

What is the best solution for maintaining backup and revision control on live websites?

Asked 16 years, 4 months ago Modified 10 years, 3 months ago

Viewed 1k times



What is the best solution for maintaining backup and revision control on live websites?

3



As part of my job I work with several live websites. We need an efficient means of maintaining backups of the live folders over time. Additionally, updating these sites can be a pain, especially if a change happens to break in the live environment for whatever reason.



What would be ideal would be hassle-free source control. I implemented SVN for a while which was great as a semi-solution for backup as well as revision control (easy reversion of temporary or breaking changes) etc.

Unfortunately SVN places .SVN hidden directories everywhere which cause problems, especially when other developers make folder structure changes or copy/move website directories. I've heard the argument that this is a matter of education etc. but the approach taken by SVN is simply not a practical solution for us.

I am thinking that maybe an incremental backup solution may be better.

Other possibilities include:

1. [SVK](#), which is command-line only which becomes a problem. Besides, I am unsure on how appropriate this would be.
2. [Mercurial](#), perhaps with some triggers to hide the distributed component which is not required in this case and would be unnecessarily complicated for other developers.

I experimented briefly with Mercurial but couldn't find a nice way to have the repository separate and kept constantly in-sync with the live folder working copy. Maybe as a source control solution (making repository and live folder the same place) combined with another backup solution this could be the way to go.

One downside of Mercurial is that it doesn't place empty folders under source control which is problematic for websites which often have empty folders as placeholder locations for file uploads etc.

3. [Rsync](#), which I haven't really investigated.

I'd really appreciate your advice on the best way to maintain backups of live websites, ideally with an easy means of retrieving past versions quickly.

Answer replies:

- *@Kibbee:*
 - It's not so much about education as no familiarity with anything but VSS and a lack of time/effort to learn anything else.
 - The xcopy/7-zip approach sounds reasonable I guess but it would quickly take up a lot of room right?
 - As far as source control, I think I'd like the source control to just say that "this is the state of the folder now, I'll deal with that and if I can't match stuff up that's your fault, I'll just start new histories" rather than fail hard.
- *@Steve M:*
 - Yeah that's a nicer way of doing it but would require a significant cultural change. Having said that I very much like this approach.
- *@mk:*
 - Nice, I didn't think about using Rsync to deploy. Does this only upload the differences? Overwriting the entire live directory everytime we make a change would be problematic due to site downtime.

I am still curious to see if there are any more traditional options

version-control

backup

hosting

Share

Improve this question

Follow

edited Sep 15, 2014 at 11:24



Martijn Pieters

1.1m ● 319 ● 4.2k ● 3.4k

asked Aug 26, 2008 at 2:32



Matt Mitchell

41.8k ● 35 ● 121 ● 185

4 Answers

Sorted by:

Highest score (default)



4



You can still use SVN, but instead of doing a checkout on your live environment, do an export, that way no .svn directories will be created. The downside, of course, is that no code changes on your live environment can take place. This is a good thing.

As a general rule, code changes on production systems should never be allowed. The change should be made and tested in a development/test/UAT environment, then once confirmed as OK, you can tag that code in SVN with something like RELEASE-x-x-x. Then, on the live system, export the code with that tag.

Share Improve this answer

Follow

answered Aug 26, 2008 at 2:37



Steve M

10.6k ● 12 ● 53 ● 63



We use option 3. Rsync. I wrote a bash script to do this along with some extra checking, but here are the basics

2

of what it does.



1. Make a tag for pushing to live.
2. Run svn export on that tag.
3. rsync to live.



So far it has been working out. We don't have to worry about user conflicts or have a separate user for running svn up on the production machine.

Share Improve this answer

answered Aug 26, 2008 at 2:49

Follow



[mk.](#)

26.2k ● 13 ● 39 ● 41

hello can you share that bash script, i am trying to implement a same approach. thanks a lot. – [Gabriel Solomon](#) Mar 9, 2009 at 11:15



1



Any source control solution you pick is going to have problems if people are moving, deleting, or adding files and not telling the source control system about it. I'm not aware of any source control item that could solve this problem.



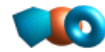
In the case where you just can't educate the people working on the project[1], then you may just have to go with daily snapshots. Something as simple as batch file using xcopy to a network drive, and possibly 7-zip on the command line to compress it so it doesn't take up too much space would probably be the simplest solution.

[1] I would highly disbelieve this, probably just more a case of people being too stubborn and not willing to learn, or do "extra work". Nevermind how much time source control could save them when they have to go back to previous versions, or 2 people have edited the same file.

Share Improve this answer

answered Aug 26, 2008 at 2:37

Follow



Kibbee

66.1k ● 28 ● 144 ● 184



1

rsync will only upload the differences. I haven't personally used it, but [Mark Pilgrim](#) wrote a long time ago about how it even [handles binary diffs brilliantly](#).



svn+rsync sounds like a fantastic solution. I'll have to try that in the future.



Share Improve this answer

answered Aug 29, 2008 at 22:40

Follow



cpm

1,543 ● 10 ● 16