

## Quickie: how to make Clover ignore private constructors in Ant

by Benny Bottema - Friday, April 15, 2011

<http://www.bennybottema.com/2011/04/15/how-to-make-clover-ignore-private-constructors-in-ant/>

Last week we took a quick look on [how to make clover ignore private constructors in Eclipse](#). This week, we'll quickly cover Ant

- Also: [How to make Clover ignore private constructors in Maven](#)

First, let's look at the regular expression again:

```
(.* )?private +[a-zA-Z0-9_]+ *( *).*
```

Clover uses the concept of [coverage contexts](#), which you can include/exclude during the report phase. A Coverage Context is a filter that matches on method or statement level. Assuming you've got Clover setup with your Ant scripts, here's how you define a method context for your private constructors. Look up your **clover-setup** call:

```
<clover-  
setup initString="${build.dir}/clovercoverage.db" enabled="true">  
  <methodContext name="private_constructors" regexp="(.* )?private +[a-  
zA-Z0-9_]+ *( *).*" />  
</clover-setup>
```

So we've added a Method Context to match our constructors and gave it the name *private\_constructors*, which we'll need in a second. Note that all contexts you add define exclusions only. Everything else is assumed to be included by Clover.

Next to make use of this context, here's a tricky bit: it's not entirely clear in the Clover manual, but to make this Coverage Context active, you need to include it in the [format](#) element in your **clover-report** task, as follows:

```
<clover-report>  
  <current outfile="${reports.dir}/clover">  
    <format type="html" filter="private_constructors" />  
  </current>  
</clover-report>
```

For this to work you need to re-instrument your code. Have fun!

---

PDF generated by Kalin's PDF Creation Station