

How can I run updates in SVN without causing many conflicts

Asked 15 years, 10 months ago Modified 15 years, 10 months ago Viewed 166 times



0



I am using SVN to develop some software in a team of four people.

Whenever I update my working copy, it creates far more conflicts than I would expect to be created, a large number of these conflicts are like so:

If the initial was

```
import Class1
```

and I change it to

```
import Class1
import Class2
```

while another team member changes the code to

```
import Class1
import Class3
```

Subversion will not merge all three lines.

I suppose this is because subversion isn't aware of the semantic content of the code, so couldn't decide when it would/wouldn't be appropriate to do this.

Another conflict I get is if I replace

```
processA();
processB();
```

with :

```
if(x){
    processA();
    processB();
}
```

the whole thing is marked as a change, is there no way to get it to highlight the fact that only if(x) was added above/below. In cases like this i have to analyse the entire lines inside the new block, and worry about missing small changes in the lines inside.

The question is, is there a more effective way for my team to use SVN in these cases, a large portion of my time is spent merging.

Sorry for being so long winded in my expression, thanks for reading.

java svn

Share Improve this question Follow

asked Feb 23, 2009 at 14:54



Ben Page

3,080 ● 4 ● 37 ● 39

3 Answers

Sorted by: Highest score (default) ▾



6

Since SVN doesn't know about programming languages it would be very dangerous if it would try to deduce semantic meaning from change sets. Therefore it can only do relatively simple merges (i.e. if a file changed in two completely different areas).



The common solution is to use small commits. Don't keep big changes piling up. Commit often. Update often.



Share Improve this answer Follow

answered Feb 23, 2009 at 14:57



Joachim Sauer

308k ● 59 ● 565 ● 620

Thank you. Been committing regularly for a while and it defiantly solved the problems.

– Ben Page Mar 12, 2009 at 14:17



1

If you check in frequently, you'll still see the big changes (e.g. in the "if" case) but you won't have much *merging* to do.



Share Improve this answer Follow

answered Feb 23, 2009 at 14:58



Jon Skeet

1.5m ● 889 ● 9.3k ● 9.3k





1



Try to keep developers working on disjoint parts of the project to reduce merge problems. If developers do overlap, encourage communication to avoid mismerges. Also, try to commit early / often to avoid massive merges but don't allow for commits which render your baseline unbuildable. A continuous integration system helps.

[Share](#) [Improve this answer](#) [Follow](#)

answered Feb 23, 2009 at 15:02



[basszero](#)

30k ● 9 ● 58 ● 80