

Things to check before you consider submitting a (final) CDK patch #1

Egon Willighagen 

Published July 31, 2009

Citation

Willighagen, E. (2009). Things to check before you consider submitting a (final) CDK patch #1. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/h3v19-vh903>

Keywords

Cdk, Rse

Abstract

Mark the final in the above title; if you merely seek advice on your patch, feel free to send them in whatever state.

Copyright

Copyright © Egon Willighagen 2009. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

chem-bla-ics

Mark the *final* in the above title; if you merely seek advice on your patch, feel free to send them in whatever state. However, if you bring up your patch for peer review, make sure to have gone through the following steps, in random order:

- be prepared for peer review feedback
- realize your code will have to be [LGPL](#) or LGPL-compatible
- make sure the copyright lines are properly updated (see [Making patches; Attribution; Copyright and License.](#))
- your code is fully unit tested
- your code does not cause [PMD](#) failures
- your code is fully JavaDoc-umented
 - no empty templates
 - JavaDoc for every class field, method and class
 - use of {@link}
 - use of CDK tags @cdk.bug, @cdk.cite, etc
 - period at the end of the first sentence
 - ...
- make sure all the code still compiles
- make your code readable
 - 80 characters per line
 - variable names that reflect their purpose and nature
 - no code complexity errors with PMD
 - camelCasing as custom in Java
 - comment your code where appropriate, explaining what your code is supposed to do
 - ...

These are reasons to reject your patch, so better make sure to not have to be reminded of that. The build environment comes with some code to make these checks easier (though not the fixing). For example, say I introduced a new module `uff` (for the UFF force field):

```
$ cd cdk/  
$ ant clean dist-all test-dist-all  
$ ant -Dmodule=uff test-module  
$ ant -f javadoc.xml -Dmodule=uff doccheck-module  
$ ant -f pmd.xml -Dpmd.test=custom -Dmodule=uff test-module  
$ ant -f pmd.xml -Dpmd.test=custom -Dmodule=test-uff test-module
```