
Subject: Re: SVN plan...

Posted by [mr_ped](#) on Mon, 12 Nov 2007 13:42:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

mdelfede wrote on Mon, 12 November 2007 14:14

That's my 3rd point above.... doew work, but it's slow !

IMHO the best would be fetch past revision in current tree AND provide some locking mechanics to avoid editing of files when an older revision is fetched in.

The slow, but most safe solution.

With going back in the same directory you will keep the unversioned local files available, which may affect the running of executable.

While I tend to keep local files around, I prefer to have in SVN always fully build-able sources, and this can be tested reliably only by check-outing into empty directory and building from scratch.

mdelfede wrote on Mon, 12 November 2007 14:14

Just another example : you're working on rev. 50, you find a bug. You fetch past revisions, doing a binary search, and you find that the bug appears between 44 and 45. You do it recompiling each time, but it should be fast enough, as only few files change. So, you find the bug, ask theide to create a diff between buggy and non buggy code, revert to latest version and apply the patch, or correct manually the code.

That could work on separate trees, but would be awfully slow...

You think you will compile only files which did change during going to different revision?

So far from my experience with TheIDE the dependency check is extremely unreliable, I often have to do "Rebuilt all" twice a day (as the only package which I can safely assume is compiled correctly already is "Core", and it's too much work to select+clean package other 13 of them by hand) to get correct binary.

If I would find with your method that the bug is in between 44 and 45, I would think twice if I should firstly check source difference (if I can spot the bug quickly), or to rebuild them both from scratch to make sure it's really those two and not some old object file hunting the test results. (especially with U++ and TheIDE)

Another thing is that with fresh cloned nests you can keep several revision on the disk at the same time, making the switching between them and comparing easy even for tools which are not aware of SVN.

It's surely slower, but with modern 2GHz Core2 CPU I can rebuild my current project in 10sec, what is absolutely fantastic for me.

If the rebuild would take 2mins, I would probably like the option of your approach, just to compile modified files and worry about binary is trustworthy after I see strange results.

So I think the speed issue depends a lot on the HW used and project size.

About single button for SVN update...

From Mirek posts I have the feel he uses the same approach even for commits?

While I appreciate simple Update (3 click with TortoiseSVN: right click on the file folder, left click

on the "Update" menu, left click on the "OK" button in results window with modified files listed), I do need to write commit log comments and to check out modified files ahead of commit one more time, if everything looks ok. So any simple commit button will not really work for me.

So we have already 3 different views on SVN usage (from Mireks "it's not good for me" trough yours and my little details and differences) ... This will be a tough thing to do it right.
