

Re: VSS best practice for promoting code to production?

Source:

<http://www.tech-archive.net/Archive/VisualStudio/microsoft.public.vstudio.sourcesafe/2007-10/msg00005.html>

- *From:* Mike <TechGuy@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 08 Oct 2007 09:40:08 -0700
-

I'll tell you what I've seen done at some companies I've worked at, then I'll tell you how I think we SHOULD be doing it. But first, you've used a lot of TLAs and FLAs (Three Letter Acronyms and Four Letter Acronyms), not all of which I'm familiar and some have too many uses that I can't be sure which you mean (and I don't have the time to study each definition to find the right context):

CIFS has at least 7 definitions (OK, I can figure which one you mean here :)

<http://www.acronymfinder.com/af-query.asp?Acronym=cifs&string=exact>

MSB has at least 37 definitions:

<http://www.acronymfinder.com/af-query.asp?Acronym=msb&Find=find&string=exact>

SMB has at least 33 definitions:

<http://www.acronymfinder.com/af-query.asp?Acronym=smb&string=exact>

So, I'll refrain from pretending that I'm smarter than I am by not pretending that I know what you're talking about in your TLAs and instead focus on the primary gist of your question regarding best practices for moving code from VSS to production:

Here's what I've seen in real life practice:

1. When we've got code that's a release candidate, we developers might label it in VSS, then we will move it from our dev environment to our staging environment (done by a SysAdmin, not a developer, for security reasons).
2. After the testers shake it out and determine that it's production ready, we invoke our SysAdmins again to move it from staging to production.
3. The developer may (or might not) label it again as a production release in VSS.

In short, actual practice at companies I've worked at is that only the developers actually touch VSS. They then label it (if they feel so inclined) as it moves along the path to production. Some developers

Re: VSS best practice for promoting code to production?

don't even check it in on each release. I've not worked at a company that has strict rules on this, but I think it should be a requirement that whenever you release anything to staging or production, you should, at the very least, check it in and label it as such.

Here's what I recommend:

First, the run-time environments:

There should be at least 3 run-time environments:

1. Development Environment. This can be either the developer's own workstation or an environment that duplicates production's environment. The developer uses this day in and day out during the development process. Code begins here.
2. Staging/Testing/QA Environment: Code is moved from dev to staging for user acceptance and testing.
3. Production: Tested and approved code moves from staging to here where it will live it's life.

Second, the VSS environments:

1. One for developers.
 - Read/Write access by devs.
 - Read Only by SysAdmins.
2. One for SysAdmins.
 - Read/Write by SysAdmins.
 - Read Only by devs (if at all).

I'll explain these later.

Third, a security model: There should be a security wall between developers and production and between the developers VSS and the SysAdmin's VSS. This means that developers should not have direct access to anything in production. This helps prevent dev's with a grudge or ill intent of any kind from inserting malicious code into production (I've seen this happen many times, usually as a back door). Of course, this won't totally block it, but it helps. This also protects developers (which is why I'm so in favor of it) from blame if the code in production is different than what was tested.

Security on each run-time environment:

- Development: Developer(s) have total access.
- Staging: SysAdmins have access and developers are blocked (read only would seem to be OK, but they shouldn't see the DB connection strings).
- Production: SysAdmins have access and developers are blocked.

The flow of coding goes like this:

1. Developer checks out latest code and writes code and tests in development environment. When the developer is ready for users to test it, he checks it into the developers VSS db, labels it, and submits the VSS labeled version to the SysAdmin(s) to move it into staging. This is the only point in which anyone other than the developer

Re: VSS best practice for promoting code to production?

touches the developer's VSS DB.

2. The SysAdmin performs a get version from the developer's VSS. He'll also check out the latest version from the SysAdmin's VSS database (not to get the files from it, but to make the project "checked out"). Then he checks into the SysAdmin's VSS database what he just got out of the developer's VSS db. Depending on the complexity of the project and the amount of changes, he may do a version comparison. Whether he does it or not, the developer knows he can and is less likely to insert malicious code.
3. The SysAdmin then publishes that code into the staging environment and the developer notifies his testers of the new code to be tested.
4. After the testers have accepted the new changes, notification is sent to the SysAdmin that the staging code is ready for production. At that point, the SysAdmin simply moves the code from staging to production and re-label's the SysAdmin's VSS DB project. Optionally, the SysAdmin can inform the developer that the code is in production and the developer can relabel his VSS db project as well.

This is basically a collection of the best of the best practices I've seen in the various companies I've worked for with a few sprinkles of some of my own suggestions.

In short, I would NEVER move code directly from a developer into production, which means not moving directly from the developer's VSS straight into production. There should always be a testing/evaluation process on the developer's code before it makes it to production. In other words, never take any code from a developer at face value. And I say that as a developer. I want others signing off on the code.

Hope that helps!

Owner/Operator of
www.MichaelsAttic.com

On Oct 5, 12:49 pm, David <Da...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

Hi all –

I have a question. We are starting to build out our ecommerce and web services offerings. I am the admin responsible for all of our servers, and I gotta say, right now our process for promoting code to production is a little silly. The developers will make a copy of thier code (asp, asp.net, whatever) and email the files to thier manager who will then access a CIFS share on the server to post the files.

Not to be a control freak, but I am really not comfortable with this. I would like to see us improve this process, so I am curious how some of you might handle this in your shops? I think about having a specific "promotion" user in the VSS system so that we can check out the latest files right to the production systems? But we are on VSS 6, which uses standard SMB/CIFS shares to access, right? I just hate to poke MSB/CIFS holes in our firewall – even

Re: VSS best practice for promoting code to production?

if it is just from our dmz to our private lan...

Any thoughts or advice is greatly appreciated!

David