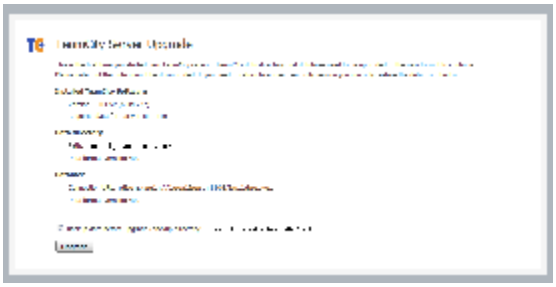
Eluru 6.0 EAP (build 15638) Release Notes

Upgrade UI

We believe that TeamCity upgrade procedure is one of the less painful among other servers on the market. Usually upgrade of TeamCity to a new version implies conversion of database and configuration files. Until now this activity was not clear to our users, because it was performed automatically. As a result users sometimes did not understand what happened and how upgrade worked.

We want to make it as explicit and clear as possible. With this EAP we introduce new upgrade process. If newly installed TeamCity decides that data conversion is required, it will ask for confirmation from server administrator and also offer a possibility to perform a backup (which we highly recommend in any case).

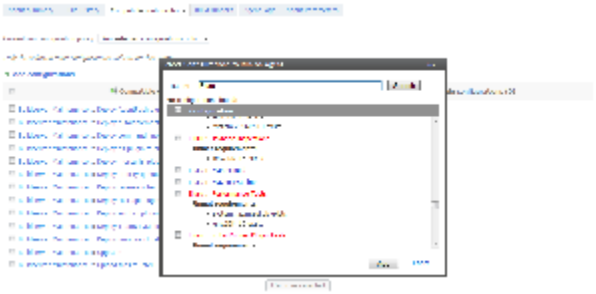


Note that automatic backup option will be available if previously installed TeamCity version is 6.0 or higher. For TeamCity 5.1 you'll need to make a backup manually.

Manual build configurations selection for an agent

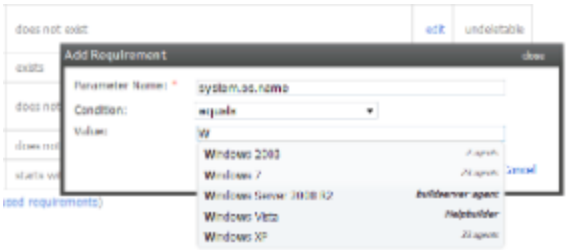
There are two different approaches how our users distribute agents among build configurations. Some of them configure this distribution with help of agent requirements and then TeamCity decides what set of agents can be used to run build configurations. Others want to specify explicitly which configurations can be built on which agents, and do not want to allow building of other configurations on these agents.

When we started developing TeamCity we decided that the first case would be more common, but at the same time we allowed the second case too. However, UI for the manual configurations selection for the second case is far from ideal, and quickly becomes unusable with increasing number of configurations. In TeamCity 6.0 we partially address this problem by providing better and more convenient user interface.



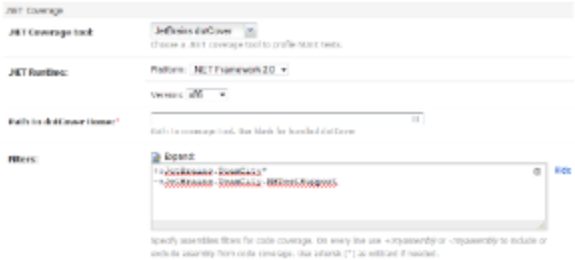
Autocompletion in agent requirements

Agent requirements dialog now has autocompletion enabled. Autocompletion works for both parameter name and parameter value. The whole set of parameters on all of the connected agents is considered, and you can see how many agents have this parameters defined, and with what values.



JetBrains dotCover integration

Jetbrains dotCover is now bundled with TeamCity, and is listed as another coverage engine for .NET runners. Also HTML coverage report for coverage data gathered by dotCover is available.



Gradle integration improvements

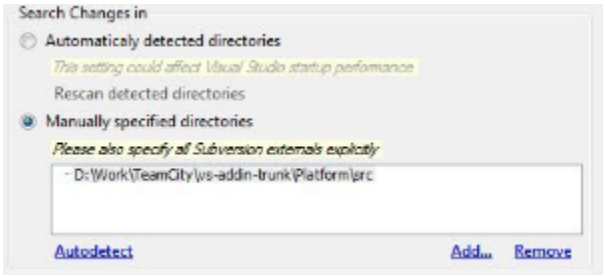
Gradle support in TeamCity has got a number of improvements:

- Improved test reporting, including Gradle parallel test execution.
- Gradle Wrapper scripts support. To use wrapper script "gradlew" just set appropriate checkbox on build step configuration page.
- System properties defined in TeamCity are now available in Gradle build. These properties can be accessed via "teamcity" project attribute.
- IntelliJ IDEA coverage engine is now supported for Gradle

Visual Studio Addin

For the past months our main focus for VS addin was performance. We've done several improvements and would like to see your feedback on that matter.

