

# Offline CDK development using git-svn

Egon Willighagen 

Published October 31, 2007

## Citation

Willighagen, E. (2007). Offline CDK development using git-svn. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/fvbts-sc941>

## Keywords

Git, Svn, Cdk

## Copyright

Copyright © Egon Willighagen 2007. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## chem-bla-ics

While [Subversion](#) is a significant improvement over [CVS](#), they both require a central server. That is, they do not allow me to commit changes when I am not connected to that server. This is annoying when being on a long train ride, or somewhere else without internet connectivity. I can pile up all my changes, but that would yield one big ugly patch.

Therefore, I tried [Mercurial](#) where each client is server too. The version I used, however, did not have the move command, so it put me back into the old CVS days where I lost the history of a file when I reorganize my archive.

## Git

Then [Git](#), the version control system developed by [Linus Torvalds](#) when he found that existing tools did not do what he wanted to do. It seems a rather good product, though with a somewhat larger learning curve, because of the far more flexible architecture (see [this tutorial](#)). Well, [it works for the Linux kernel](#), so must be good :)

Now, [SourceForge](#) does not have Git support yet, so we use Subversion. [Flavio](#), of [Strigi](#) fame, however, [introduced me to git-svn](#). Almost two month ago, already, but finally made some time to try it out. I think I like it.

This is what I did to make [a commit to CDKs SVN repository](#):

```
$ sudo aptitude install git-svn git-core
$ mkdir -p git-svn/cdk-trunk
$ cd git-svn/cdk-trunk
$ git-svn init https://cdk.svn.sourceforge.net/svnroot/cdk/trunk/cdk
$ git-svn fetch -rHEAD
$ nano .classpath
$ git add .classpath
$ git commit
$ git-svn dcommit
```

The first `git-svn` command initializes a local Git repository based on the SVN repository. The `git-svn fetch` command makes a local copy of the SVN repository content defined in the previous command. Local changes are, by default, not committed; unless one explicitly `git` adds them to a patch. Once a patch is ready you can do all sorts of interesting things with them, among with commit them to the local Git repository with `git commit`.

Now, these kind of commits are on the local repository, and I do not require internet access for that. When I am connected again, I can synchronize my local changes with the SVN repository with the `git-svn dcommit` command.

A final important command is `git-svn rebase`, which is used to update the local `git` command for changes others made to the SVN repository.