finish button.

Now switch from the CVS perspective to the Java perspective ( **Window > Open Perspective** ; you may need to choose **Other...** and select Java from the resulting dialog).

Presto! You have a checkout of `jakarta-jetspeed-2`. You'll probably get a ton of errors. Luckily they're all variations on the same theme: `Unbound classpath variable: MAVEN_REPO/blah/blah/blah` .

Before we go any further, make sure that the Maven settings are correct in Eclipse. Open the Eclipse Preferences dialog and select the **Maven > Locations** . In the author's experience on Mac OS X, it was necessary to enter the Java Home path as a URL:

```
file:/System/Library/Frameworks/JavaVM.framework/Versions/1.4.2/Home
```

Maven Home will probably be something like: `[[BR]] /usr/local/maven-1.0`

Maven Local Home: `[[BR]] /your/home/.maven`

Maven Local Repository: `[[BR]] /your/home/.maven/repository`

OK -- now we're ready to fix those `MAVEN_REPO` errors. In the Preferences window, click **Java > Build Path > Classpath Variables** . Click the "New..." button and enter the following:

```
Name: MAVEN_REPO [[BR]] Path: the path to your Maven Local Repository, i.e.  
/your/home/.maven/repository
```

Click **OK** until you've saved your preferences and the Preferences window is closed.

The build errors should disappear. Right-click (control-click) the project name "jakarta-jetspeed-2" and choose **Maven > Run Maven...** . In the text field to the left of the "Choose..." button, enter `allClean allBuild` and click "Run".

That's it, you're building Jetspeed! Any build problems from here on are left as an exercise to the reader. :)