

Parabuild user transitioning to...



[Dave Kap](#) 27 posts since

Mar 3, 2010

Considering TeamCity is developing in a direction I like, while Parabuild is barely developing, I'd like to switch over to using TeamCity as my automated build system of choice. However, in testing, I've found a few problems and I was hoping I could get some answers to the questions that have cropped up. Before I begin, please note that my build farm is about 30 agents strong (10 WinXP, 5 Mac, 15 Linux) and I would often have about 20-30 different configurations (per platform) active at any one time, most of which are connecting to a Mercurial repository with dozens of branches.

1. It seems like the VCS Root connector requires the TeamCity server to do an hg identify of any repo I try to configure before it agrees to let any agents run the checkout. This is causing problems where my master needs ssh keys that my agents already have. The fix could be as simple as "give the master the keys" but I don't see why I can't completely skip the step of the master trying to have a checkout directory and run an hg command when it already has agents allotted to the task. Is there a way to skip this requirement?
2. I love the fact that I can use variables to set up my configurations to do things like, using the configuration name, determine a directory for a branch to be checked out to. This is made even easier with the configuration template model given. However, I do not understand why I cannot use these same variables within the VCS Root configurations. It's strange to me that it can be so easy to make a new configuration from a template and name it, but I then have to go and create a new VCS Root in order for it to end up doing what I want it to do. The optimal process should be: 1. Copy template 2. Rename appropriately 3. Done. Such that the name of the configuration would be enough to determine where the repo is located and which branch to check out from it. This kind of ease is one of the few reasons I'm still hanging on to Parabuild.
3. I'm fairly certain the log isn't really displaying everything that's happening. For example, I currently have a hanging bug when I try to use one of our Mac agents to clone over http. If I do this manually from the Mac console, I can see exactly where it is hanging. However, if I do this in TeamCity, I only see a line being given from TeamCity rather than the straight stdout. Am I missing something here?

Thanks for fielding my questions!

Tags: vcs, mercurial, question



[Yegor Yarko](#) 1,459 posts since

May 5, 2004 1. **Re: Parabuild user transitioning to TeamCity has a few questions.** Mar 18, 2010 5:32 PM

Dave,

Parabuild user transitioning to...

Welcome to TeamCity! 

1. It seems like the VCS Root connector requires the TeamCity server to do an hg identify of any repo I try to configure...

If I understand you right, you ask why the server needs to talk to Mercurial repository. This is because TeamCity gathers changes information (authors, files, comments) on the side of the server. This is essential for the features like displaying pending changes, viewing file diff in the web, ability to run a build on a selected change, etc. This is a core requirement and it cannot be workarounded if you use TeamCity VCS integration capabilities.

2. I love the fact that I can use variables to set up my configurations to do things like, using the configuration name, determine a directory for a branch to be checked out to. This is made even easier with the configuration template model given. However, I do not understand why I cannot use these same variables within the VCS Root configurations...

We have a [feature request](#) on this. Feel free to add your vote. This is not yet implemented because TeamCity watches the VCS roots continuously and making the VCS root change at arbitrary points does not scale well. However we are considering the feature for implementation.

3. I'm fairly certain the log isn't really displaying everything that's happening. For example, I currently have a hanging bug when I try to use one of our Mac agents to clone over http. If I do this manually from the Mac console, I can see exactly where it is hanging. However, if I do this in TeamCity, I only see a line being given from TeamCity rather than the straight stdout. Am I missing something here?

The build log is focused on the build activity while the details of VCS-related activity are usually hidden from the user. See logs/teamcity-vcs.log (on the server if you use server-side checkout and on agent otherwise) to get more details on what's happening. If you believe there is a bug or we can improve the progress reporting, feel free to create an issue in our [tracker](#) with appropriate details, attached logs, etc.

BTW, if you can sum up and publish differences between Parabuild and TeamCity after some time using TeamCity, that can be a great resource for users considering different options while choosing their CI server.

Parabuild user transitioning to...



[Dave Kap](#) 27 posts since

Mar 3, 2010 2. **Re: Parabuild user transitioning to TeamCity has a few questions.** Mar 18, 2010 10:07 PM

Thanks for the quick response! 😊

If I understand you right, you ask why the server needs to talk to Mercurial repository. This is because TeamCity gathers changes information (authors, files, comments) on the side of the server. This is essential for the features like displaying pending changes, viewing file diff in the web, ability to run a build on a selected change, etc. This is a core requirement and it cannot be workarounded if you use TeamCity VCS integration capabilities.

