

How do you setup a shared Working Copy in Subversion

Asked 16 years, 2 months ago Modified 13 years, 6 months ago

Viewed 6k times



I still very new using Subversion.

11



Is it possible to have a working copy on a network available share (c:\svn\projects\website) that everyone (in this case 3 of use) can checkout and commit files to? We don't need a build server because it is an asp site and the designers are used to having immediate results when they save a file. I could try and show them how to set it up local on their machines but if we could just share the files on the development server and still have the ability to commit when someone is done, that would be ideal.



An easy solution would be for all of us to use the same subversion username and that would at least allow me to put files under version control.

But is it possible to checkout a folder from the svn repository but still require each person to login with their user/pass to commit?

EDIT: I'm trying to take our current work flow, which is editing the LIVE version of a site using Frontpage Extensions or FTP. And move it to something BETTER. In this case a copy of the live site on a development server

that I setup to mirror the live server, remove frontpage extensions access. Then the designers can still have the same effect of instant gratification but I will not have to worry they are editing the live files. Even using a shared user/pass in subversion is still version control. It may not be ideal and if the designers were actually programmers I would try to get them fully on board but that's just not the case. This is the best I can do in this case and avoid a huge learning curve and work stoppage.

svn

Share

edited Oct 1, 2008 at 11:40

Improve this question

Follow

community wiki

2 revs

Brian Boatright

I don't understand why you keep modding the question down? There is nothing wrong with the question and it is a valid one, even if it doesn't fit with everyones idea of how it SHOULD work. – [kaa](#) Oct 1, 2008 at 11:38

@kaa - thanks Kaa! please vote it up,even one vote helps. they are voting it down because what I want to do isn't "normal" in their view. seems like a lot of people forget they have a brain and don't have to stick to only what is found in a book, which I have btw and mostly read. thanks again.

– [Brian Boatright](#) Oct 1, 2008 at 11:42

9 Answers

Sorted by:

Highest score (default)



6



In my experience it will work just fine out of the box. At my company we have had this setup for a number of years and not experienced any problems (outside the obvious ones of having a shared working copy).

You should however look into having separate working copies and a trigger (hook) that updates the shared location on commits if you need a "live" version of the site.

Share Improve this answer

answered Oct 1, 2008 at 11:34

Follow



[kaa](#)

1,461 ● 2 ● 13 ● 18

Hooks are evil, just like triggers on the DBs – [Rinat Abdullin](#)
Oct 1, 2008 at 12:39

7 Hooks, like triggers, like alcohol should be enjoyed responsibly. A blanket statement like "Hooks are evil" isn't helpful, you should always strive to provide a reasoning.
– [kaa](#) Oct 1, 2008 at 14:05



2



You can't checkout a working copy, a **working copy** is the term used for code that has already been checked out. If you are asking multiple developers to work with the same set of working files at the same time, then you are seriously undermining one of the main uses of having a version control system, which is to allow your developers



to make changes independently of one another without breaking things for anyone else.

That said, if you really want to do this you can. With a Linux server, the way to go is to have each of your users running a different ssh user agent (for windows machines we use [Pagent](#)) with a different ssh identity for each user. Then have the svn server recognize the ssh-tunnels from different identities as being from different users. Unfortunately, I don't know how to set that up in Windows.

Share Improve this answer

edited Oct 1, 2008 at 11:36

Follow

answered Oct 1, 2008 at 11:31



[Adam Bellaire](#)

110k ● 19 ● 152 ● 165

thanks adam. I understand my use is outside the norm. but as I told ironfroggy it is better than nothing and having a shared user/pass to commit files is as close as I can get to their existing "Frontpage Extensions" workflow ;-)

– [Brian Boatright](#) Oct 1, 2008 at 11:33



2



You need to use svnserve (light-weight SVN server that comes with SVN) or apache mod.

With it, you can [configure permissions](#) like this:

```
[general]
password-db = userfile
```



```
realm = example realm
```

```
# anonymous users can only read the repository  
anon-access = read
```

```
# authenticated users can both read and write  
auth-access = write
```

Share Improve this answer

answered Oct 1, 2008 at 12:09

Follow



[Rinat Abdullin](#)

23.6k ● 8 ● 60 ● 80



I know this is an old thread, but I found it because I'm trying to do the same thing.

1



I don't think this problem can be solved with subversion, though I've tried on multiple occasions and I think the need is completely legitimate.



The use case that comes up repeatedly for us is complicated configuration files that are maintained outside of an application.

A good example might be apache httpd.conf files. They're complicated and we want to track changes to the files. We don't want everyone to use "root" because then we can't track who did what.

Mercurial can do this:

[Mercurial Multiple Committers](#)

Share Improve this answer

answered Jun 9, 2011 at 15:24



0

Working copies are meant for each user to have his own. And the repositories are shared. Only the person who checked out the WC can commit changes from it.



