Windows Guide to Installing Wicket on Eclipse with Maven

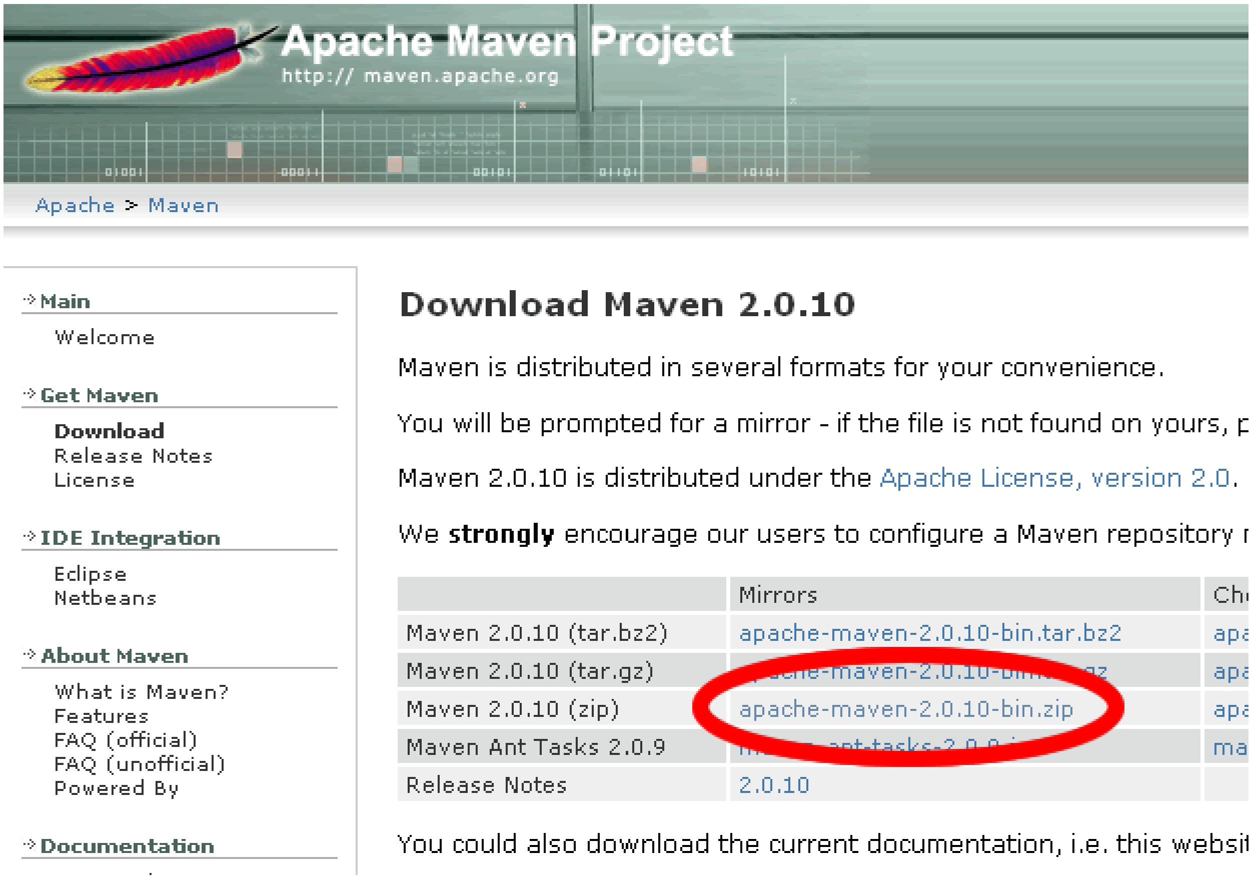
Windows Guide to Installing Wicket on Eclipse with Maven

Download & Install Maven

Maven is a project management tool. It does a million things, but I only know three of those things. We'll use Maven to get the Wicket Quickstart, convert it to an Eclipse project, and package our WARs. But first we need to get Maven.

1. Go to http://maven.apache.org/download.html.

2. Click the "apache-maven-2.0.10-bin.zip" link.



Download Maven 2.0.10

Maven is distributed in several formats for your convenience.

You will be prompted for a mirror - if the file is not found on yours, p

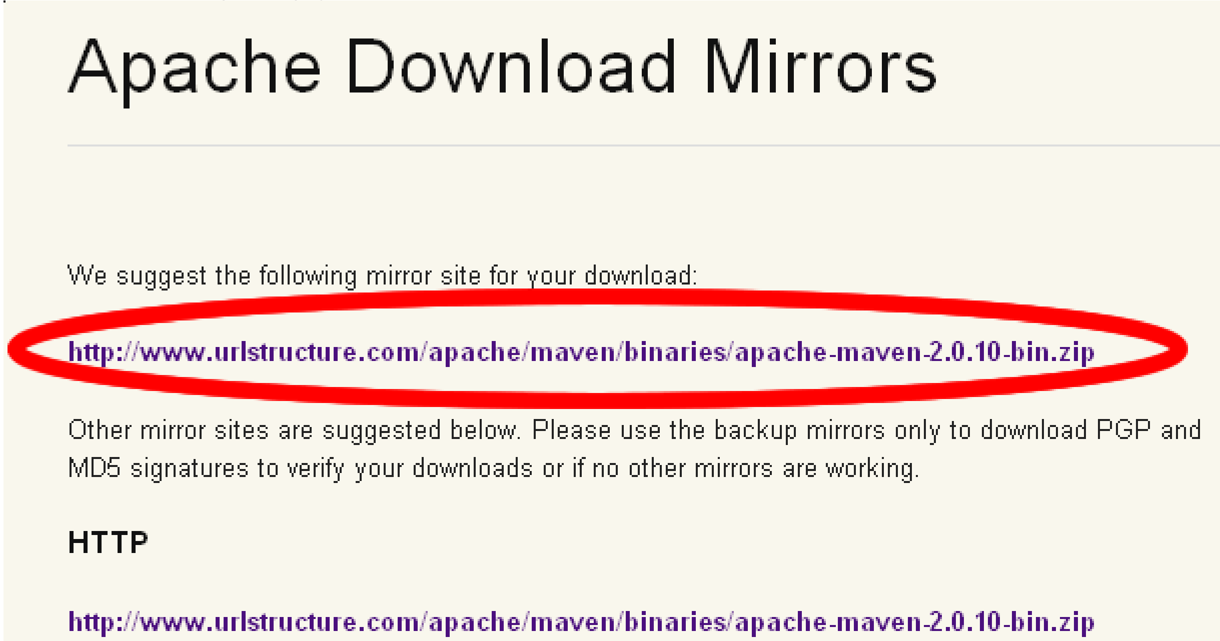
Maven 2.0.10 is distributed under the Apache License, version 2.0.

