HowToSetupTapestry41EclipseWtp

- Purpose of this document
- Benefits to using the Eclipse Web Tools Platform for Tapestry development
- Tested Configuration
  - Prerequisites
- Steps
  - 1. Launch Eclipse
  - 2. Define the JDK
  - 3. Associate Tapestry file extentions with the XML editor
  - 4. Define XML templates
  - 5. Define Tapestry XML DTD Location
  - 6. Define a Server Runtime
  - 7. Define a Server Instance
  - 8. Disable Caching of Tapestry files
  - 9. Create a Dynamic Web Project
  - 10. Add Tapestry libraries to the project
  - 11. Setup a Skeleton Site
  - 12. Attach Project to Server Instance
- Verification
  - View the site
  - Verify that templates and specifications are not cached
  - Verify that context restarts automatically after modifying a Class file
- General troubleshooting strategies
  - Do a clean build
  - Relaunch Eclipse
  - Inspect the contents of the temporary deployment directory
- Solutions to specific problems
  - A change in a specification file does not appear in the browser
  - Some times, files are never deployed any more
  - HTTP Status 404

Purpose of this document

Once setup correctly WTP is easy to use. However it can be a real pain to setup a new Tapestry project in WTP. This document exists to so that new users can setup WTP without hours of troubleshooting and experienced users don't forget anything.

This How-To was copied from HowToSetupEclipseWtp but updated for current software. (It still needs to be cleaned up some, particularly at the end)

Benefits to using the Eclipse Web Tools Platform for Tapestry development

- No Ant scripts needed to build and deploy
- Start, stop & restart the web server within Eclipse
- Automatic, on-the-fly deployment of all classes, libraries and resources
- Server context reloads automatically when Java source files change (often faster than a full server restart)
- Tomcat installation is not modified
- Content assist and validation in XML editors of elements and attribute names (i.e. HTML templates, application, page & component specifications).
- Can define Templates for common Tapestry files
- Debugger *just works* for deployed code

Note: for some reason content assist does not work with the application specification. It is validated however.

Note: projects are deployed as part of the build process. You will have to manually initiate a build if you prefer to have ***Project > Build Automatically*** *unchecked.*

Tested Configuration

- Tapestry 4.1
- java-7-icedtea
- Ant 1.7.0
- Tomcat 6.0.16
- Ubuntu 8.04
- Eclipse 3.2.2
- Web Tools Platform 2.0.2

and

- Tapestry 4.1.5
- Java 1.5_013 (MacOSX 10.5.2)
- Tomcat 6.0.16
- Apple MacOSX 10.5.2
- Eclipse 3.2.2

- Web Tools Platform 2.0.2

Prerequisites

- Install JDK 1.5.0 or higher http://java.sun.com/j2se/1.5.0/download.jsp
- Install Ant
- Install Tomcat 6 http://tomcat.apache.org/download-60.cgi
- Install and/or build Tapestry 4.1. See BuildingTapestry
- Install Eclipse and Web Tools Platform. The easiest way is to grab the All-in-one download. This contains Eclipse, WTP, EMF, GEF & JEM, all of which you need to run WTP. http://download.eclipse.org/webtools/downloads/drops/R-1.0-200512210855/

Steps

1. Launch Eclipse

- Launch Eclipse
- Close the **Welcome** tab (you can get to it later by **Help > Welcome**)

2. Define the JDK

Eclipse will launch with your machine's default JRE.

- **Window** > **Preferences...**
- Click **Java** > **Installed JREs**
- Add JDK 1.5, or JDK 1.6 if necessary (**not** a JRE)

3. Associate Tapestry file extentions with the XML editor

- In **Preferences** go to **General** > **Content Types**
- Add `.application`, `.page`, `.jwc` and `.library` extensions to the **Text** > **XML** node

ContentTypes.gif!

