

# How to use GitHub for [CDK|Bioclipse] code review

Egon Willighagen 

Published May 10, 2010

## Citation

Willighagen, E. (n.d.). In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/nfcyx-jxj82>

## Keywords

Cdk, Bioclipse, Github

## Copyright

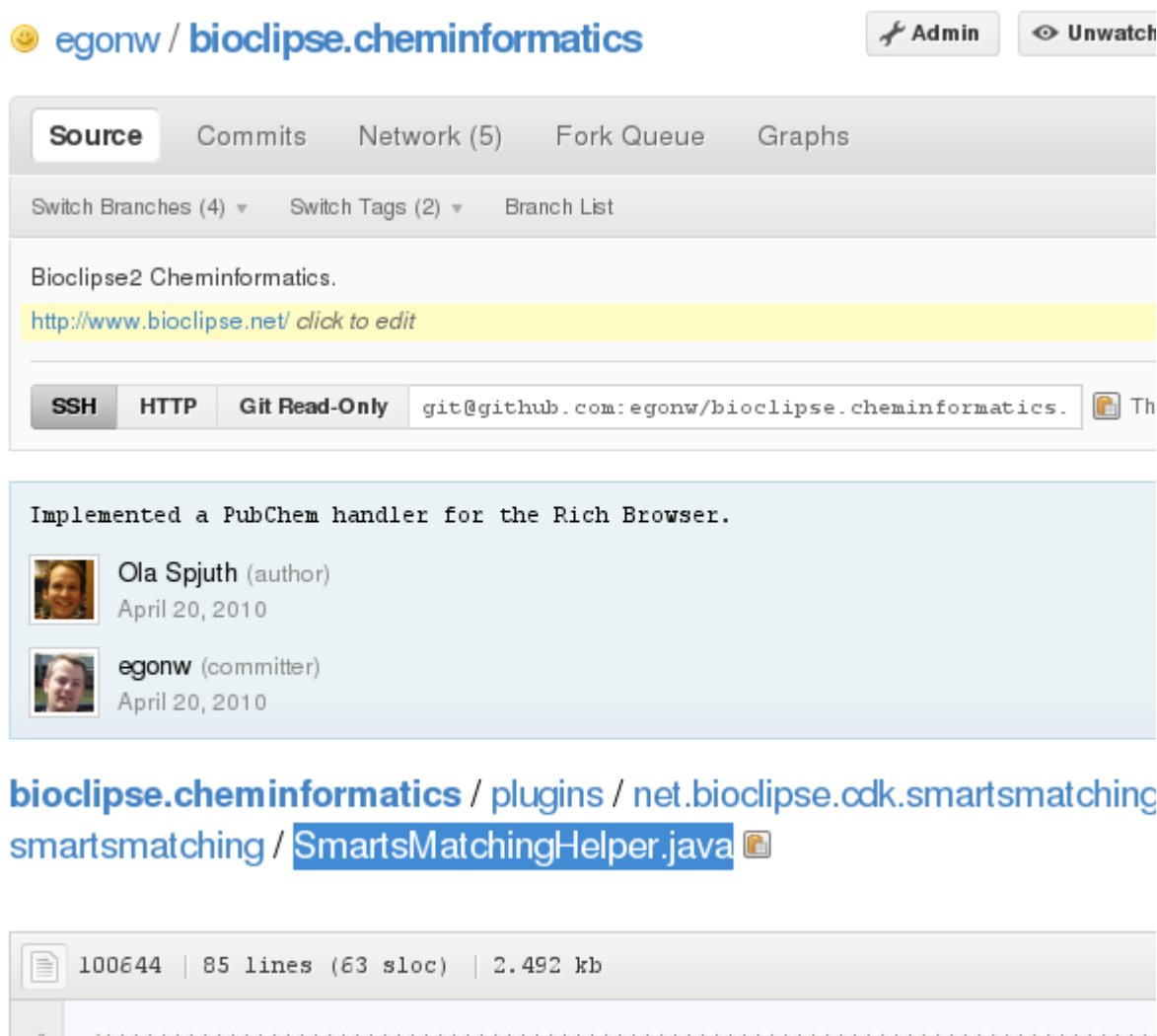
Copyright © Egon Willighagen 2010. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Triggered by posts in the past [three days](#), I thought about writing up a short tutorial on how to perform code review for existing code on [GitHub](#). Therefore, this applied to [CDK](#) and [Bioclipse](#) source code, many but will work for any project hosted in GitHub. Even if it is not, you could consider putting up a copy there yourself. This example will demonstrate the procedure on CDK functionality in Bioclipse in the [bioclipse.cheminformatics](#) repository.