We **strongly** encourage our users to configure a Maven repository r

	Mirrors	Ch
Maven 2.0.10 (tar.bz2)	apache-maven-2.0.10-bin.tar.bz2	apa
Maven 2.0.10 (tar.gz)	apache-maven-2.0.10-bin.tar.gz	apa
Maven 2.0.10 (zip)	apache-maven-2.0.10-bin.zip	apa
Maven Ant Tasks 2.0.9	maven-ant-tasks-2.0.9.jar	ma
Release Notes	2.0.10	

You could also download the current documentation, i.e. this websi

3. Click the link at the top of the page.



Apache Download Mirrors

We suggest the following mirror site for your download:

http://www.urlstructure.com/apache/maven/binaries/apache-maven-2.0.10-bin.zip

Other mirror sites are suggested below. Please use the backup mirrors only to download PGP and MD5 signatures to verify your downloads or if no other mirrors are working.

HTTP

http://www.urlstructure.com/apache/maven/binaries/apache-maven-2.0.10-bin.zip

4. That will prompt you to download a file called *apache-maven-2.0.10-bin.zip* . Save it to your desktop.

5. Inside the zip file is a folder called *apache-maven-2.0.10*. Drag that folder and put it directly onto your C: drive--~~*C:\apache-maven-2.0.10 .*~~ ~~**IMPORTANT**~~*--Maven has problems if its path has any spaces in it. For example, don't put Maven under your "Program Files" folder, since there's a space in "Program Files".*

6. Add an environment variable called MAVEN_HOME with a value of *C:\apache-maven-2.0.10* . You can learn how to set environment variables here: http://www.chem.gla.ac.uk/~louis/software/faq/q1.html#winXP

7. Add Maven's "bin" directory to your PATH environment variable. In this case, you can just paste
; C:\apache-maven-2.0.10\bin
to the end of your PATH. Don't forget to put the semicolon in front, to separate it from anything that's already in your PATH.

8. If you don't have a JAVA_HOME environment variable, then create one of those too, pointing at a JDK installation on your computer.

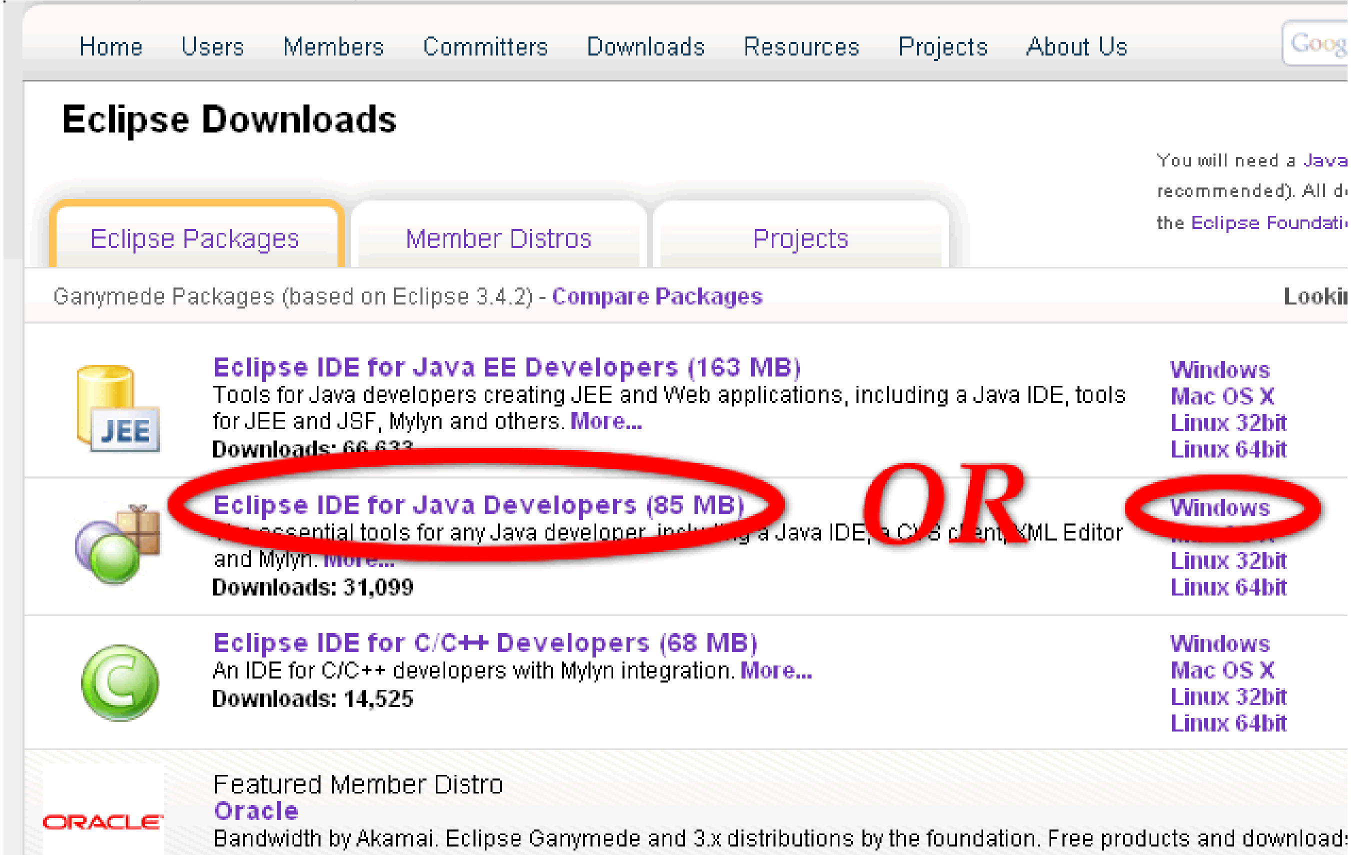
9. Now you have Maven ready to go. You can check that it's been installed correctly by opening a command prompt (Start -> Run... -> cmd) and typing in "mvn -version". That should display a couple lines of information about your Maven installation.

Download Eclipse

Eclipse is a great IDE for developing Java applications.

