# Comparing VB6.exes

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

Viewed 728 times

2

We're going through a massive migration project at the minute and trying to validate the code that is deployed to the live estate matches the code we have in source control.

Obviously the .net code is easy to compare because we can disassemble. I don't believe this is possible in vb6 exes because of the manner of compilation.

Does anyone have any ideas on how I could validate the source code and the compiled executable matches the file I have in Live.

Thanks

vb6 migration compare

Share

Improve this question

Follow

asked Oct 6, 2008 at 15:16

Lee Gathercole

115 ● 1 ● 7

Sounds like you need better source control and/or policies.
– GEOCHET Oct 6, 2008 at 15:21

## 6 Answers

Sorted by: Highest score (default)

3

Visual Basic had (has) two ways of compiling, one to the interpreter ( called P-code) that would result in smaller binaries, and a second one that generates "regular" windows .exe file (called native) that was introduced because it was supposed to be fastar than p-code; although the compiled file size increased with this option. If your compilation was using p-code, it is in theory possible to restore the sources.

Either way is pretty difficult to do, but there are tools that claim they can partially do this, one that I know of ( never tried but there is a trial version ) is VB-decompiler http://www.vb-decompiler.org/

Share Improve this answer

Follow

answered Oct 6, 2008 at 15:36

AlePani

519 ● 2 ● 7

> Great tool. This was never going to be a perfect science. What we're doing is decompiling with the exe and saving the project files. We are then using the Beyond Compare tool to make the comparison. Not perfect, but it's a start. Thanks – Lee Gathercole Oct 7, 2008 at 11:00

0

Unfortunately that's almost impossible. Bear in mind that VB6 code compiled on different machines will have different exe sizes and deployment requirements.

This is why the old VB'ers had a dedicated machine to compile their code.

Share Improve this answer

Follow

answered Oct 6, 2008 at 15:18

ChrisLively

88k • 27 • 173 • 247

---

0

This won't help you with already deployed items, but if you upped the revision number on every compile (there is a project setting to do this for you automatically) then you could easily compare version numbers.

Share Improve this answer

Follow

answered Oct 6, 2008 at 15:37

Matt Dawdy

19.7k • 19 • 67 • 92

---

0

My old company bought a copy of that VB-Decompiler and as noted before VB5/6 generates P-Code extra, that tool did produce some code and if not Assembly code which could be "read".

Share Improve this answer

Follow

answered Oct 6, 2008 at 15:41

Sarkie

272 • 3 • 18

---

0

If you have all the code you compiled, you could compare the CRC's of that code to what is deployed in the field. But if you don't have the original compiled code,

depending upon how you compiled the code you (if you used P-Code rather than Native Code you may be able to disassemble but the disassembly will look nothing like your source code). I doubt you would have shipped the PDB's with the exe's, but if you did, you could certainly use those to compare with the source code in your repository.

Share Improve this answer

Follow

answered Oct 6, 2008 at 17:22

Kris Erickson

33.8k ● 26 ● 121 ● 179

---

0

Have a trusted computer that can check out the various libraries and exes you make and compile them automatically. Keep those in a read-only but accessible location. Then do a binary comparison between the deployed site and your comparison site.

However I am not sure of the logic over disassembling the the complied units. My company and most other places I know of use a combination of a build computer and unit testing. In our company the EXE we make is a very thin shell over a bunch of libraries. For example a button click will be passed to a UI Active X DLL that does the actual processing. What we do after a build is run a special EXE that perform our list of unit test. If they all passed we know our libraries, where 90% of our code is, are good. As for the actual EXE we have a hand procedure that takes about two hours to do and then we are good. IIt is rare for any errors to happen in the EXE.

Share Improve this answer

Follow

answered Oct 6, 2008 at 19:47


RS Conley

**7,206** ● 2 ● 22 ● 37