Faradi 7.0 EAP (build 20184) Release Notes

This is the first EAP build of TeamCity 7.0 (code name Faradi).

Note: starting with version 7.0 TeamCity server and agent require Java 6.0 or later. Also TeamCity Ant build runner no longer supports Java 1.3 as JVM of the project, the minimum version is 1.4. As a workaround you can use command line runner.

Agent pools

Instead of having a single, common set of agents, you can now break it into separate parts called agent pools. A pool is a named set of agents to which you can assign some projects. An agent can belong to one pool only, but a project can use several pools for its builds.

Project builds can be run only on agents from pools assigned to the project. By default, all newly authorized agents are included into Default pool.

With the help of agent pools you can bind specific agents to specific projects. Also with agent pools it is easier to calculate required agents capacity.

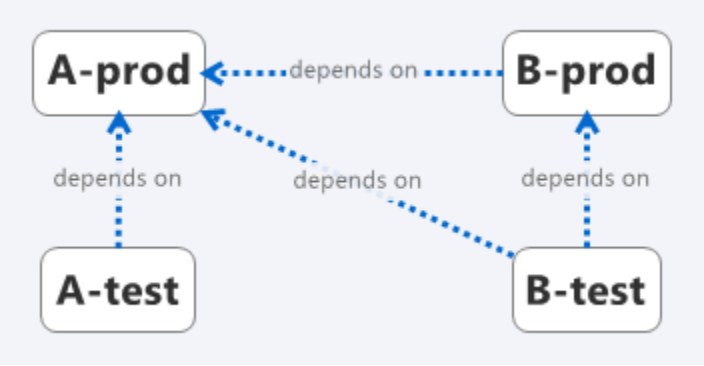
Build failure conditions

"Fail build if" settings of a build configuration were moved to a separate step. Also there are two useful additions:

- Ability to fail a build on a metric change (read more).
- Fail a build if a specific message (matching some regexp) is logged/not logged in the build log - this failure condition still has some open issues, they will be addressed in the next EAP.

Dependency based test run

Maven, Gradle and IntelliJ IDEA Project build runners now support dependency based run of tests. For example, say your project has these modules: A-prod, A-test, B-prod, B-test.
Modules with -prod suffix contain production code, while modules with -test suffix contain tests for corresponding production modules.



Now if a build starts with a change in module A-prod TeamCity agent will run tests in both modules A-test and B-test (because B-test depends on A-prod and can be affected by the change).
However, if a change was made in B-prod only, TeamCity will only run tests from B-test module.

So, in general, the more independent your modules - the better. It's a recommended way to design a software, but now you can get one more benefit from such approach: faster builds.

To enable this functionality for Gradle or IntelliJ IDEA project runners simply turn on "Run affected tests only (dependency based)" checkbox.

Note that since IntelliJ IDEA project runner operates with run configurations, instead of individual tests, if "Run affected tests only (dependency based)" checkbox is enabled, runner will execute run configurations depending on affected modules only.

Maven "Incremental building"

For Maven similar functionality is called "Incremental building" and has some additional settings. When incremental building is on, a Maven build is split into two phases - the preparation phase and the main phase - both separate Maven executions. The "main" phase is like a regular (not incremental) build in which main goals ("test", "verify", etc.) are executed with the only difference - they are executed only for modules affected by changes. The "preparation" phase is intended for building the dependecies of affected modules. Unfortunately these two phases couldn't be accomplished in a single Maven execution. Why is to be explained in detail later in a separate blog post. Until then it's quite easy to remember the following pattern:

1. if your tests are executed with the goal "test", then the preparation goal is "compile" (unless you have custom executions required for tests in affected modules and bound to later Maven lifecycle phases).
2. if you run both unit and integration tests with "verify", then the preparation goal is "package" (with the same considerations described above) with additional Maven argument "-Dmaven.test.skip=true" to avoid executing unit tests in the preparation phase.

NuGet support

This TeamCity EAP build comes with bundled NuGet Support plugin. We already announced availability of this plugin in a series of blog posts:

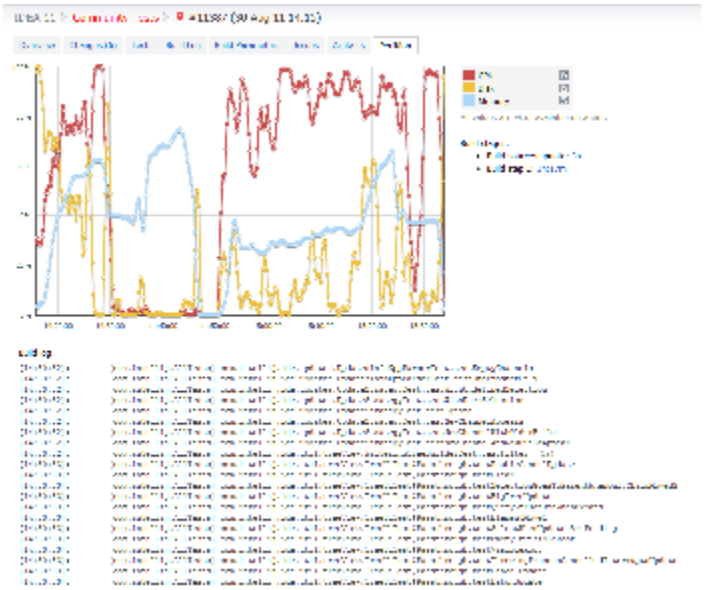
- http://blogs.jetbrains.com/dotnet/2011/08/native-nuget-support-in-teamcity/
- http://blogs.jetbrains.com/teamcity/2011/07/20/nuget-plugin/
- http://blog.jonnyzzz.name/2011/07/nuget-publish-build-runner.html

The plugin is also compatible with TeamCity 6.5, so if you want to use it in your existing production server, you can download it at teamcity.jetbrains.com.

Build performance monitor (experimental)

Each new build now has additional tab, called PerfMon. On this tab you can see CPU/Disk and Memory usages on the agent during the build. You can also click on a point in the chart and corresponding part of the build log will be shown.

For example, from the picture below it is clear that at some point CPU and Disk usage is very low. This lasts for about 20 minutes. Seems like the tests executing at this time need some investigation, probably, most of the time they are blocked on some lock or wait for some event:



Performance monitor supports Windows, Linux and Solaris operating systems. Note that performance monitor reports the load of the whole operating system. It will not report proper results if you have more than one agent running on the same host, or agent and server installed on the same machine.

Build log

Build log now has collapse all and expand all links in the tree view which now also works for builds that are still running. Moreover you can share a link to some particular message in the tree view.

Artifact dependencies

Artifact dependencies now support syntax similar to checkout rules, i.e. you can define a set of new line delimited rules: `[+:|-:]SourcePath[!ArchivePath][=>DestinationPath]`

My Changes page

- multi-line commit messages are collapsed
- builds carpet is more "vertical"
- you can view all related builds, not only suspicious

Visual Studio Addin

- Subversion 1.7 support
- Commit without remote run

Other improvements

- MSBuild/Visual Studio structured build log (tree view) improved
- multi-line text fields for build steps command line parameters
- new icons for build statuses
- new projects are not added on the overview automatically (yellow warning is shown instead)
- investigations and muted tests pages improved
- full list of fixed issues