- Startup performance improved
- Local changes collecting performance improved
- Subversion integration reworked

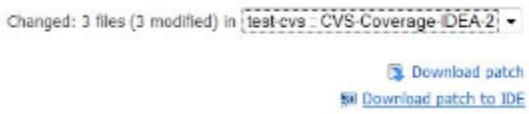


Eclipse plugin

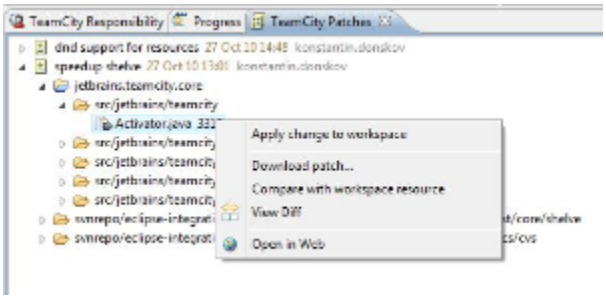
Download patch to IDE

One more link is now available for each change list in the web UI (right now it works for Eclipse plugin only, support for IntelliJ

IDEA is coming):

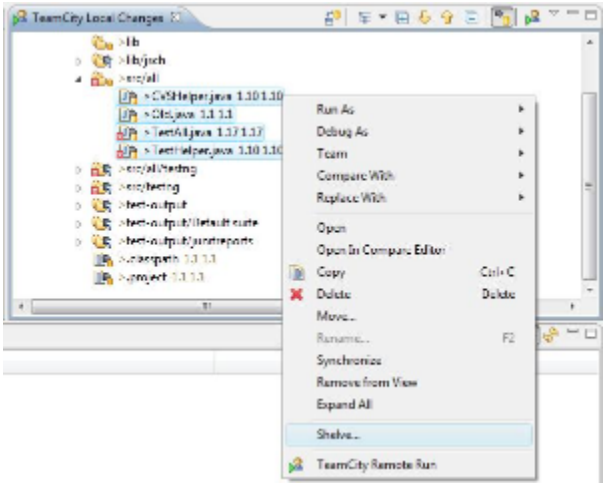


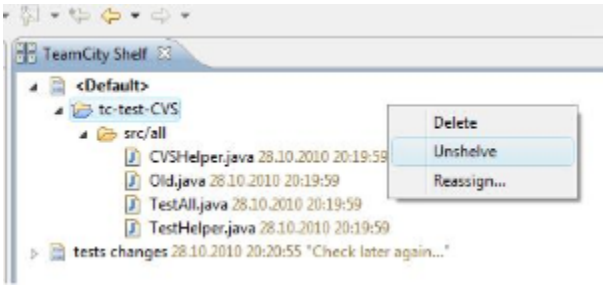
When you click the link, IDE automatically downloads patch from the server:



Shelve support

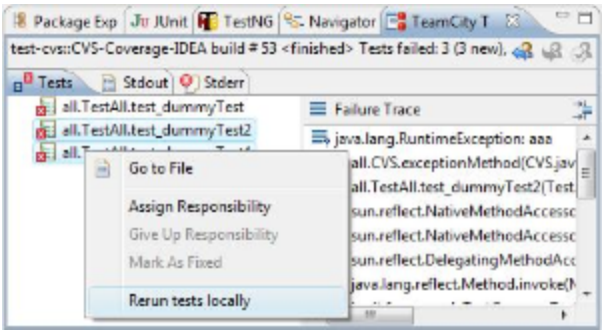
With the help of shelves you can save your changes locally and restore them later:





Re-run failed tests

IntelliJ IDEA plugin and Visual Studio Addin already have feature to re-run locally tests failed on the TeamCity agent. With this EAP this feature is available in Eclipse plugin too:



Other improvements

- Option for snapshot dependency to force running of the build on the same agent where its dependency is built
- Failed tests re-run reworked in IntelliJ IDEA plugin
- For Maven builds agent gathers information about used artifacts versions, dependencies and so on. This information is then shown as a tab on build results page.
- Support PMD CPD to produce code duplicates report
- Added support for NUnit 2.5.8
- See full list of changes for details