

# What are the advantages of using SVN over CVS?

Asked 16 years, 4 months ago   Modified 4 years, 2 months ago

Viewed 44k times



66

My company is using CVS as our de-facto standard for source control. However, I've heard a lot of people say that SVN is better.



I know SVN is newer, but other than that, I'm unfamiliar with its benefits.



What I'm looking for is a good, succinct comparison of the two systems, noting any advantages or disadvantages of each in a Java/Eclipse development environment.

svn

version-control

cvs

Share

Improve this question

Follow

edited Apr 11, 2013 at 20:22



Troy Alford

27.2k ● 11 ● 65 ● 83

asked Aug 4, 2008 at 14:06



shsteimer

28.8k ● 30 ● 80 ● 95



68



CVS only tracks modification on a file-by-file basis, while SVN tracks a whole commit as a new revision, which means that it is easier to follow the history of your project. Add the fact that all modern source control software use the concept of revision so it is far easier to migrate from SVN than it is from CVS.

There is also the atomic commit problem. While I only encountered it once, it is possible that 2 people committing together in CVS can conflict each other, losing some data and putting your client in an inconsistent state. When detected early, these problems are not major because your data is still out there somewhere, but it can be a pain in a stressful environment.

And finally, not many tools are developed around CVS anymore. While the new and shiny-new tools like Git or Mercurial definitely lack tools yet, SVN has a pretty large application base on any system.

**EDIT 2020:** Seriously, this answer is 12 years old now. Forget SVN, go use Git like everyone else!

answered Aug 4, 2008 at 14:17

**Vincent Robert****36.1k** ● 15 ● 83 ● 122

- 
- 1 I want the same answer for git vs subversion and git vs mercurial – [Isaac Rabinovitch](#) Jan 15, 2017 at 20:12
- 
- 3 "Forget SVN, go use Git": I don't think that's a matter of old/new or about how many people use it, they are different things – [golimar](#) Oct 7, 2020 at 11:06
- 
- > "Seriously, this answer is 12 years old now. Forget SVN, go use Git like everyone else!" | It is 2022, and the Foundation I work for just switched from Git to CVS. I guess our use-case/workflow is similar to OpenBSD. – [gsl](#) Jul 4, 2022 at 12:48 ✎
- 
- 1 The 2020 Edit above is misleading. SVN does have some advantages over git especially working with large repos containing large binary files. In this case, SVN is faster since you only download (and upload) the files you need to modify, not the whole repo. SVN is also easier to master for newbies. See [blog.programster.org/advantages-of-svn-over-git](http://blog.programster.org/advantages-of-svn-over-git) But for most software development projects, git is typically the better choice. – [RobK](#) Feb 12, 2023 at 1:36
- 



One of the many comparisons:

[http://wiki.scummvm.org/index.php/CVS\\_vs\\_SVN](http://wiki.scummvm.org/index.php/CVS_vs_SVN)



Now this is very specific to that project, but a lot of stuff applies in general.



Pro Subversion:



- Support for versioned renames/moves (impossible with CVS): Fingolfin, Ender
- Supports directories natively: It's possible to remove them, and they are versioned: Fingolfin, Ender
- File properties are versioned; no more "executable bit" hell: Fingolfin
- Overall revision number makes build versioning and regression testing much easier: Ender, Fingolfin
- Atomic commits: Fingolfin
- Intuitive (directory-based) branching and tagging: Fingolfin
- Easier hook scripts (pre/post commit, etc): SumthinWicked (I use it for Doxygen after commits)
- Prevents accidental committing of conflicted files: Salty-horse, Fingolfin
- Support for custom 'diff' command: Fingolfin
- Offline diffs, and they're instant: sev

Follow



Michael Stum

181k ● 119 ● 407 ● 540

---

2 What is a "fingolfin" ? Googling that word makes it only more confusing – [DuKes0mE](#) May 15, 2015 at 12:57 ✎

---

3 Fingolfin and Ender are just the Names of People of that Forum: Behind each "argument" is a list of people who agree that this particular argument is valid. – [hinneLinks](#) Jul 6, 2015 at 7:02 ✎

---



15



SVN has 3 main advantages over CVS

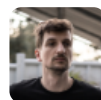
- it's faster
- supports versioning of binary files
- and adds transactional commit (all or nothing)



Share Improve this answer

answered Aug 4, 2008 at 14:35

Follow



lubos hasko

25k ● 10 ● 57 ● 62



7



The Subversion book has [an appendix](#) that details important differences from CVS, which may help you make your decision. The two approaches are more or less the same idea but SVN was specifically designed to fix long standing flaws in CVS so, in theory at least, SVN will always be the better choice.



