How do you manage .vcproj files in source control which are changed by multiple developers?

Asked 16 years, 2 months ago Modified 7 years, 7 months ago

Viewed 3k times

8

We use Subversion as our source control system and store the VisualStudio project files (vcproj) in the source control system as is normal I think. With Subversion we don't use any form of file locking, so if two developers are working on the same project at the same time and both add files to the project, or change settings, the second one to commit has to merge the changes.

How do you merge these changes?

The vcproj files are just text files so it is possible to edit them by hand but they are not very amenable to hand editing, especially by junior developers.

The ways I can think of are

- Get the latest version from svn and re-add all local changes manually
- Edit the file by hand to resolve any conflicts from an automatic merge

- Implement some form of locking scheme to prevent simultaneous changes
- Have an agreement between developers so they do not make simultaneous changes

Currently we are using the first option of re-adding all changes manually but this is time consuming and I was wondering if there is a better way.

With source files the automatic merge feature works most of the time and we don't get many conflicts.

visual-studio **svn** **merge**

Share

Improve this question

Follow

edited Apr 27, 2017 at 17:29

Cœur

38.6k ● 26 ● 202 ● 276

asked Oct 3, 2008 at 13:18

David Dibben

18.7k ● 6 ● 43 ● 41

You may find that if you're having frequent check-in collisions, you could benefit by changing how your project tree is organized. IE: instead of having one .vcproj with 500 files in it, have 1 project with 50 files that links to 9 libraries with 50 files each. If you find you *can't* split your project into separate libraries, it's possible that spending the time to make that possible will in the end increase developer efficiency, because modularity helps with more than just project file maintenance. – KeyserSoze Oct 23, 2010 at 6:17

7 Answers

Sorted by: Highest score (default)

4

I've found that option 2 (edit the files by hand) generally works fairly well, as long as you're using a good diff tool (I use WinMerge). The main problem I've run into is that Visual Studio will sometimes reorder the file. But, if you have a good diff/merge tool then it should be able to differentiate between changed content and moved content. That can help a lot.

Share Improve this answer Follow

answered Oct 3, 2008 at 13:22

Herms

38.7k ● 13 ● 79 ● 104

This is sad. 2010 and VCPROJ files have to be maintained by hand so the revision control system doesn't whack it out. – dwj Jan 14, 2010 at 20:08

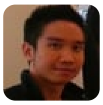
Agreed, +1. 95% of the time, the changes are very small (such as adding or deleting 1 or 2 files from the project) and will merge flawlessly. As long as you're not making grand, sweeping changes to the project file and you look carefully at the merged result before auto-accepting it, you'll be just fine. In the rare cases where the merge doesn't work, you should remake your local changes. – Adam Rosenfield Oct 23, 2010 at 6:05

3

You might find Project: Merge or Tools for SLN file useful

Share Improve this answer Follow

answered Oct 23, 2010 at 5:41



sivabudh
32.6k ● 65 ● 166 ● 231

1

This is a tough problem and I think a weakness in the Visual Studio architecture. The way we found round it was to not have the proj files in source control at all and to have a build script that handled the configuration settings.

The alternative was very messy and we could not guarantee consistent builds or environments between developers. This led to a huge number of downstream integration problems and eventually we took the draconian step of removing the project files from source control.

The developers environments could still become misaligned but it showed up when they tried to build things themselves.

Share Improve this answer

Follow

answered Oct 3, 2008 at 13:26

Simon

80.6k ● 26 ● 92 ● 119

1

Using TFS here, but I don't think it makes a difference. We also don't lock, and sometimes have to deal with merging project files. I've never found it to be that complex or much of an issue. Rarely do we ever experience issues that can't be merged automatically, and the manual merge process is pretty much trivial.

There's only one caveat to this: Check in often! If you make major changes to the project structure and don't check them in immediately those changes can start compounding the complexity of later merges. If I make a major change to the structure of a project, I usually give everybody a heads up. I'll ask them all to check in their current work, and then take care of the merge myself.

Share Improve this answer

Follow

answered Oct 3, 2008 at 13:43

user1228

1

I found this recently: http://www.codeproject.com/KB/macros/vcproj_formatter.aspx If you run this tool on a vcproj file and on a modified version of it then you can merge them together easily with your favorite text merge tool, and in addition the result is a more compact pretty vcproj file.

Share Improve this answer Follow

answered Dec 5, 2010 at 22:12

Valerius

11 ● 1

0

Options 1 and 2 are not mutually exclusive - if the developer is junior level, let them use option 1 (re-get the project file and re-do the changes) if that's more comfortable for them. For more senior developers, option 2 (merge using a merge tool) is perfectly fine.

I think this is a situation that currently has no magic bullet - sometimes merging is a pain.

Share Improve this answer Follow

answered Oct 4, 2008 at 0:02

Michael Burr

340k ● 52 ● 548 ● 769

-1

We use a diff tool (WinMerge) to merge changes. The project files are (for the most part) really straight-forward XML. The key here, though, is that there never should be any surprises when merging, because good communication is part of the bed-rock of effective source control.

Simultaneous changes *to the project* are perfectly fine as long as people communicate.

Share Improve this answer Follow

edited Mar 4, 2009 at 21:50

answered Oct 3, 2008 at 13:23

Bob King
25.8k ● 7 ● 57 ● 66

Heh. My brain did a small substitution on your last sentence there. "Simultaneous changes are perfectly fine...as long as I check in first". :) – Herms Oct 3, 2008 at 13:31

if you get above 10 developers or work across several locations or tie zones this is simply not reliable enough to bet your development on. – Simon Oct 3, 2008 at 13:32

I disagree completely. If you're having problems *then you are doing something* wrong. This is precisely what source control should solve! – Bob King Oct 3, 2008 at 13:32