1. Go to http://www.eclipse.org/downloads/ .

2. Click the *Eclipse IDE for Java Developers* link.



Eclipse Downloads

You will need a Java
recommended). All d
the Eclipse Foundati

Ganymede Packages (based on Eclipse 3.4.2) - **Compare Packages**

Lookii

Eclipse IDE for Java EE Developers (163 MB)
Tools for Java developers creating JEE and Web applications, including a Java IDE, tools for JEE and JSF, Mylyn and others. **More...**
Downloads: 66,633

Windows
Mac OS X
Linux 32bit
Linux 64bit

Eclipse IDE for Java Developers (85 MB)
The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. **More...**
Downloads: 31,099

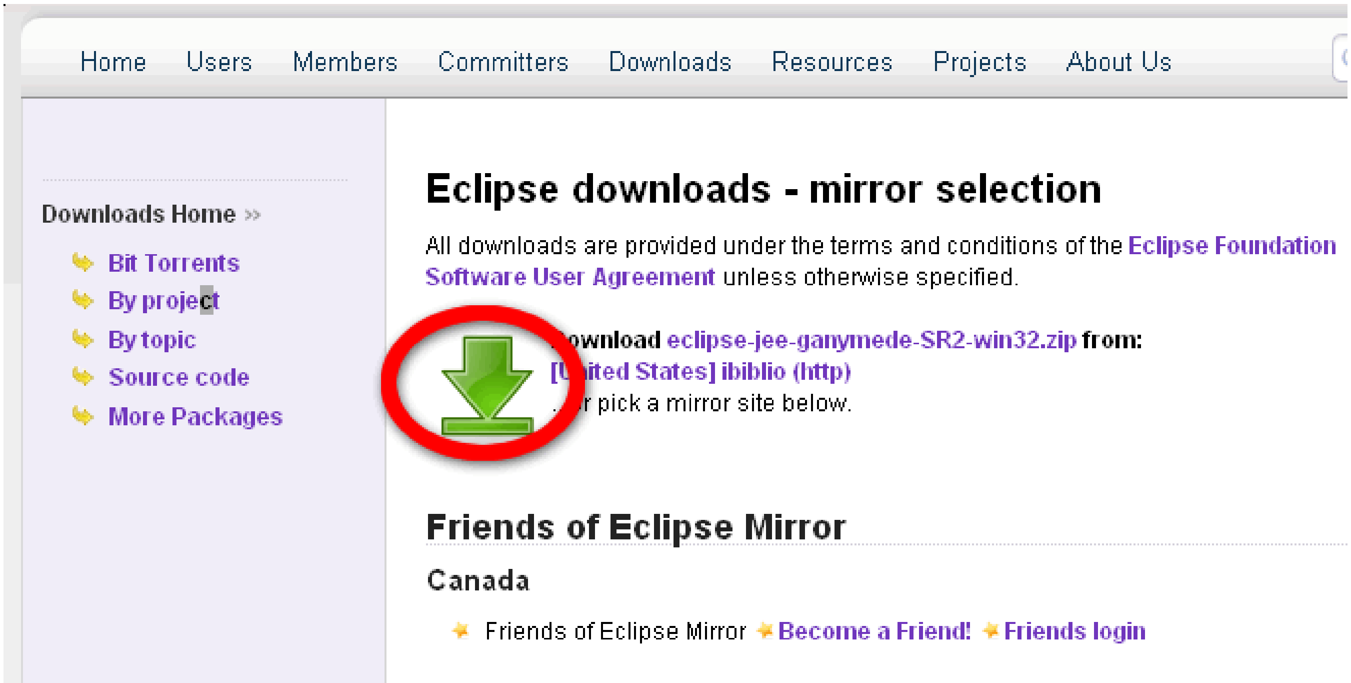
Windows
Mac OS X
Linux 32bit
Linux 64bit

Eclipse IDE for C/C++ Developers (68 MB)
An IDE for C/C++ developers with Mylyn integration. **More...**
Downloads: 14,525

Windows
Mac OS X
Linux 32bit
Linux 64bit

Featured Member Distro
Oracle
Bandwidth by Akamai. Eclipse Ganymede and 3.x distributions by the foundation. Free products and download

3. Click the big green download arrow.



Eclipse downloads - mirror selection

All downloads are provided under the terms and conditions of the **Eclipse Foundation Software User Agreement** unless otherwise specified.

Download eclipse-jee-ganymede-SR2-win32.zip from:
[United States] ibiblio (http)
... or pick a mirror site below.

Friends of Eclipse Mirror

Canada

Friends of Eclipse Mirror **Become a Friend!** **Friends login**

4. That will prompt you to download a file called *eclipse-jee-ganymede-SR2-win32.zip*. Save it to your desktop.

5. Inside the zip file is a folder called *eclipse*. Drag that folder and put it directly onto your C: drive---_C:\eclipse_ .

