Do you use the branches/tags/trunk convention?

Asked 16 years, 3 months ago Modified 16 years, 3 months ago

Viewed 3k times

▲

7

▼

Do you always follow the convention of putting branches, tags and trunk directories at the top level of your Subversion repository? Lately, I have stopped bothering and nothing bad has happened (yet)!

It should be possible to move directory trees around if there's ever a need to create them. Am I building up trouble for later?

svn

Share

Improve this question

Follow

edited Sep 24, 2008 at 1:10

Orion Edwards

123k ● 66 ● 245 ● 339

asked Sep 23, 2008 at 19:40

Steve Pitchers

7,336 ● 5 ● 44 ● 42

15 Answers

Sorted by: Highest score (default)

12

✓

Have you tried branching or tagging yet? Until then, there's no problem. However, an added benefit of using the branches,tags,trunk convention is that it's exactly that -- a convention. People expect to see that, so they know what to do when they need to fork.

Share Improve this answer

Follow

answered Sep 23, 2008 at 19:43

Danimal

7,710 ● 8 ● 48 ● 57

No there are very few tags in these repositories, as they happen to be intended for documentation and project management rather than active software development. – Steve Pitchers Sep 23, 2008 at 20:00

2

Quick answer is "do whatever best suits your procedures".

As Danimal said the structure of branch/trunk/tag is a convention. However, I don't agree that it's the location of the b/t/t that is important, merely the existence of them.

What you should have is somewhere that is obviously designated for branches, somewhere designated for your trunk and same for your tags. Exactly where they fall depends very much on the structure of your repository and the nature of the files that you're keeping.

For example, if you are keeping multiple projects in one repository you'll probably find that it makes more sense to create your b/t/t directories under your projects. If you have distinct modules within your project then the b/t/t should be created under the module directories.

Ask yourself what the biggest logical chunk is going to be that you wish to branch and be guided by that.

Share Improve this answer Follow

answered Sep 24, 2008 at 1:44

Andrew Edgecombe
40.3k ● 3 ● 38 ● 63

▲
1
▼

This depends on how big the project is. We have some stuff (granted, in git, but the concept is the same) that is fairly big. Every developer uses his/her own branch, there is a testing and mainline branch. We also tag the releases, and if there are version-specific fixes, a branch is created so fixes can be integrated fairly easy.

This setup has advantages: we don't get in each others hair during developement. But the downside is that we need an integrator to put the commits from the developers branch into the testing branch, and then to the mainline one.

If the project is small, then it's just overhead, but you never know how big a project will get.

Share Improve this answer Follow

answered Sep 23, 2008 at 19:44

terminus
14.4k ● 8 ● 37 ● 38

Tags in Git are at least useful - you can see which tag a commit is before/after, they act as mnemonics for commits, etc, etc. In subversion tags are a second class citizen and a complete waste of time. You can't do "svn log -r (tag-name)" or whatever. Even CVS did tags better. – richq Sep 23, 2008 at 19:49

SVN treats everything as a directory. In order to tag a branch, you copy it to a directory in the ./tags/ directory. Likewise for branches in SVN. In git branches and tags are idependent and done proper – Amandasaurus Sep 23, 2008 at 22:14

▲

1

▼

I just started to actually use the convention, and I agree with Danimal. If you have one build in QA, and another in Production, and another in crazy-new-experimental feature development, it's nice to quickly switch back and forth between them.

Share Improve this answer Follow

edited May 23, 2017 at 11:57
Community Bot
1 ● 1

answered Sep 23, 2008 at 19:46
Dave Neeley
3,625 ● 2 ● 26 ● 42

▲

I've written tools in the past to automate certain pieces of SVN. Creating a basic repository is one of them. Step 1: create an empty repository. Step 2: create trunk,

1

branches and tags folders - commit Step 3: Copy hook scripts to new repository

One of my hook scripts is to make sure that the items in the tags directory cannot be modified. This makes tags have a meaning different from branches.

Share Improve this answer Follow

answered Sep 24, 2008 at 2:18

Brad Bruce
7,807 ● 3 ● 42 ● 61

0

Nope, have abandoned that approach for the projects currently in the queue. While the concept seems very valid, it just seems to waste more time than it saves in practice.

Share Improve this answer Follow

answered Sep 23, 2008 at 19:44

Brian Knoblauch
21.3k ● 16 ● 61 ● 96

2 Until you need to create a branch or tag a release. At that point you will waste hours of developer effort moving the existing codebase around. Easier to take the 30 seconds to create those directories at the start. – Tim Booker Sep 23, 2008 at 19:54

No, but once you move your code from the root into trunk you have a whole bunch of developers and automated processes pointing at nothing – Tim Booker Sep 23, 2008 at 20:21

0

As I said in [What do "branch", "tag" and "trunk" mean in Subversion repositories?](), since branch and tag are the same, you are not obliged to follow any convention but your own.
Especially for a small project with sequential development (i.e. no need for parallel efforts between current development, maintenance of older versions, exploration of alternative frameworks, ...)

