

Solution deployment, CM, InstallShield

Asked 16 years, 4 months ago Modified 13 years ago Viewed 558 times



People,

0



We have 4 or 5 utilities that work in conjunction with our application. These utilities are either .bat files, or VB apps, PowerBuilder, etc. I am trying to manage these utils in source control, and am trying to figure out a better way to assign versions to them. Right now, the developers use the version control's meta-data -- specifically label -- to store the version number of the tool.

My goal is to have individual InstallShield packages for each utility, and an easy means to manage and assign version numbers to these packages.

Would you recommend a separate .ini file with the info, or store the info in InstallShield .ism file itself, or just use the meta-data info from version control tool?

UPDATE:

I like the idea Orion. I have one concern though. The script that increments the version number... it can not be intelligent enough to increment Major number etc. right. e.g. if one of the utils has version 1.2.3 and we are at a

point where the new version is 2.0.0. The script may not be able to handle this.

I think this has to do a lot with our branching techniques -- we don't have any. The folks thought since the utils are so small, the source may not need branches.

content-management-system

versioning

installshield

packaging

Share

Improve this question

Follow

edited Dec 17, 2011 at 4:20



T.Rob

31.8k ● 9 ● 63 ● 105

asked Aug 20, 2008 at 13:31



Ravi

3 Answers

Sorted by: Highest score (default)



1

PowerBuilder in particular has a nice trick you can do to incorporate the build number from an ini file into the compiled application.



Details here: <http://www.pbdr.com/pbtips/ex/autorev.htm>



We have ini file inside source control that stores the build number and its value is used in our build scripts to determine what label to apply to the source tree after a

successful build. Works very nicely for our needs. When we branch, we do have to manually kick the file to increment the proper number though.

Share Improve this answer

answered Sep 2, 2008 at 19:50

Follow



Doug Porter

7,867 ● 4 ● 41 ● 55



0



I managed our build system at my last job, which seemed to have some parallels to what you're asking.

There were ~30 C++ projects which needed compiling, and various .NET/Java things, and the odd perl script.

This was all built on our build machine using NAnt - If I were doing it today I'd use [rake](#), but the idea is the same.

We basically had an auto-incrementing build number which was stored in a version.txt file in the root of the repository.

Each time we did a build (automatically done each night, or also on-demand if neccessary) the script would increment this number and check the file back into source control.

All the other apps referenced this file for their version number, or for things which didn't support working like this, the script would set environment variables or perform other workarounds

- I'm pretty sure that our installshield programs referenced an environment variable for their version number, but we deprecated them in favour of [wix](#) as installshield really did suck
- in the case of visual studio, grep/replace the number within the .csproj files, and check them back in

Hope this gives you some ideas

Share Improve this answer

answered Aug 21, 2008 at 1:58

Follow



[Orion Edwards](#)

123k ● 66 ● 245 ● 339



0



Using the meta data from your version control system should keep things simpler. It's how your developers already use the system. There is no additional file to maintain. My personal experience has taught me to version the satellite applications with the same as version as the main app. K.I.S.S



Share Improve this answer

answered Sep 20, 2008 at 16:41

Follow



[Aaron Fischer](#)

21.2k ● 18 ● 78 ● 117