Building WOLips

Not Required!

You do not have to build WOLips to use it. It is easily available via Eclipse's Software Update functionality. The update site is: http://wocommunity.org/wolips/3.7/stable/

The following instructions about building WOLips are fine if you need to build a distribution of WOLips. However, they aren't so great if you're debugging WOLips. Instructions on how to do that are posted here Debugging WOLips.

Prior to building WOLips, you will want to get your dependencies installed properly. Check out the Installing WOLips page for information on the dependencies for WOLips.

Prerequisites

Install eGit or another Git client

See Getting Started With Git for information on how to obtain a Git client.

Install the m2eclipse plugin (Eclipse 3.5 instructions)

1. Help -> Install New Software
2. Click "Add..."
    1. Name = M2Eclipse
    2. Location = http://m2eclipse.sonatype.org/update/
    3. Click "OK"
3. Expand "Maven Integration"
4. Select "Maven Embedder"
5. Select "Maven Integration for Eclipse"
6. Click "Next>"
7. Click "Finish"
   (*What is the m2eclipse plugin a prerequisite for? Nothing below refers to it....*)

Checking out and Building from the Commandline

1. Checkout the `trunk/woproject` folder from Subversion
   `git clone git://github.com/wolips/wolips.git` woproject
2. `cd woproject`
3. `ant`
4. The first build will fail out telling you that it couldn't find `wobuild.properties` – That's OK
5. Edit `~/Library/wobuild.properties`
6. Add a line: `eclipse.home=/path/to/your/eclipse`
7. Call ant with the build.version argument:

    ```
    ant -Dbuild.version=x.y.z
    ```

    1. If you get the error `'build.xml:73: taskdef A class needed by class org.apache.velocity.anakia.AnakiaTask cannot be found: org/jdom/JDOMException'`, you can do one of two things. You can build with `-Dno.docs=true` to skip the javadoc step, or (untested) you can drop `jdom-b8.jar` from `woproject/projects/lib` into your `ANT_HOME/lib` folder.
8. The build went into `temp/dist`
9. Profit

(Successfully built on Ubuntu 14.04 LTS with eclipse 4.4.1, using "ant -Dbuild.version=4.4.1" as the build command. - 2015/03/12 - ray@ganymede.org with an OMG thanks to stefan.klein@buero-sde.de.)

Checking out and Building inside of Eclipse for Deployment

1. Add the wolips Git repository https://github.com/wolips/wolips/ in the Git Perspective's Repository View

2. Right-click (or equiv) on the `build.xml` inside of the "woproject" project and Run As=>Ant Build ...
3. Switch to the Properties tab and check for the definition of eclipse.home in the Properties list
4. If `eclipse.home` does not exist, uncheck "Use global properties" and then click "Add Property"
5. In the Add Property dialog, make the name `eclipse.home`, click Add Variable, select `eclipse_home` from the list, and click OK
6. Click Apply to save your changes
7. Run
8. Once it is complete, you may need to right-click=>refresh your project and you will have a `temp/dist` folder with the plugin site in it.

Checking out and Building inside of Eclipse for Development (PDE)

1. Add the wolips Git repository https://github.com/wolips/wolips/ in the Git Perspective's Repository View
2. You can select individual plugins you would like to work on by browsing to `trunk/woproject/wolips/` in the Subversion repository and checking out the plugins you are interested in. To checkout complete sets of plugins, you could also use the Eclipse Team Project Set.

An approach to setting up the development and test environment for WOLips Plugin.

In order to do WOLips Plugin development you will need two eclipse environments: Development and Test.

1. Follow the Checking out and Building inside of Eclipse for Deployment in you primary Eclipse installation (i.e. /Application/eclipse) and workspace. This will be you development environment.
2. the `temp/dist` folder is your Local Site for the Test Environment.
3. Create a second copy of "clean" Eclipse under /Users/<me>/Developer/eclipse (this could be any where). This will be your Test Environment.
4. In the test environment, follow the Installing WOLips instructions, using your `temp/dist` directory as the Local Site. Install the plugins.
5. Use the Test environment to exercise/test the changes from your locally modified Plugin.

1. When you need to repeat the process, shut down the Test Environment, delete the org.objectstyle.* files in the features and plugins directories under the /Users/<me>/Developer/eclipse directory. Change the source code in the primary environment and build. (Each build took 3 min 37+ seconds on my Mac Book Pro 17"). Once the build is done, then reinstall the plugin in the Test Environment. The Local Site should still be selected.