Share Improve this answer

answered Aug 4, 2008 at 14:14

Follow



Mat

6,705 ● 7 ● 36 ● 39



4

I'll second Eridius' suggestion of Git, but I'd expand it to the other DRCS (Distributed Revision Control System) such as [Mercurial](#) and [bazaar](#).



These products are fairly recent and the level of tooling and integration with them seems low at the moment (based on my initial research). I'd say they were best suited to the power-developers out there (and on here ;-)).

On the other hand, what **doesn't** CVS currently do for you? From your initial question, you don't really have any, "CVS sucks at this, what could I use instead?"

You've gotta weigh up the costs of any potential migration against the benefits. For an existing project, I think that it would be hard to justify.

Share Improve this answer

Follow

answered Aug 18, 2008 at 17:04



Steven Dick

301 ● 2 ● 5



4

One thing not to overlook is ecosystem. I was working at a CVSNT shop, and I was finding more and more open source tools supported SubVersion by default.



Share Improve this answer

answered Aug 18, 2008 at 17:22



Follow



engtech

2,791 ● 3 ● 20 ● 24



btw: CVSNT supports atomic commits

2

Share Improve this answer

answered Aug 18, 2008 at 16:22

Follow



David Sykes

49.7k ● 18 ● 74 ● 81



2

As someone who is in the middle of switching between CVS and SVN (initially we switched all of our projects with cvs2svn and then decided that we would transition by only using svn on new projects), here are some of the problems we have had.



- Merging and branching are very different, and if you branch and merge frequently, unless you have SVN 1.5 running on your server have to know when you branched (this isn't very clear in the Tortoise SVN dialogs). Michael says the branching and merging is intuitive, I would argue that after using CVS for 10 years, it is not.
- If your are running the SVN server on Linux, it may be hard to get your SA to move to svn 1.5, as the default install 1.4.x.

- Merging conflicts is not nearly as easy or as clear (at least to me and my co-workers) in TortoiseSVN as it is in TortoiseCVS. The three pane approach takes some getting used to and the WinMerge (my preferred merge tool) doesn't do a three pane merge.
- Beware: many of the online tutorials and magazine articles I have read obviously don't branch and merge, you should set up your main repository as <https://svn.yoursvnserver.com/repos/YourProject/Trunk> and branches on <https://svn.yoursvnserver.com/repos/YourProject/Branches/BranchX> . You can clean up if you start your repos in the wrong place, but it leads to confusion.

Share Improve this answer

Follow

answered Sep 4, 2008 at 17:44



**Kris Erickson**

**33.8k** ● 26 ● 121 ● 179

---





1



You should take a look at [Git](#) instead of SVN. It's a DVCS that's blazing-fast and very powerful. It's not as user-friendly as SVN, but it's improving in that regard, and it's not *that* hard to learn.

Share Improve this answer

answered Aug 6, 2008 at 23:09

Follow



[Lily Ballard](#)

185k ● 34 ● 388 ● 349

---

You, sire, deserve a medal for fortune telling. – [altinturk](#) Nov 16, 2017 at 9:05

---



1



CVS (Concurrent Versions System) and SVN (SubVersion) are two version control file systems that are popularly used by teams who are collaborating on a single project. These systems allow the collaborators to keep track of the changes that are made and know who is developing which and whether a branch should be applied to the main trunk or not. CVS is the much older of the two and it has been the standard collaboration tool for a lot of people. SVN is much newer and it introduces a lot of improvements to address the demands of most people.

Share Improve this answer

answered Jul 17, 2018 at 7:19

Follow



[vidy](#)

1,630 ● 6 ● 27 ● 43



0

you might also choose to migrate only the latest code from CVS into SVN and freeze your current CVS repo. this will make migration easier and you might also build your legacy releases in the old CVS repo.



Share Improve this answer

answered Oct 8, 2008 at 16:19



Follow



[webwesen](#)

1,262 ● 4 ● 17 ● 31



0

Well, a few things which i feel makes svn awesome.



1. The SVN-Altassian crucible combination is a far superior method of reviews and quality checks
2. Better management of conflicts and merges
3. It's obviously faster for taking checkouts, performing commits, etc.
4. The atomic commit problem - It is possible that 2 people committing together in CVS can conflict each other, losing some data and putting your code base in an inconsistent state



Migration can easily be done in a few hours using cvs2svn.

Share Improve this answer

answered Sep 18, 2016 at 14:01

Follow



[richie](#)

11