Share Improve this answer

answered Oct 1, 2008 at 11:28

Follow



Vasil

38k ● 27 ● 93 ● 116



ok. so the only option is what I suggested which is to use a "shared" user/pass? – [Brian Boatright](#) Oct 1, 2008 at 11:28

but as ironfroggy said that would defeat the purpose of using version control. – [Vasil](#) Oct 1, 2008 at 11:31

it doesn't "defeat" version control, it only allows one user, the shared user, versions. – [Brian Boatright](#) Oct 1, 2008 at 11:34

ok, I should correct my comment. Working on that kind of setup will disable them from using a lot of vcs functionality. – [Vasil](#) Oct 1, 2008 at 11:42

- 1 @JeeBee - I understand the point of version control. I'm just not a robot and use tools like subversion to better my own situation. Not just conform to what someone else tells me in a book is the best way. – [Brian Boatright](#) Oct 1, 2008 at 11:45



0



Maybe you could look at something other than subversion if you don't want a server, like a distributed VCS (Bzr, git, and mercurial are popular these days) or you should look at subversion hosting services.

Sharing a single working copy is not recommended. That really defeats the purpose of version control. Please don't do that.

Share Improve this answer

answered Oct 1, 2008 at 11:29

Follow



[ironfroggy](#)

8,109 ● 7 ● 37 ● 44

it doesn't defeat it entirely. in this case there are still "versions" of the file but only by one user. I understand what you are saying just try to get two designers to setup and host their own WC and explain how it all works. this is the closest I can get to their existing workflow without FrnPg

– [Brian Boatright](#) Oct 1, 2008 at 11:31



0



You are meant to have your SVN repository, and each user (with their own username and password to access the SVN repository) should check out their own working copy.

It is possible to do this on a single PC (is that your problem, PCs with multiple developers sharing it?) by having different PC user accounts and having the people check out into their own account, or even by sharing a PC user account and having the people check out into a

different working folder. I don't think this is particularly neat or nice, and if a company can't afford a PC per developer these days then it is hardly worth working for!

I recommend:

1. Each employee has their own working copy on their own PC.
2. There should be an ANT or Maven or similar build script that will allow a developer to build and deploy from their working copy onto the development web server so they can see how it is. This could be simple as "copy files to this shared location".
3. As each employee has their own SVN username/password you can see who made which change, and lock people out when they leave the company.
4. This might create a process that a designer has to follow rather than the anarchy you have currently, but if it takes any of them more than half a day to pick up SVN and how to run a build script to deploy to the development web server then you've got bigger problems.

Share Improve this answer

edited Oct 1, 2008 at 11:58

Follow

answered Oct 1, 2008 at 11:35



JeeBee

17.5k ● 5 ● 52 ● 60

each employee has a computer. it's more of a learning curve issue – [Brian Boatright](#) Oct 1, 2008 at 11:36



0



I think you're selling your team short. Non-devs *can* easily learn to use SVN, especially with something like tortoise. If you're gonna go through the trouble of setting up SVN, then just give everyone a separate login and let them work on their own local working copy. Do a quick and dirty CruiseControl automated build to pull from SVN to create the staging site content. It's just a little more work and the result will be so much nicer.

Share Improve this answer

answered Oct 1, 2008 at 12:24

Follow



[Chris Farmer](#)

25.4k ● 36 ● 123 ● 168

I think you may be underestimating the surprisingly-common technophobia of designers and content managers. – [3Dave](#) May 27, 2010 at 16:51



0



I know this is an old question/wiki, but a solution that occurred to me is as follows.

The problem could be restated as follows: How can we use a real source control system (like SVN), and still allow non-technical designers to enjoy the same save-preview-save-preview cycle that they've come to know and love?

Points:

- In order to realize the full benefit of SVN, each user needs to have their own working copy. That's just the way it is.
- If you've still got classic ASP (NOT .NET) in your app, a lightweight local server like Cassini won't do the job.
- It'd be preferable for the user to not have to install or learn to use an SVN client like the (admittedly-simple) Tortoise.

My approach:

- Create a branch in SVN for each user
- On each client, provide a mapped drive to a user-specific working directory on your dev server. They will use FrontPage, Expression, SharePoint designer or whatever to make their changes here.
- In IIS on the dev server, create a user-specific website (for example, `alice.www.mysite.com` or `bob.www.mysite.com`) with a user-specific host header. They will browse the site through this URL to see their changes. This also allows them to show their changes to others before merging it into the trunk.
- Using CruiseControl.NET, provide tasks to check out, update, add and commit changes for each user's branch. Figure out how to make this so that each user can only see their own tasks.

- Using CruiseControl.NET, create a task that will merge their changes into the trunk
- Using CruiseControl.NET, create a task that will update the real dev site (dev.www.mysite.com) with the merged changes. This site will show everyone's work combined, and acts as a staging and debugging area. If you are using a WAP, you'll want this task to trigger a build as well.

Sounds like a lot of steps, but it's really pretty simple. Create branches, map drives, set up new IIS sites and let them go crazy.

Under the covers, this is exactly the same as giving them a local IIS install and letting them commit their changes with SVN whenever necessary, just as developers do. The difference is that their working copy is on a server, IIS/Cassini doesn't have to be on their box, and they'll use a web interface to perform SVN actions like commit, update, etc.

Good luck!

Share Improve this answer

answered [May 27, 2010 at 17:13](#)

Follow

community wiki
[3Dave](#)