Building WOLips with Jenkins Continuous Build server

Hudson is a Java base continuous build server that allows you to build and test your project automatically. It can be easily configured to build WOLips in a few simple steps.

1. Install Eclipse 3.7. You need it to be able to build WOLips.
2. Copy the necessary plugins from your machine to the Eclipse installation on the server. In this example it has been copied to /usr/local/eclipse
   The currently required plugins for a deployment build are
   * JProfiler (Update Site: http://wocommunity.org/documents/tools/jprofiler6/)
   * JRebel (Update Site : http://www.zeroturnaround.com/update-site/)
3. Create a new job in Jenkins to build WOLips. You can use the following sample config or you can add a new job with the following configuration:
    1. Project Name: WOLips
    2. Git Repositories: git://github.com/wolips/wolips.git
    3. Use update: Checked
    4. Poll SCM: 59 * * * * *(59 can be replaced with any number between 0 and 59)*
4. Add the following build tasks to the job

Execute Shell

```
echo build.version=3.7.${BUILD_NUMBER} > build.properties
echo eclipse.home=/usr/local/eclipse >> build.properties
echo user.home=${WORKSPACE} >> build.properties
```

Invoke Ant

```
Targets: dist
```

If you want to deploy the build to a web server to serve as an update site add a task that is something like this

Execute Shell

```
rm -fr  /usr/local/www/apache22/data/wolips/*
/bin/cp -prv ${WORKSPACE}/wolips/temp/dist/
/usr/local/www/apache22/data/wolips/
```

Preparation for building WOLips on Windows operating systems (NT 5.0 and higher)

For building WOLips on Windows operating systems, some minor changes have to be made. Instead of following and solving the build errors, you may follow the steps described below.

1. Install the BuildingWOLips-Prerequisites. Do also follow the BuildingWOLips-CheckingoutandBuildinginsideofEclipseforDeployment section, but don't run the build until the changes described below have been made.
2. Set the VMs encoding to MacRoman
    - Right click on `build.xml` -> Properties -> JRE -> VM arguments
        - -Dfile.encoding=MacRoman
3. Edit `woproject/wolips/goodies/plugins/build.xml`
    - Comment out the subsequent lines (mac goodies)
        - <ant dir="org.objectstyle.wolips.goodies.core.mac" target="all" />
        - <ant dir="org.objectstyle.wolips.goodies.ui.mac" target="all" />
        - <ant dir="org.objectstyle.wolips.goodies.core.mac" target="dist" />
        - <ant dir="org.objectstyle.wolips.goodies.ui.mac" target="dist" />
4. Edit `woproject/wolips/features/build.xml`
    - Comment out the subsequent lines (mac features)
        - <ant dir="org.objectstyle.wolips.goodies.mac.feature" target="all" />
        - <ant dir="org.objectstyle.wolips.goodies.mac.feature" target="dist" />
5. Edit `woproject/wolips/build.xml` (you may skip this one if you've got JProfiler installed)
    - Comment out the subsequent lines
        - <ant dir="jprofiler" target="all" />
        - <ant dir="jprofiler" target="dist" />
6. Run the build

Installing from a Local Build

1. If you made changes locally, Eclipse will not automatically install your changed plugins if the version number is the same. You have two options - either uninstall WOLips or cheat and increment the WOLips build version prior to building your local copy. To uninstall WOLips, just delete all org.objectstyle features and plug ins from your eclipse folder. To increment the build server version, open buildserver.version in your projects folder and increment the last number. Note that you will now be "competing" with version numbers from the official build server, so the next official build will not install because you already used the next number, but sometimes it's a quick hack to force an update if you're lazy 🙂
2. Assuming you now have your build versions under control, you should follow the instructions for Building WOLips
3. Launch Eclipse
4. Select the menu: Help->Software Updates.
5. Click on the 'Available Software' tab at the top of the window.
6. Click on the 'Add Site...' button. This will bring up a dialog box.
7. Click on the 'Local...' button on the dialog box.
8. Browse to and select your workspace/projects/temp/dist folder or your woproject/temp/dist folder (depending on whether you checked out in eclipse or command line). Check the box next to it.
9. If you updated the version number in your local copy (instead of uninstalling), you must now click on the 'Installed Software' tab at the top of the window and click on the 'Update...' button.
10. Then continue, and just keep selecting the "Yes"/"OK"/"Approve"/etc buttons