4. Define XML templates

- Download templates.xml
- In **Preferences** go to **Web and XML > XML Files > Templates**
- Click **Import** and select `templates.xml`

? Unknown Attachment

- Click **OK** to close the **Preferences** dialog

Note: these templates are configured to be available in empty files. You can make them available in any XML file by editing the template and changing ***Con text*** *to* `All XML`*.*

5. Define Tapestry XML DTD Location

Registering the Tapestry XML DTD allows <ctrl>+<space> auto-completion when editing tapestry files

- In **Preferences** go to **Web and XML > XML Catalog**
  Highlight **User Specified Entries**
  Click **Add...**
  Click on the folder icon to the right of the **Location** field.
  Choose **File System...**
  Browse to and select the tapestry jar file
  Now, you need to manually edit the **Location** field as follows.
- 5.1. You will have a full path to the tapestry jar, preceeded by `file:///`. Change this to `jar:file:`
- 5.2. Now, move the carat (thing showing you where you are typing) to the right of the field.
  You should see something like **tapestry-framework-4.1.5.jar**.
- 5.3. Directly after **tapestry-framework-4.1.5.jar**, type `!/org/apache/tapestry/parse/Tapestry_4_1.dtd`.
  Note that it is case sensitive. N.B. For reference, this step just tells eclipse where the dtd is
  located within the jar file.
- 5.4. Ensure that the **Key Type** field is set to **Public ID**
- 5.5. Set the **Key** field. The value in here comes from somewhere near the top of the dtd file we referenced
  in step 3, and in my case, I found this to be
  `-//Apache Software Foundation//Tapestry Specification 4.1//EN`
  N.B. This is all case sensitive too
- 5.6. Click **OK**

- 5.7. Now, look at the new entry that has appeared under **User Specified Entries**. If it has a red cross next to it, then it means that the DTD could not be found at that location, so the thing to do is to open the jar file with winzip (or whatever zip program you normally use), and see where the DTD really is. After establishing where the DTD is, go back to step 5.3, and modify the path as appropriate.
  Note : I removed all instance of '`%20`' and replaced them with a '{{ }}' in step 5.3 because that was what I saw was in the other examples on that setting page, though I did not check if this was required.

6. Define a Server Runtime

This tells Eclipse where Tomcat is installed.

- In **Preferences** go to **Server > Installed Runtime**
- Click **Add...**
- Select **Apache** > **Apache Tomcat v6.0** and click **Next**
- Browse to and select your Tomcat root directory e.g. *C:\Program Files\Apache Software Foundation\Tomcat 5.0* or */opt/java/tomcat_60*
- Select JDK 1.5

TomcatServerRuntime.gif!

- Click **Finish**
- Click **OK** to close the Preferences dialog

7. Define a Server Instance

- **Window** > **Show View** > **Other**
- Open the **Server** > **Servers** view
- Right click in the **Servers** view and select **New > Server...**

NewServer.gif!

- Select **Apache > Tomcat v6.0 Server**
- Select the server runtime you created previously

NewServer2.gif!

- And click **Finish**
- You should now have a server in the **Servers** view

*A side-effect of creating a Server Configuration is a new **Servers** project will be created.*

ServersProject.gif!

8. Disable Caching of Tapestry files

WTP does not automatically reload the context when page specifications and HTML templates are changed. If you set the Tapestry flag to disable caching you don't need to restart the context to see changes. See http://jakarta.apache.org/tapestry/UsersGuide/configuration.html#N10106

- Double-click the server instance you created previously to open it in the editor view.
- **General > Open launch configuration**
- Select the **Arguments** tab
- Add `-Dorg.apache.tapestry.disable-caching=true` to the **VM arguments**
- Add `-enableassertions` if you use assertions

ServerArguments.gif!

9. Create a Dynamic Web Project

The Dynamic Web Project is a part of WTP. It adds automatic deployment support.

- **File > New > Project**
- **Web > Dynamic Web Project**

NewDynamicWebProject.gif!

- Use defaults for rest of wizard and click *Finish*
- When the project is created you will be asked whether you want to open the *J2EE* perspective. Click **No**.

*Note: You can open the **J2EE** perspective later via **Window > Open Perspective > Other...***

10. Add Tapestry libraries to the project

Note: I was unable to employ the user library* approach for adding Tapestry libraries to the project. Tapestry was not loaded at runtime*

- Go to the `lib` directory of your Tapestry installation (e.g. `D:\tools\tapestry-4.0\ext-package\lib`)

- Copy all the jars into `WebContent/WEB-INF/lib`. You can simply drag the jars to the `lib` folder in Eclipse or do a file system copy (do a **File > Refresh** if necessary).

