What are the best practices for moving between version control systems?

4

There are about 200 projects in cvs and at least 100 projects in vss. Some are inactive code in maintenance mode. Some are legacy apps. Some are old apps no longer in use. About 10% are in active development. The plan is to move everything to perforce my end of year 2009.

Has anyone done a large migration like this?

Has anyone come across best practices for moving from cvs to perforce? Or a similar migration. Any gotchas to look out for?

version-control   cvs   perforce

Share

Improve this question

Follow

asked Sep 17, 2008 at 1:37

sal
23.6k ● 15 ● 69 ● 85

6 Answers

Sorted by: Highest score (default) ⬍

▲

5

▼

🔖

✔

↺

On the VSS side, there are conversion tools that are available to help with migration. They can mostly maintain version history (there are caveats that are explained in the readme and docs). I have migrated well over 50 VSS projcts into perforce using the VSS to perforce tool. Getting the data out of VSS can be a bit finicky and not terribly speedy, but it works. If you have direct access to the disks (i.e. not over a network share) to the VSS repository, the conversion can go much quicker. You can find information about the scripts [here](#).

There is a simlar page for CVS to perforce conversion [here](#), although I don't have direct experience with that. These links are good places to start. You can also search through the Perforce mailing lists at the Perforce Knowledge Base located [here](#). I'm pretty sure that you might find some conversion information in the mailing list archives.

Migrate your old projects first. You can make sure that your process works. When we migrated active code to Perforce, I took one weekend and basically took down access to the servers and moved the code over to Perforce. Honestly, it was a pretty easy migration and when people came back on Monday they were ready to go. You might think about preparing your employees with Perforce cheat sheets after you start doing the migration.

The biggest gotchas might actually be preparing your people to use Perforce. Had I done it over again, I would have migrated our smaller active projects first and prepared smaller numbers of people to use Perforce at once. As it was, I had to train 120+ people on day 1 after the migration and that was a bit much. Also, make sure that you don't have 100+ people hitting your server for a fresh sync on day 1 either. We wound up taking our server down multiple times during the first few days. We used a windows 32 bit server which I would not recommend. We have a windows 64bit server now and it's much more robust. If you can, I would actually use Linux as your OS for your perforce server. Again, there should be good info on the Perforce site about performance.

Share Improve this answer

edited Sep 17, 2008 at 3:05

Follow

answered Sep 17, 2008 at 2:59

Mark
10.2k ● 3 ● 40 ● 41

▲

2

▼

🔖

I haven't had to do something of this scale, but I have a few ideas. First off, start by taking a small, unimportant project, and migrate that. That will give you an idea of how much trouble it is going to take to migrate the rest of the projects. Immediately after that you should choose a medium size project as there may be issues with

migrating a larger project (say with branches) that might not be apparent on a small project.

Make sure you spend a bit of time seeing how easy it is to convert cvs projects to vss, or the other way around. If converting from vss to perforce is a real pain, you can convert vss to cvs, and then to perforce. Don't sink days into it, but it could back you out of a sticky situation. I think the key here is go incremental.

Backups are good. Period.

Consider a cutoff date, and any projects that are inactive, and older then that, should be mothballed. Check out the final revision and store that in Perforce. Do you really need 15 yearold visual basic code?

Share Improve this answer

Follow

answered Sep 17, 2008 at 2:18

Jonathan Arkell
10.8k ● 2 ● 26 ● 32

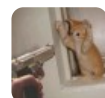What ever you do, keep the old repositories in read-only mode some where.

Share Improve this answer

Follow

answered Sep 17, 2008 at 3:56

Kevin Sheffield
3,616 ● 2 ● 25 ● 22

0

Forgive my answering a question with a question, but doesn't [Perforce](#) provide tools for this? Or, at the very least, documentation? I'd be beating up my Perforce salesperson...

Share Improve this answer

Follow

answered Sep 17, 2008 at 2:14

Kevin Little
12.9k ● 6 ● 42 ● 48

> the p4 tools will do the technical part of the migration. But there 100s of projects used by dozens of teams in multiple locations. A cold-turkey switch is too risky. We need to test IDE, command line and automated tool access. We also need to make sure no code is lost. – sal Sep 17, 2008 at 2:26

0

Consider not migrating dead and inactive projects. Simply put their repositories in read-only mode. The data will still be available if needed and you save the time effort of migrating them. Just migrate the 10% that are in use. Document the process thoroughly.

If one of the un-migrated projects gets resurrected some time in the future you can easily migrate it using your documentation as a reference.

Share Improve this answer

Follow

answered Sep 17, 2008 at 3:51

Matthew Jaskula
1,306 ● 1 ● 16 ● 16

0

We migrated our svn repository with a tool that we wrote, and just took the head revision of our starteam projects.

Watch out for differences between single-file checkins (CVS) and multi-file changesets (Perforce).

Watch out for branches is separate space (CVS) vs. branches in filepath-space (Perforce).

Share   Improve this answer

Follow

answered Sep 17, 2008 at 7:14

Douglas Leeder
53.3k ● 9 ● 99 ● 138

---

do you have link? – user765443 Jun 18, 2017 at 5:13

I'm afraid other people did the migration a long time ago and probably didn't save the tools, or commit them to perforce – Douglas Leeder Jun 18, 2017 at 9:16

Thanks. I am working cvs vs perforce with advantage and disadvantage as I have 2TB data. Do you have any link or info – user765443 Jun 18, 2017 at 15:01