Contributing to Wicket

Also see Oli's entertaining presentation (including photos showing what to wear) at http://jweekend.com/dev/ArticlesPage or http://code.google.com/p/londonwicket.

Report a Bug

Wicket manages bug reports via the Apache Jira site: http://issues.apache.org/jira/browse/WICKET

It requires you to register with the site, to avoid spamming, and attribute credit where its due. It's relatively painless. Much harder is writing a clear and concise bug report.

Etiquette for a bug report is:

1. **Always search for an existing bug report about the issue before adding your own.** This can be tricky, but several duplicate bug reports just cause an administration headache for developers. If in doubt you can always post the details of the bug to the mailing list, and wait to be asked to submit a bug report by a dev. But the jira search functionality is pretty good, so do have go.
2. **Be clear and concise in your language.** Wording a short but useful summary, with a description that contains all the steps leading up to the error plus any extra circumstantial evidence, is not always easy, but its vital to getting the issue resolved, and really valuable.
3. **Watch the bug.** The developers may ask you for more information via the issue comments, so its good practices to check in on the issue whilst its still open.
4. **Don't be shy of maintaining the issue.** If you find its a duplicate of some other issue that was reported before yours, then mark it as a duplicate, and watch the original. If you get more info on the bug, add it as a comment. If it doesnt get noticed by a dev, try submitting a quickstart or better still a patch. Ask for it to be reviewed via the mailing list.

You will be rewarded with subtle kudos and the bug is much more likely to be fixed promptly.

Build a Quickstart

See http://wicket.apache.org/start/quickstart.html for a good introduction to building a barebone Wicket project from one command. There is even a tool to let you customise the command for your project.

Or you can watch Al Maw's excellent screencast here: http://herebebeasties.com/2007-10-07/wicket-quickstart

The maven command provided there is the quickest way to get a working Wicket project that you can use to clearly demonstrate a bug.

Once you have an example that shows the minimum conditions under which the issue occurs, then you can zip up the project and attach it to the relevant issue.

When a Wicket dev asks you to submit a quickstart, that is what they mean.

That is far more useful than posting reams of your application code and telling everyone it doesn't work. The former will help diagnose the problem, the latter will mainly get you abused on the mailing list.

Submit a patch

This is the most involved, but the most rewarding route 😎.
In a nutshell this involves:

- Checking out the Wicket source code from the Subversion repository.
- Building it via mvn or IDE plugin.
- Proving the bug exists, either via a Quickstart, or writing a unit test.
- Submitting your fix as a Subversion patch file.

This guide will outline how to do it with Eclipse + Subclipse + m2eclipse.
If you prefer the command line, then start here: http://wicket.apache.org/building-from-svn.html
If you can provide a walk through for your IDE of choice then please do add it.

Setting up

Using an Eclipse 3.4.x flavour of your choice add install the following plugins

- Subclipse for accessing Subversion - Eclipse update site URL http://subclipse.tigris.org/update_1.6.x
- m2eclipse for maven support - Eclipse update site: http://m2eclipse.sonatype.org/update/

I usually install the entire Subclipse plugin, and all of m2eclipse except the Project configurators unless I am using a J2EE version of Eclipse.
I had a spurious error when adding the m2eclipse update site, "No repository found at http://www.md.pp.ru/~eu/12/" but it didn't cause any problems with the install.

Checking out Wicket from Subversion

- **Add the Wicket Subversion repository** - Navigate to Window > Show Perspective > Other... > SVN Repository Exploring (eclipse uses the term perspective to mean a collection of views, and a view is one single panel with some specific function like the navigator view or the editor view) In the SVN Repositories tab on the left, right click > New > Repository Location... and paste in the repository url: http://svn.apache.org/repos/asf/wicket

- **Checkout the code** - Click on the new Repository and navigate to: trunk > right click > checkout as maven project. Some serious churning will now occur as you download of Wicket source, and then maven crawls the internet for all the required libraries. Seriously, go for a stroll in the sunshine; the checkout and build took about 6minutes on my machine, and thats without maven needing to download anything.

Building Wicket

- **The good news**, all that waiting wasn't in vain, as you should now have a fully checked out, dependency managed, built, possibly working, Wicket code base.

Kick the tires

- Create a Quickstart via the wicket-archetype-quickstart
- Navigate to the pom.xml in your Quickstart project.
- Set version element to the same value as the one found in the Wicket project pom.xml (e.g. 1.4-SNAPSHOT)
- The plugin should automagically resolve the quickstart's Wicket dependency from the project rather than a .jar
- Find WicketApplication.java
- Click through to WebApplication class from the extends section of the class definition.
- If all is well, you should be taken to the Wicket project .java version of WebApplication rather than a .class file from a .jar

Proving a bug exists

- **Writing a unit test with WicketTester**

Submitting a patch

- **Extract your changeset into a patch file** - As we don't all have commit access to the svn repository, we need some other way of getting our changes to the developers. This is done by making an svn patch file, essentially a plain text diff, with some subversion specific formatting. With the Subclipse plug-in creating a patch file is as simple as:

1. Right click on the project root (e.g. wicket)
2. > Team > Create Patch...
3. Save In File System (and provide a filename, of the form **WICKET-<bug id number>-fix.patch**)
4. > Next >
5. Patch Root: Project
6. Finish, you now have your patch file where you chose to put it in step 3
7. Attach it to the relevant bug report with any further information you have