6. That's it. Now you can run Eclipse by going into the *eclipse* folder and running eclipse.exe.

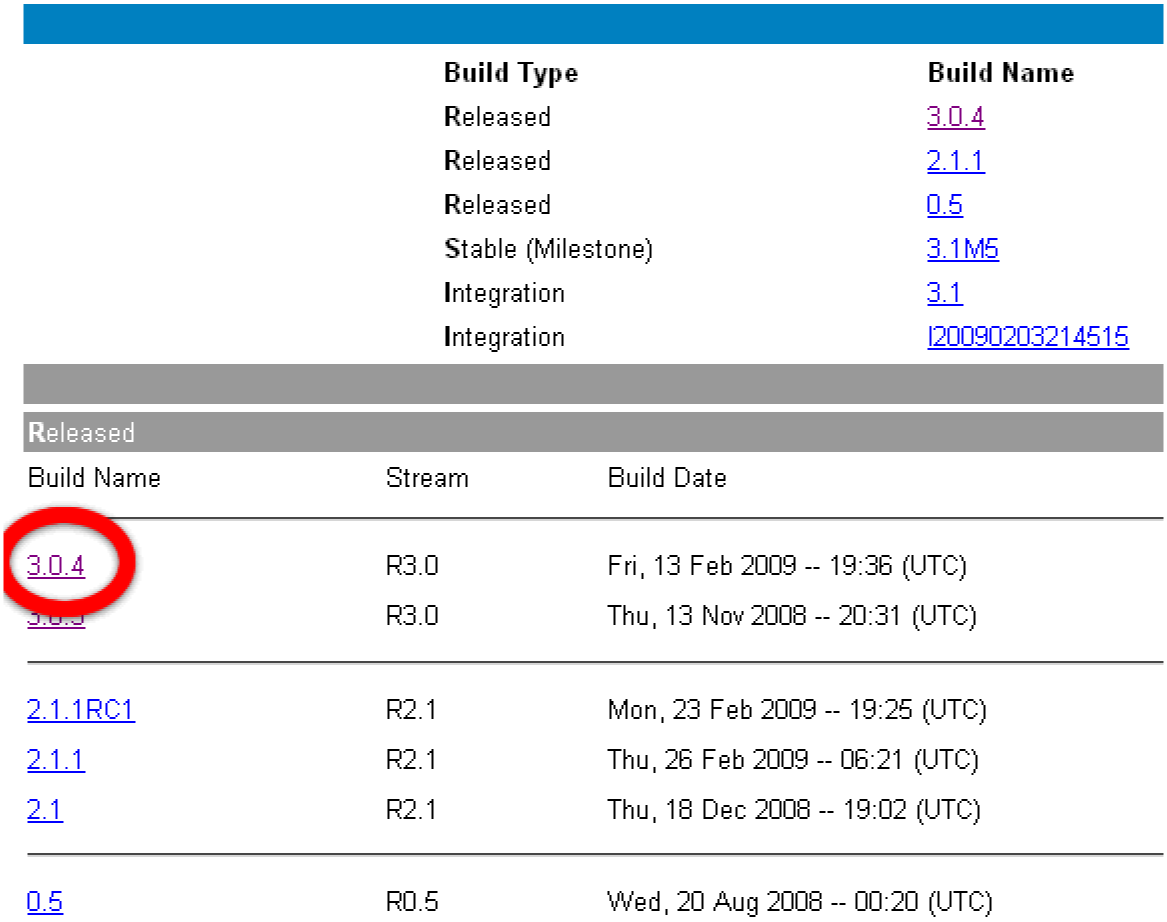
Download WTP (optional)

WTP is an Eclipse plugin that provides nice web editing tools, including a good HTML editor. It's not necessary, it's nice to have. If you don't want WTP, skip this part. You can always come back and do it later.

1. Go to http://download.eclipse.org/webtools/downloads/ .

2. Click the *3.0.4* link.

.

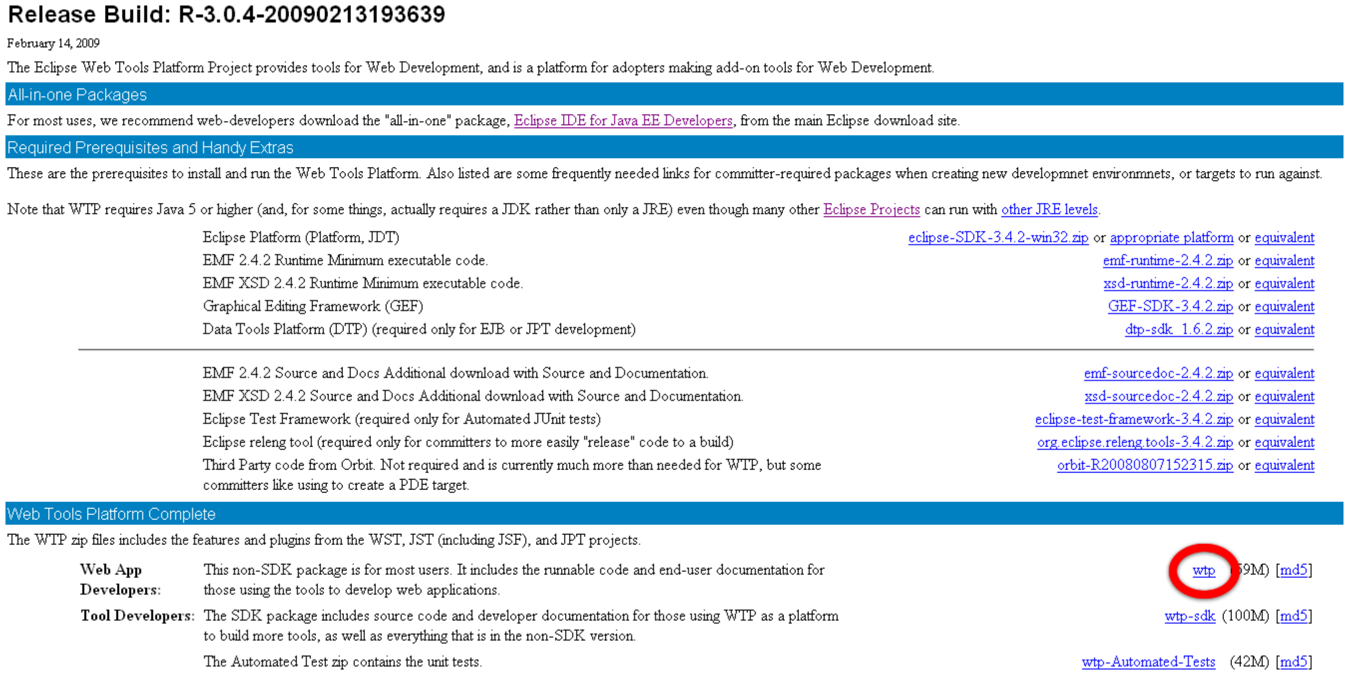


Build Type	Build Name
Released	3.0.4
Released	2.1.1
Released	0.5
Stable (Milestone)	3.1M5
Integration	3.1
Integration	I20090203214515

Released

Build Name	Stream	Build Date
3.0.4	R3.0	Fri, 13 Feb 2009 -- 19:36 (UTC)
3.0.3	R3.0	Thu, 13 Nov 2008 -- 20:31 (UTC)
2.1.1RC1	R2.1	Mon, 23 Feb 2009 -- 19:25 (UTC)
2.1.1	R2.1	Thu, 26 Feb 2009 -- 06:21 (UTC)
2.1	R2.1	Thu, 18 Dec 2008 -- 19:02 (UTC)
0.5	R0.5	Wed, 20 Aug 2008 -- 00:20 (UTC)

3. Then click the *wtp* link under *Web App Developers.*



Release Build: R-3.0.4-20090213193639

February 14, 2009

The Eclipse Web Tools Platform Project provides tools for Web Development, and is a platform for adopters making add-on tools for Web Development.

All-in-one Packages

For most uses, we recommend web-developers download the "all-in-one" package, Eclipse IDE for Java EE Developers, from the main Eclipse download site.

Required Prerequisites and Handy Extras

These are the prerequisites to install and run the Web Tools Platform. Also listed are some frequently needed links for committer-required packages when creating new developmnet environmnets, or targets to run against.

