

Bioclipse git experiences #1: Strip away unwanted plugins

Egon Willighagen 

Published July 2, 2020

Citation

Willighagen, E. (2020, July 2). Bioclipse git experiences #1: Strip away unwanted plugins. *Chem-bla-ics*. <https://doi.org/10.59350/a75k0-fjg73>

Keywords

Bioclipse, Git

Copyright

Copyright © Egon Willighagen 2020. 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

This is a series of two posts repeating some content I [wrote up back in the Bioclipse days](#) (see also [this Scholia page](#)). They both deal with something we were facing: restructuring of version control repositories, while actually keeping the history. For example, you may want to copy or move code from one repository to another. A second use case can be a file that must be removed (there are valid reasons for that). Because these posts are based on Bioclipse work, there will be some specific terminology, but the approach I regularly apply in other situations.

For this first post, think of a *plugin* as a subfolder, tho it even applies to files.

Strip away unwanted plugins

In this case, you remove everything you do not want in your new git repository. Do:

```
git clone --bare --no-hardlinks old.local.clone/ new.local.clone/
```

Then use:

```
git filter-branch --index-filter 'git rm -r -q --cached --ignore-unmatch  
plugins/net.bioclipse.actionHistory plugins/net.bioclipse.analysis' HEAD
```

It often happens that you need to run the above command several times, in cases when there are many subdirectories to be removed. When you removed all the bits you need removed, you can clean up the repository and reduce the size considerably with:

```
git repack -ad; git prune
```