*Click on the images to get a higher resolution version.*

## Step 1: find the class you want to review

Use the GitHub web interface to browse your way towards the source code of the class you want to review. For example, the [SmartsMatchingHelper.java](#):



egonw / **bioclipse.cheminformatics** [Admin](#) [Unwatch](#)

**Source** Commits Network (5) Fork Queue Graphs

Switch Branches (4) ▾ Switch Tags (2) ▾ Branch List

Bioclipse2 Cheminformatics.  
<http://www.bioclipse.net/> [click to edit](#)

**SSH** **HTTP** **Git Read-Only** `git@github.com:egonw/bioclipse.cheminformatics.` [Th](#)

Implemented a PubChem handler for the Rich Browser.

 **Ola Spjuth** (author)  
April 20, 2010

 **egonw** (committer)  
April 20, 2010

**bioclipse.cheminformatics** / **plugins** / **net.bioclipse.odk.smartsmatching**  
**smartsmatching** / **SmartsMatchingHelper.java** [Th](#)

 **100644** | **85 lines (63 sloc)** | **2.492 kb**

## Step 2: identify something you like to comment on

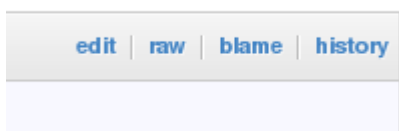
Next step is to perform some code reviewing. For example, we might want to ask something about how `parseProperty()` works:

```
29
30     /**
31      * Deliver a list of integers to highlight
32      * @param o
33      * @return
34      */
35     public static List<Integer> parseProperty( Object o ) {
36
```

Now, this page on GitHub does not provide the means to leave comments; instead, you comment on commits.

## Step 3: find the last commit that touched the line you like to comment on

Git has a [blame option](#) (also called annotate) which will show you for each line who last changed that line. The GitHub web page makes this functionality available with the 'blame' link just above the first line of the source code:



This link will lead us to a page with a new column on the left side showing commit hashes, name of the commit author, and the first few characters of the commit message. For example, the [web page](#) bits relevant to code we want to comment on, looks like:

3ce78ba5 » Ola Spjuth  
2009-10-23 Added support for SmartsMat...

```

11 package net.bioclipse.cdk.smartsmatching;
12
13 import java.util.ArrayList;
14 import java.util.Collections;
15 import java.util.List;
16 import java.util.Set;
17 import java.util.StringTokenizer;
18
19 import net.bioclipse.cdk.business.CDKManager;
20 import net.bioclipse.cdk.domain.ICDKMolecule;
21
22 import org.apache.log4j.Logger;
23 import org.openscience.cdk.interfaces.IAtom;
24
25
26 public class SmartsMatchingHelper {
27
28     private static final Logger logger = Logger.ge
29
30     /**
31      * Deliver a list of integers to highlight
32      * @param o
33      * @return
34      */
35     public static List<Integer> parseProperty( Obj
36
37         if (!( o instanceof String ))

```

This shows us that commit [3ce78ba5](#) is the one we are interested in:

Added support for SmartsMatching on MolTable.  
 \* Added checkbox for SmartsMatchingView  
 \* Rewrote models to support SmartsFiles  
 \* Added import SMARTS from file  
 \* Running SmartsMatching with MolTable open highlights hits in chemical structures.

Smartsmatching in MolTable is very slow, mainly because results are not persisted among redraws (see bug 1676).

Solves bug 1675.



Ola Spjuth (author)  
October 23, 2009

```

commit 3ce78ba572e0fbb32536
tree 8579187c208442d9571a
parent b0a8e73a4896ffc86acc

```

## Step 4: Look up the line again and add a comment

In the web page with the appropriate commit looked in the previous step, you scroll down to the line you want to comment on. If you hover over that line, a blue comment bubble will show up on the left side:

```

16 + public class SmartsMatchingHelper {
17 +
18 +     private static final Logger logger = Logger.getLogger(CDK
19 +
20 +     /**
21 +      * Deliver a list of integers to highlight
22 +      * @param o
23 +      * @return
24 +      */
25 +     public static List<Integer> parseProperty( Object o ) {
26 +

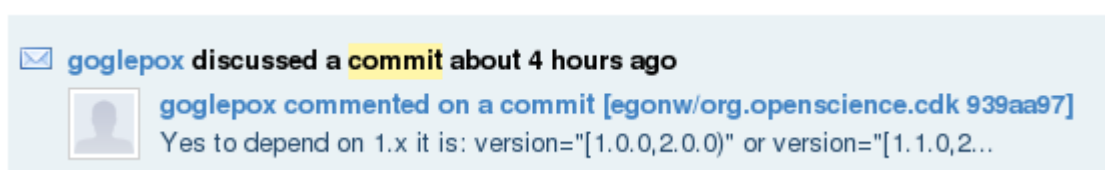
```

Clicking that blue comment icon, you get a dialog where you can enter your comment:

The 'Add Line Note' button confirms and saves your comment:

## Step 5: inform the commiter about your review

The next step would be to inform the commit author. GitHub actually helps here, and should send a message, like this one:



But it would certainly not hurt if you filed a bug report or sent an email.

## **chem-bla-ics**

*Now, I should only convert this into a screencast...*