Note that WTP requires Java 5 or higher (and, for some things, actually requires a JDK rather than only a JRE) even though many other Eclipse Projects can run with other JRE levels.

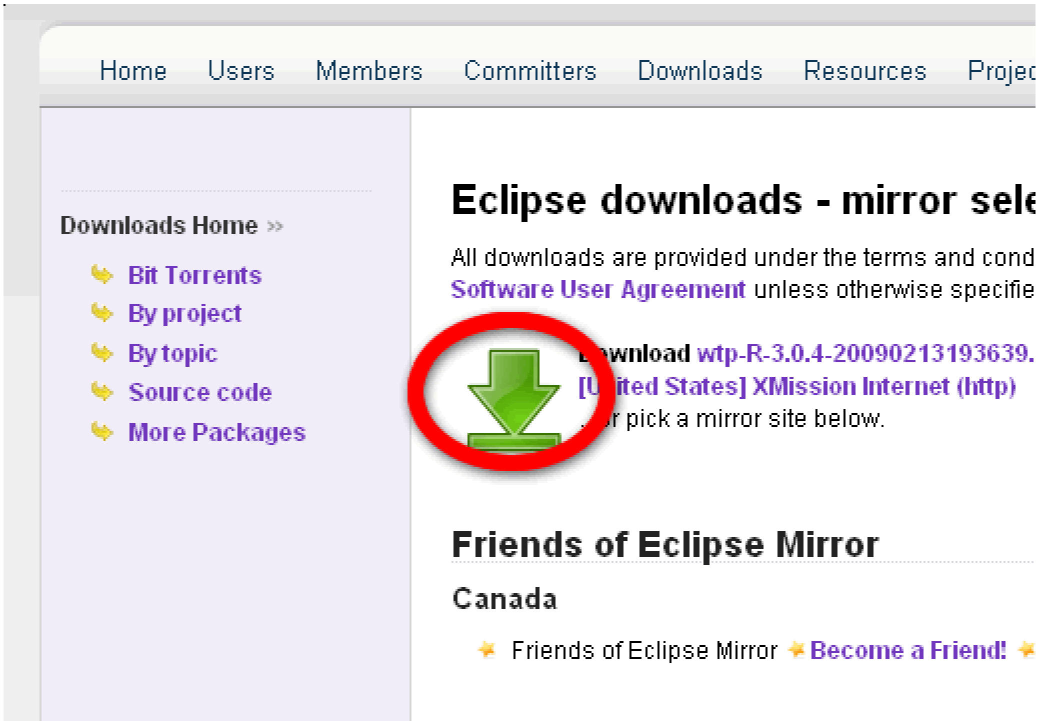
Eclipse Platform (Platform, JDT)	eclipse-SDK-3.4.2-win32.zip or appropriate platform or equivalent
EMF 2.4.2 Runtime Minimum executable code.	emf-runtime-2.4.2.zip or equivalent
EMF XSD 2.4.2 Runtime Minimum executable code.	xsd-runtime-2.4.2.zip or equivalent
Graphical Editing Framework (GEF)	GEF-SDK-3.4.2.zip or equivalent
Data Tools Platform (DTP) (required only for EJB or JPT development)	dtp-sdk_1.6.2.zip or equivalent
EMF 2.4.2 Source and Docs Additional download with Source and Documentation.	emf-sourcedoc-2.4.2.zip or equivalent
EMF XSD 2.4.2 Source and Docs Additional download with Source and Documentation.	xsd-sourcedoc-2.4.2.zip or equivalent
Eclipse Test Framework (required only for Automated JUnit tests)	eclipse-test-framework-3.4.2.zip or equivalent
Eclipse releng tool (required only for committers to more easily "release" code to a build)	org.eclipse.releng.tools-3.4.2.zip or equivalent
Third Party code from Orbit. Not required and is currently much more than needed for WTP, but some committers like using to create a PDE target.	orbit-R20080807152315.zip or equivalent

Web Tools Platform Complete

The WTP zip files includes the features and plugins from the WST, JST (including JSF), and JPT projects.

Web App Developers:	This non-SDK package is for most users. It includes the runnable code and end-user documentation for those using the tools to develop web applications.	wtp 59M) [md5]
Tool Developers:	The SDK package includes source code and developer documentation for those using WTP as a platform to build more tools, as well as everything that is in the non-SDK version.	wtp-sdk (100M) [md5]
	The Automated Test zip contains the unit tests.	wtp-Automated-Tests (42M) [md5]

4. Then click the big green arrow.



Eclipse downloads - mirror sele

All downloads are provided under the terms and cond Software User Agreement unless otherwise specifie

Download wtp-R-3.0.4-20090213193639. [United States] XMission Internet (http) or pick a mirror site below.

Friends of Eclipse Mirror

Canada

Friends of Eclipse Mirror Become a Friend!

5. That will prompt you to download a file called *wtp-R-3.0.4-20090213193639.zip*. Save it to your desktop.

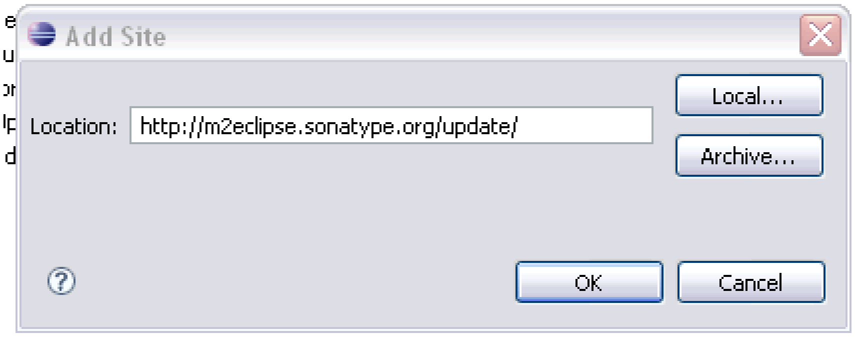
6. Inside the zip file is a folder called *eclipse*. Drag that folder and put it directly onto your C: drive. A popup window will tell you that there's already a folder called *eclipse* there, and ask if you want to continue. Say yes. That will copy the contents of this *eclipse* folder into the *eclipse* folder that's already on your C: drive.