Actually, this part makes total sense, so let me ask the question a different way. Is there any way to tell the server how it should connect to the Mercurial repository and make that a *separate* method of calling than the way I want my agent to call it? If this was possible, I'd be able to get the server to contact the repo via http like it should, while still allowing my Mac agent to contact the repo via ssh.

We have a [feature request](#) on this. Feel free to add your vote. This is not yet implemented because TeamCity watches the VCS roots continuously and making the VCS root change at arbitrary points does not scale well. However we are considering the feature for implementation.

Eh, I'm not holding out hope for a feature that was requested 2 years ago and has an answer that won't work with Mercurial repositories. In theory the use of checkout rules looks good, but when you try to do a VCS Root connection test against hg.example.com/branches it will fail, because only the repositories at hg.example.com/branches/repo1 and hg.example.com/branches/repo2 are seen by TeamCity as actual hg repositories. I suppose then my question has to be re-formed here as well. Is there a way to make the checkout rules do what I need for a Mercurial repository; whether it be by allowing the VCS Root to seem connectable to the repo's parent directory or by "peeling back" what it thinks it's trying to check out and redirect to a different, specified repository. I'll be sure to add to the feature request thread.

The build log is focused on the build activity while the details of VCS-related activity are usually hidden from the user. See logs/teamcity-vcs.log (on the server if you use server-side checkout and on agent otherwise) to get more details on what's happening. If you believe there is a bug or we can improve the progress reporting, feel free to create an issue in our [tracker](#) with appropriate details, attached logs, etc.

I will. It should be an option to see the direct stdout as the TC log IMHO.

Parabuild user transitioning to...

BTW, if you can sum up and publish differences between Parabuild and TeamCity after some time using TeamCity, that can be a great resource for users considering different options while choosing their CI server.

Possibly. I don't have much time on my hands these days so no guarantees; especially with my current inability to use VCS Roots the way I need. Does the fact that Mercurial is technically a TC plug-in means I should be discussing its problems over in the plug-in forum?

Just in case I don't have to ask there, I have one more question for here. How exactly does TC check against Mercurial repositories to see if there are any changes made? Under Parabuild (which does not natively support Mercurial but does support generic VCS manipulation) we have our own scripts which need to check against a repo's RSS feed in order to see if there are new changes made. One of the nice things about this is that the whole process is transparent to us, the not-so-nice thing being that we are the ones who needed to build and still maintain the scripts.

Thanks again for the fast response! Much appreciated.



Yegor Yarko 1,459 posts since

May 5, 2004 3. Re: Parabuild user transitioning to TeamCity has a few questions. Mar 20, 2010 12:11 AM

Dave,

Is there any way to tell the server how it should connect to the Mercurial repository and make that a *separate* method of calling than the way I want my agent to call it?

Unfortunately, no. Currently, TeamCity uses the same settings to reach for repositories from both server and agents. Feel free to file a feature request on this.

Is there a way to make the checkout rules do what I need for a Mercurial repository; whether it be by allowing the VCS Root to seem connectable to the repo's parent directory or by "peeling back" what it thinks it's trying to check out and redirect to a different, specified repository.

I am afraid that checkout rules won't probably help here. Their meaning (at least currently) is filtering of the changes, while the main operations are performed with the whole repository in mind (e.g. your repositories will probably have different sequence of revisions while VCS root in TeamCity should have a single revision meaning).

Parabuild user transitioning to...

However, we do consider using build configuration properties in a VCS root and if we figure out the technical bits, chances are the feature won't reach it's 2nd birthday in unfixed state :).

For now, it seems you need to use different VCS roots in different build configurations ([TeamCity 5.1 EAP](#) allows to attach new VCS roots to template-using build configurations).

Does the fact that Mercurial is technically a TC plug-in means I should be discussing its problems over in the plug-in forum?

No. The plugins forum is more for plugin writers and using TeamCity Java API.

How exactly does TC check against Mercurial repositories to see if there are any changes made?

The procedure is common for all VCS integrations in TeamCity. There is a Java plugin (it is actually [open-source](#)) that implements [VCS integration](#) TeamCity API. The plugin uses hg executable to perform all the main operations and stores .hg directories on the server.

General procedure is that the plugin can tell TeamCity the current repository version and can provide the list of changes between certain revisions on TeamCity request. Also, plugin can serve the sources either as full or incremental "patch".