Share Improve this answer Follow

edited May 23, 2017 at 12:33
Community Bot
1 ● 1

answered Sep 23, 2008 at 19:47
VonC
1.3m ● 558 ● 4.7k ● 5.6k

0

Do you at least have a trunk? If not, when you do need to branch or tag, you will have to have those sitting in your root project directory, alongside the actual code/contents. Yikes!

EDIT: I guess you could create a trunk folder, then move everything into that, then create your branches etc...

To those saying "just do it later, don't waste time, etc..." Honestly though, how much overhead is it to create them at the outset of your project? 2 minutes, tops? Why not just do it then? It will take much longer to move everything later - even if you only end up needing to branch 1 in 5 times, I still think you'd use less time starting with a branch, tag, trunk structure.

Share Improve this answer Follow

edited Sep 23, 2008 at 19:51

answered Sep 23, 2008 at 19:44

Chris Marasti-Georg
34.6k ● 17 ● 93 ● 137

It's not the time to create them, it's the need to type the extra path element during everyday use that irritates me! ~~~ – Steve Pitchers Sep 23, 2008 at 19:50

Why not use one of the several SVN frontends (IDE integration, TortoiseSVN, etc)? – Chris Marasti-Georg Sep 23, 2008 at 19:52

0

I use trunk, tags and branches on every project. Seriously, how hard is it to create 2 extra directories when you create the project. There is some benefit to following the convention just to maintain consistency. I find that I have lots of tags (each push of an app outside the developer environment gets versioned and tagged). I don't have so many branches because I'm generally not working with people I don't trust with a commit prior to review. So, usually when I get branches, it's because I have a permanent splitting of the codebase - usually for different clients. Once the code becomes irreconcilable, I generally stop a branch and move it to it's own trunk.

Share Improve this answer

Follow

answered Sep 23, 2008 at 19:59

mrankin

2,384 ● 3 ● 25 ● 29

▲

0

▼

Lately i´m using a model more focused in agile and you can take a look here.

It´s really important to following some policies in version control, cuase even using a well defined model, code version is in nature something that leads you to commit mistakes, messy merges, and all that bad stuff, so be careful.

This model gives responsibilities for each repository and does not let you overlap where you production, deliverable and under-construction code lies.

Share Improve this answer

Follow

answered Sep 23, 2008 at 22:04

paulosuzart

1,092 ● 2 ● 13 ● 28

▲

0

▼

I follow the convention for numerous reasons

1. Reference material and procedures which use the b/t/t convention can be instantly applied to your svn repo structure.
2. All developers coming into the team who are familiar with the convention have a minimal learning curve to get used to your svn repo structure.

3. Where as trunks & branches have an immediate and obvious benefit, it's only when you're having to trawl through histories and logs to cover your or your companies ass, that you realise the benefit of maintaining a consistent tagging procedure.

In short it may not be immediately obvious why the convention is a good thing, but it's when you need help, advice, or some management craziness that it becomes a proverbial godsend.

Share Improve this answer

Follow

answered Sep 24, 2008 at 0:56

flungabunga

390 ● 2 ● 9

▲

0

▼

I like to use the branches for "mini-projects" for simple proof of concepts. It's fast, easy and generally helps to keep up with your main project. I put proof of concepts in the branches directory since it isn't apart of the main project but it is of value to the project.

Like others have mentioned, I use the tags for releases. Most releases I do are in versions so I generally just have a zip file of the package or the version'd installer.

Share Improve this answer

Follow

answered Sep 24, 2008 at 1:15

nyxtom

2,929 ● 2 ● 24 ● 24

0

No. Not the last 3 work situations. I work with non-programmers who need to write, fix, and recall processing scripts. Programming is mostly casual, with occasionally deeper or bigger work. There's no expectation of following big-time software developer's practices. The standard repository terminology can clash with jargon used in the field we work in. So we make up our own repository directories.

Share Improve this answer Follow

answered Sep 24, 2008 at 2:01
DarenW
16.9k ● 8 ● 69 ● 104

0

In a very straightforward environment, you can get away with leaving out the branch, tag, trunk from the top of your SVN repository. For example, if you're using SVN for your university assignments, you're not going to be very concerned about changes to the code after it gets released to your customer (the person marking the assignment), and so you could sensibly dispense with branch, tag, trunk, and just have one structure. (Effectively, the whole thing is the 'trunk'.)

If, on the other hand, you've been managing code that is deployed to 700 different sites and that is split across separate product lines, you'd be insane not to use 'branch, tag, trunk' near the top of your structure (there's a sensible case for splitting your products before going down the BTT route), since you're going to need to know what code went where, and to be able to separate major rewrite activity (the stuff you do in the trunk) from spot fixes to help a site having an immediate problem (which you do in a branch, then merge into the trunk). And if you want to be able to answer the question "Why did the Foobar stop working when we rolled out patch 1.2.3?" then tags are essential.

Share Improve this answer

Follow

answered Sep 24, 2008 at 2:09

Nick Argall

▲

-1

▼

I'll generally keep my trunk in the root of the repository and only move it into a Trunk folder if I actually need to create a tag of a branch. I think with SVN, as long as your structure is logical, you should have no trouble rearranging it later if your needs change.

Share Improve this answer

Follow

answered Sep 23, 2008 at 19:48

Nick Higgs

1,702 ● 1 ● 18 ● 21