7. That's it. Now you will have access to the WTP tools when using Eclipse.

Install M2Eclipse

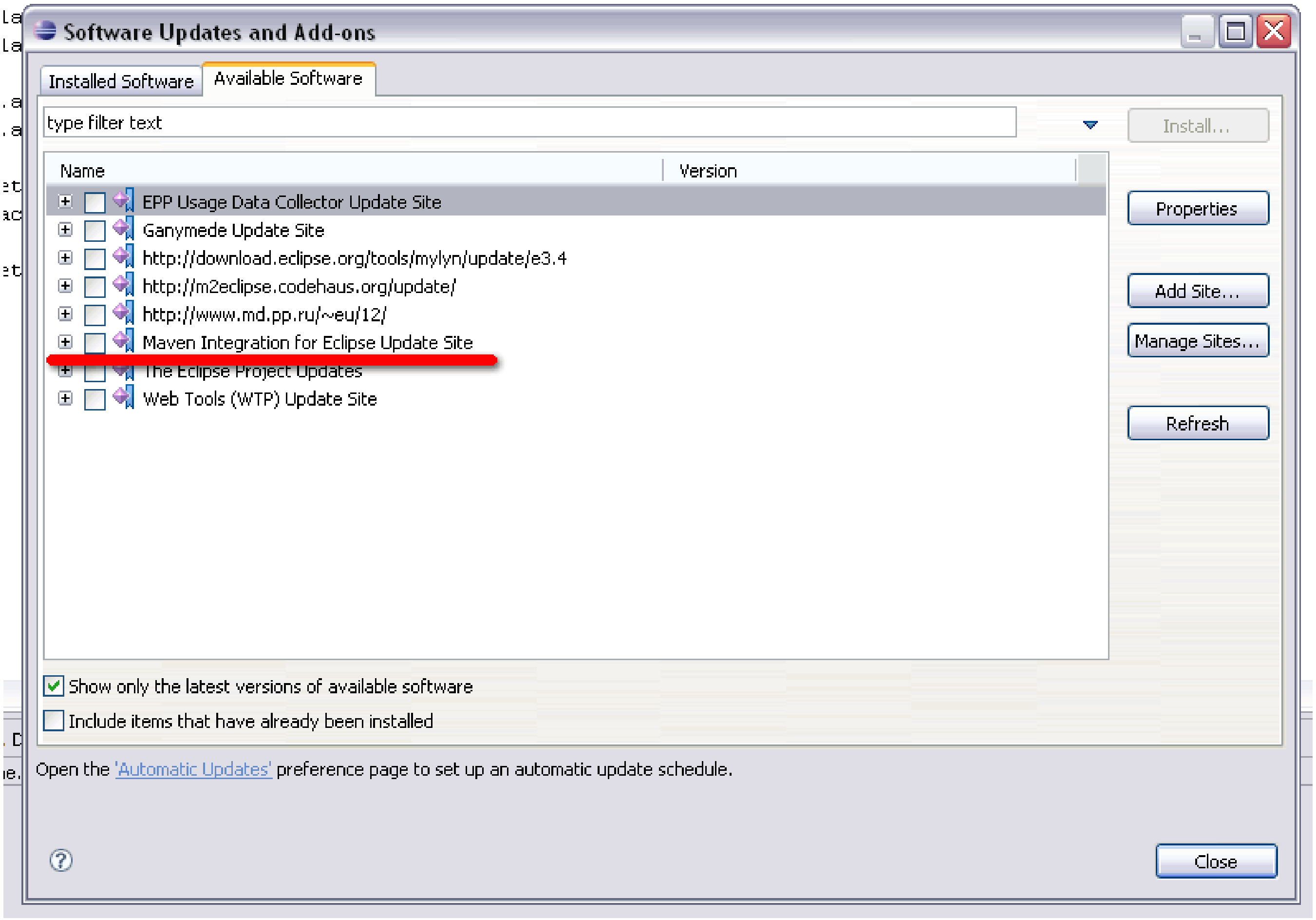
M2Eclipse is an Eclipse plugin that allows you to run your Maven project from within Eclipse.

1. Open Eclipse.
2. Go to the *Help* menu and select *Software Updates...*
3. A pop-up window will open. Make sure the *Available Software* tab is selected at the top of the pop-up window.
4. Click the "Add Site..." button on the right. That will pop up another window.
5. Paste http://m2eclipse.sonatype.org/sites/m2e into the Location bar. Then click OK.



Location: http://m2eclipse.sonatype.org/update/

6. This will add a line to the Available Software list titled *Maven Integration for Eclipse Update Site*



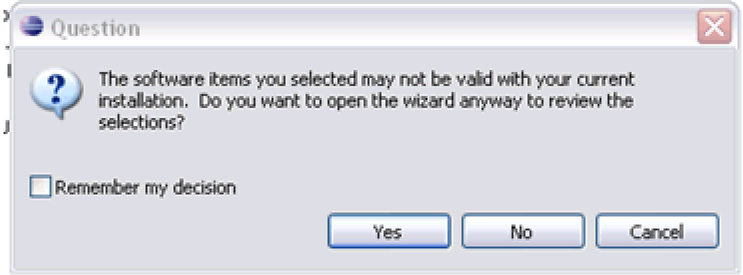
Name	Version
□ EPP Usage Data Collector Update Site	
□ Ganymede Update Site	
□ http://download.eclipse.org/tools/mylyn/update/e3.4	
□ http://m2eclipse.codehaus.org/update/	
□ http://www.md.pp.ru/~eu/12/	
□ Maven Integration for Eclipse Update Site	
□ The Eclipse Project Updates	
□ Web Tools (WTP) Update Site	

☑ Show only the latest versions of available software

□ Include items that have already been installed

Open the 'Automatic Updates' preference page to set up an automatic update schedule.

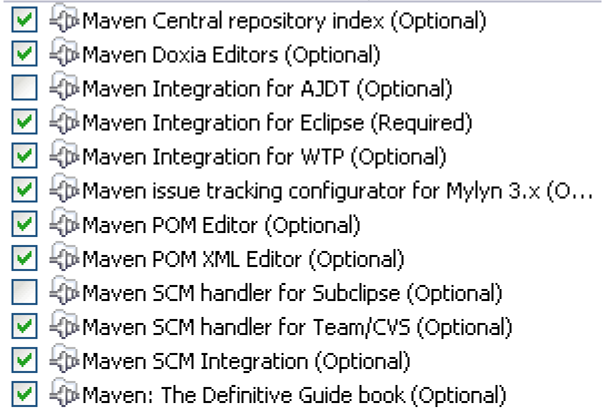
7. Check the checkbox next to *Maven Integration for Eclipse Update Site* and click "Install...".
8. A window will pop-up saying that the items you selected may not be valid yadda yadda yadda. Click "Yes".