WebAppLibraries.gif!

WTP automatically adds the libraries to the build path and publishes them for Tomcat to use.

Note by cyrille37: There is a way to add libraries at runtime without add them as real files in the web-inf/lib folder ; Have a look at Eclipse's WTP documentation: Adding Web library projects

11. Setup a Skeleton Site

Creating Tapestry specification files is easy using the XML templates we defined previously.

- Create an empty file (e.g. `Home.page`)
- Click in the editor, switch to source if it is on the design tab, and Press **Ctrl-Space** on the mac. Original instructions have **Alt-Space** (on a german Keyboard, it ist **Strg-Space**) I assume that is correct for Microsoft Windows.
- And select the template you want (e.g. `page spec`).

ContentAssistPageSpec.gif!

Below are the files for the *Hello World* application used in the rest of this How-To. You can download these or re-create them from scratch.

At the moment these are just stubs - the absolute minimum that runs, I have not added enough to make the examples at the bottom work.

- Put these files in the `WebContent/WEB-INF/` directory

web.xml

TapestryTest.application

Home.html

Home.page

- Put this Java file in the `src` directory

Home.java

- Select your project and **File > Refresh**

12. Attach Project to Server Instance

- Select the server instance you defined in the **Servers** view
- Right-click and select **Add and Remove Projects...**
- Select your project in the **Available Projects** field
- Click **Add >**
- Click **Finish**

AddRemoveProjectsDialog.gif!

Your project should appear under the server in the **Servers** view. It will be published when the server starts.

Verification

View the site

- Start the server

StartServer.gif!

- browse to the context path you defined (e.g. http://localhost:8080/TapestryTest/) and you should see something like this:

HelloWorld.gif!

Verify that templates and specifications are not cached

- Open **Home.html** and change the title from `HelloWorld` to `HelloMoon`
- Open **Home.page** and change the value of `insertHeading` from `literal:Hello World` to `literal:Hello Moon`
- **File > Save All**
- Reload the page in your browser *(do not restart the server)*

The title and heading should now say `Hello Moon`. If this is not the case then there is a problem with the `org.apache.tapestry.disable-caching` VM argument.

Verify that context restarts automatically after modifying a Class file

- Open `Home.java`
- Change `return "Hello World";` to `return "Hello Moon";`
- **File > Save All**
- If you watch the **Server** view you should see it reload the context after a moment.
- Reload the page in your browser

The text should now say `Hello Moon`.

General troubleshooting strategies

Some things you can try if your project is not running as expected

Do a clean build

- Stop the server

StopServer.gif!

- **Project > Clean...**

Relaunch Eclipse

- **File > Switch Workspace**
- Press **OK**

Eclipse will restart and reload your current workspace

Inspect the contents of the temporary deployment directory

During a build, WTP copies all deployable resources to a temporary directory. Tomcat loads the site from here. Sometimes it may become unsynchronized (particularly if you modify a jar while the server is running). Doing a clean build and restarting the workspace should take care of any synchronizing problems you have. But if you're still having problems you can examine the deployment directory.

- If you look in the console you should see a message like this" `INFO: Installing web application at context path from URL file:D:\TapestryDemo\workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\webapps\ROOT`.
- Go to the file system and ensure all the needed files are present

Solutions to specific problems

A change in a specification file does not appear in the browser

Make a small change to the specification file and resave it. Reload the page in the browser. In his book *Enjoying Web Development with Tapestry* Kent Tong reports the cause of this probem as Windows caching the file.

Some times, files are never deployed any more

Please, correct my poor english (Cyrille37)

Some times I add to erase WST.Server temporary deployement files because the "Run as Server" never copy files like home.html and more...

Stop Eclipse then

Erase file :
D:\evote.java\essais.withServer\.metadata\.plugins\org.eclipse.wst.server.core\publish.xml

and folders :
D:\evote.java\essais.withServer\.metadata\.plugins\org.eclipse.wst.server.core\publish D:\evote.java\essais.withServer\.metadata\.plugins\org.eclipse.wst.server.core\temp0

Then restart Eclipse.

HTTP Status 404

This can happen when the Tapestry libraries are missing