The software items you selected may not be valid with your current installation. Do you want to open the wizard anyway to review the selections?

□ Remember my decision

9. Uncheck the boxes next to *Maven Integration for AJDT* and *Maven SCM handler for Subclipse*.



☑ Maven Central repository index (Optional)
☑ Maven Doxia Editors (Optional)
☐ Maven Integration for AJDT (Optional)
☑ Maven Integration for Eclipse (Required)
☑ Maven Integration for WTP (Optional)
☑ Maven issue tracking configurator for Mylyn 3.x (O...
☑ Maven POM Editor (Optional)
☑ Maven POM XML Editor (Optional)
☐ Maven SCM handler for Subclipse (Optional)
☑ Maven SCM handler for Team/CVS (Optional)
☑ Maven SCM Integration (Optional)
☑ Maven: The Definitive Guide book (Optional)

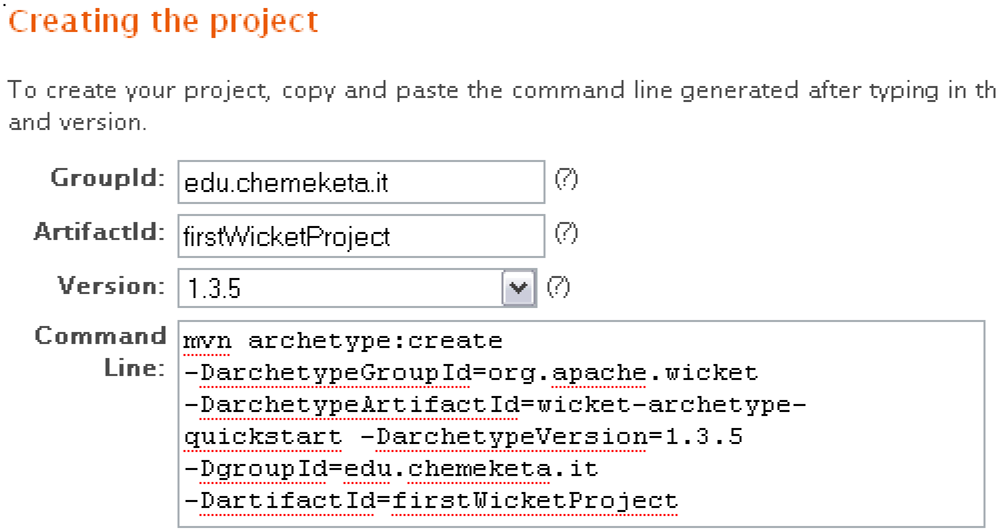
10. Click "Finish". That's it. Now you've got everything ready to build your first Wicket application.

Build Wicket Quickstart

We're almost there. Everything you've done so far will never need to be done again. This section contains all the directions to start a new project. Whenever you want to start a new project, you can come back to these steps.

The Wicket Quickstart contains all of the files and libraries necessary to make a Wicket project. It lives in the Maven repository, so we'll use Maven to download it and build it, and then we'll import it into Eclipse for you to play with.

1. Create a folder called on *wicket* on your C: drive to contain your Wicket projects---*C:\wicket*.
2. Now open a browser and go to http://wicket.apache.org/quickstart.html .
3. Under the "Creating the project" section of the page, enter a GroupId and an ArtifactId. The GroupId can be any string that reasonably identifies your organization. The ArtifactId will be the name of your project. Leave Version at 1.3.5. (You're welcome to try 1.4, but I haven't had much luck with that.)



Creating the project

To create your project, copy and paste the command line generated after typing in the groupId, artifactId and version.

GroupId: edu.chemeketa.it

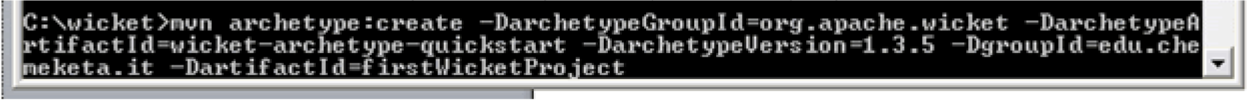
ArtifactId: firstWicketProject

Version: 1.3.5

Command Line:

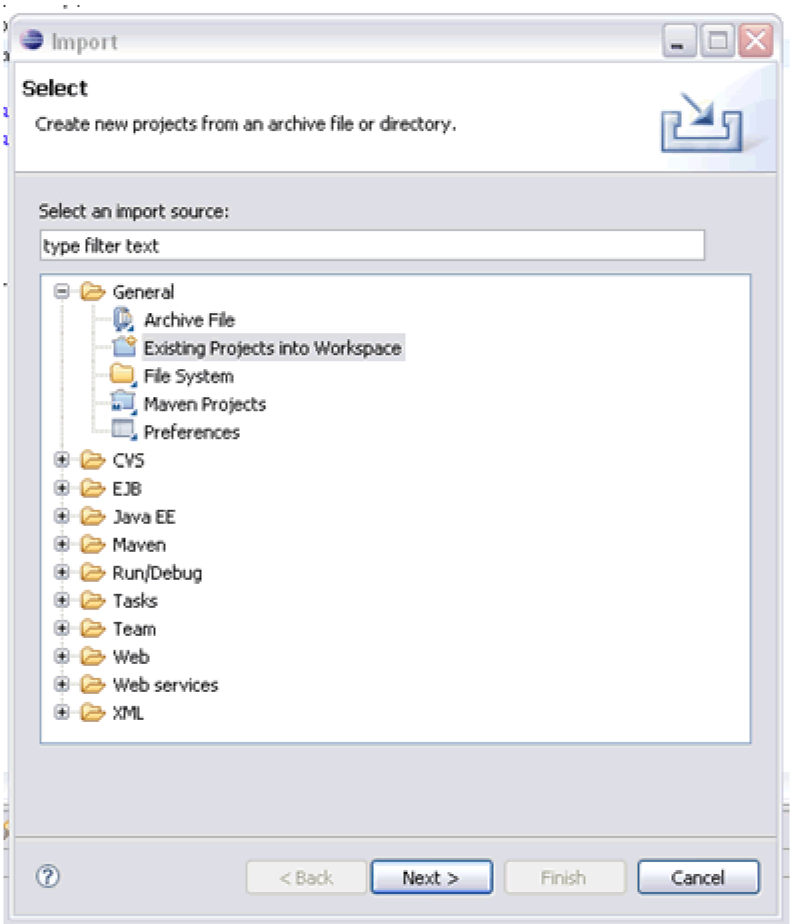
```
mvn archetype:create
-DarchetypeGroupId=org.apache.wicket
-DarchetypeArtifactId=wicket-archetype-
quickstart -DarchetypeVersion=1.3.5
-DgroupId=edu.chemeketa.it
-DartifactId=firstWicketProject
```

4. Now open a command prompt (Start -> Run... -> cmd) and navigate to _C:\wicket_.
5. Copy the text inside the "Command Line" textbox from step 3 above and paste it into your command prompt. Hit enter. Then wait as Maven downloads a bunch of stuff.



```
C:\wicket>mvn archetype:create -DarchetypeGroupId=org.apache.wicket -DarchetypeA
rtifactId=wicket-archetype-quickstart -DarchetypeVersion=1.3.5 -DgroupId=edu.che
meketa.it -DartifactId=firstWicketProject
```

6. When that finishes, you will see that a new folder called *firstWicketProject* has been created in your *C:\wicket_ folder. From your command prompt, navigate into _firstWicketProject*.
7. Your command prompt should now be in *C:\wicket\firstWicketProject*. From here, type
mvn eclipse:eclipse
and hit Enter. Wait while Maven download more stuff.
8. Now open Eclipse, if you don't already have it open. Go to the *File* menue and select *Import...* A window will pop up with several folders.
9. Expand the top folder, named "General". Select the "Existing Projects into Workspace" option and click "Next".

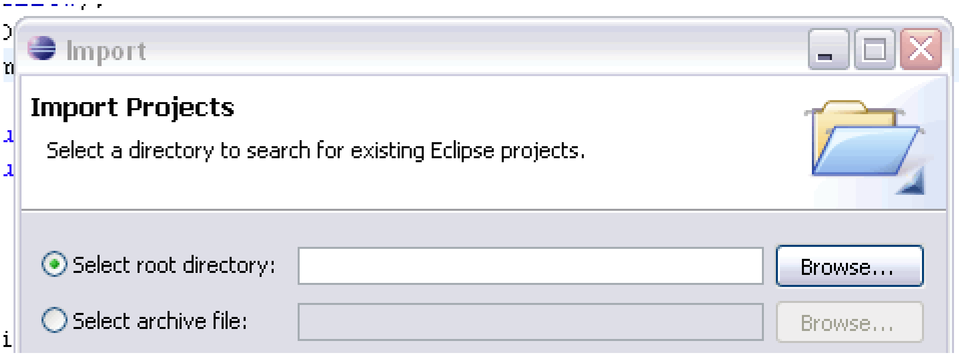

Select

Create new projects from an archive file or directory.

Select an import source:

type filter text

10. Click the "Browse..." button to the right of the "Select root directory" prompt.


Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

11. Navigate to *C:\wicket\firstWicketProject* and click "OK".
12. The Projects area of the window will now show "firstWicketProject" with a checkbox checked next to it. Click Finish.

Running the Application

Congratulations! You've installed the necessary software and built your first Wicket project. Now lets look at the project files and get it running.

1. In the Package Explorer on the left hand side of the screen, you'll now see a folder called "firstWicketProject". If you expand that folder you'll lots of stuff. The important items are *src/main/java*, *src/test/java*, and *src*.
2. *src/main/java* contains the project Java and HTML files. This is where you'll add new pages.
3. *src/test/java* contains the embedded server. We'll come back here in a minute.
4. *src* contains your web.xml file. You'll find it under *src/main/webapp/WEB-INF*. Use that to set up your project configuration.
5. Now go back to *src/test/java*. Inside you'll see a package that has the same name as the GroupId you entered on the Quickstart website.
6. Expand the package inside *src/test/java* and you'll see a Java file called Start.java. Right-click Start.java and go down to "Debug as..." and select "Java Application". You'll see the server starting up inside the console at the bottom of the screen.
7. To test to see if everything is working, open a browser and go to http://localhost:8080 . You should see a message telling you and Wicket is running.
8. To stop the server, click the red square at the top right of the console in Eclipse. You may need to stop and start the server after you make programming changes in order for the changes to take effect.

Bonus Section

That's just about everything. Now you're ready to go into your *src/main/java* and start building your application. However, there are two more changes I suggest you make.

1. Open the pom.xml file in the project root directory and scroll toward the bottom, where you'll find:

   ```
   <plugin>
           <groupId>org.apache.maven.plugins</groupId>
           <artifactId>maven-eclipse-plugin</artifactId>
           <configuration>
                   <downloadSources>true</downloadSources>
           </configuration>
   </plugin>
   ```

   and add the line

   ```
   <version>2.5.1</version>
   ```

   right after

   ```
   <artifactId>maven-eclipse-plugin</artifactId>
   ```

2. Go back to *src/test/java* and open the Start.java file into the Eclipse editor.
3. Delete the entire contents of the file and paste in the following instead. I find this change helps the embedded server to pick up your code changes without needing to be restarted.

```java
package edu.chemeketa;

import java.lang.management.ManagementFactory;

import javax.management.MBeanServer;

import org.mortbay.jetty.Server;
import org.mortbay.jetty.nio.SelectChannelConnector;
import org.mortbay.jetty.webapp.WebAppContext;
import org.mortbay.management.MBeanContainer;

/**
 * Seperate startup class for people that want to run the examples
 * directly.
 */
public class Start {

  /**
   * Main function, starts the jetty server.
   *
   * @param args
   */
  public static void main(String[] args) throws Exception {

    Server server = new Server();
    SelectChannelConnector connector = new SelectChannelConnector();
    connector.setPort(8080);
    server.addConnector(connector);

    WebAppContext web = new WebAppContext();
    web.setContextPath("/");
    web.setWar("src/main/webapp");
    server.addHandler(web);

    MBeanServer mBeanServer = ManagementFactory
        .getPlatformMBeanServer();
    MBeanContainer mBeanContainer = new MBeanContainer(mBeanServer);
    server.getContainer().addEventListener(mBeanContainer);
    mBeanContainer.start();

    try {
      System.out.println(">>> STARTING EMBEDDED JETTY SERVER, PRESS ANY KEY TO STOP");
      server.start();
      while (System.in.available() == 0) {
        Thread.sleep(5000);
      }
      server.stop();
      server.join();
    } catch (Exception e) {
      e.printStackTrace();
      System.exit(100);
    }
  }
}
```