

Indore 2017.1 EAP1 (build 45965) Release Notes

- TeamCity Versioning Changes
- Visual Studio Team Services Connection
- Installing dotCover Centrally on Agents
- Installing Maven Centrally on Agents
- Build Failure Conditions
- REST API
- Cloud Agents
 - Update Root Project DSL Settings with Cloud Profiles
- Partial Build Log Display
- Faster Server Startup
- Web UI improvements
- Enforce clean checkout for build chain
- Other Improvements

TeamCity Versioning Changes

Since 2017, TeamCity adopts the common JetBrains versioning scheme that identifies versions by year, following the pattern: "<year>.<number of the feature release within the year>.<bugfix update number>". Hence, we'll be releasing TeamCity 2017.1, formerly known as TeamCity 10.1.

Visual Studio Team Services Connection

TeamCity improves its integrations with VCS hosting services and now TeamCity can be connected to VSTS the same way it is connected to GitHub, GitHub Enterprise or BitBucket Cloud. Once a connection is configured, it is really simple to setup projects which use VSTS repositories.

Installing dotCover Centrally on Agents

Now you can easily distribute to/ remove from all agents different versions of the JetBrains dotCover Command Line Tools centrally using the Administration | Tools page. The page displays the installed versions, the dotCover CLT usages in build configurations. The bundled version is set as default. You can install other versions and change the defaults.

Installing Maven Centrally on Agents

Now you can easily distribute to/ remove from all agents different versions of Maven centrally using the Administration | Tools page. The page displays all the bundled versions of Maven, with one set as default. You can install the version you need and change the default Maven version for all agents.

Build Failure Conditions

- You can now stop a build on exceeding a certain build metric specified using the Fail build on metric change condition.
- On encountering a specified text in the build log, you can opt to stop the build immediately.

REST API

- Retrieving all artifact dependencies for a build and a build configuration is now supported via the "artifactDependency" locator dimension
- The Swagger definition for the REST API is available under the .../[app/rest/swagger.json](#) URL

Cloud Agents

- Now several instance termination conditions can be specified. If any of the conditions is applied, the instance will be stopped. The new conditions specify:
 - the working time for the agent after which the instance will be terminated. If the time elapses while a build is in progress, the agent will wait until the current build is finished.

- how many minutes before the full hour an idle instance should be stopped: this allows avoiding charges for partial hours if you use Amazon EC2 instances, as they are billed in whole hours.
- When launching Amazon EC2 instances, TeamCity now tags all the resources (e.g. volumes and network adapters) associated with the created instances, which is important when evaluating the overall cost of an instance (taking into account the storage drive type and size, I/O operations (for standard drives), network (transfers out), etc.
- [Agent Cloud profile](#) configuration has been moved from the server level to the Root project level. In 2017.1 we intend to move cloud agent profiles to the project level, to allow Project administrators to configure their own cloud profiles without bothering System administrators.



If you use Kotlin DSL for your TeamCity Root project settings, make sure you update your DSL as described below.

Update Root Project DSL Settings with Cloud Profiles

Cloud profiles were moved from the server level to the Root project level, which means you can now use Kotlin DSL to define your cloud profiles.

If you had cloud profiles previously configured and you use Kotlin DSL for the Root project settings, the cloud profiles information will be lost unless you update your DSL.

To update your settings with the cloud profile information, perform the following:

1. Run the [Download settings in Kotlin format](#) action in the Root project and save the zip with the generated DSL
2. Copy project features of type "CloudIntegration" and "CloudProfile" from the `.teamcity/_Root/Project.kt` file to the root project config in your settings
3. Commit your changes to the VCS.
4. [Enable versioned settings](#) on the Versioned Settings tab of the Root project.

Partial Build Log Display

When opening large build logs, TeamCity now displays a part of it to avoid browser hanging. The configurable display threshold is set to 7M characters by default. You can view the full build log on clicking the corresponding link.

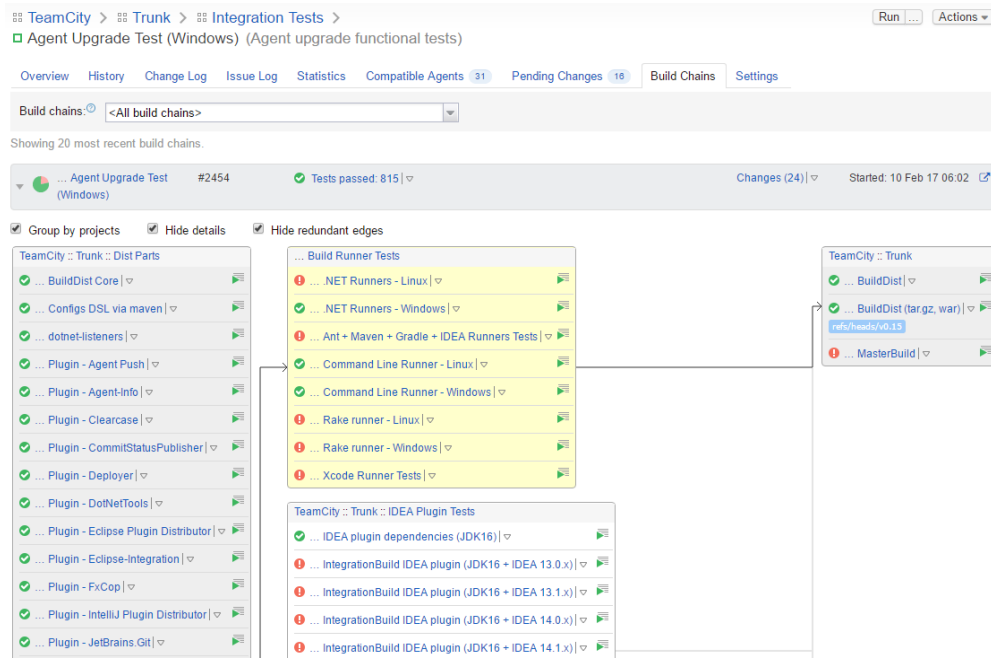
Faster Server Startup

During the server startup, several time consuming actions are now performed in parallel; in addition to that, the server warms up pages for faster display.

Web UI improvements

In this EAP, the following pages of the TeamCity Web UI were redesigned:

- the login page, register user, administrator setup, and startup pages have been revised
- the Create Project / Build Configuration pages were renewed
- The Build Chains tab now:
 - provides a more compact representation of chains. Previously, if several top builds triggered the same chain of dependent builds, TeamCity displayed several build chains. Now only one build chain is displayed with several "top builds".
 - has additional display options: "Group by projects", "Hide details", and "Hide redundant edges".
 - transitively highlights all the downstream/upstream builds when a build is selected in a build chain.




Enforce clean checkout for build chain

A new option in the schedule trigger and the custom run dialogue allows you to clean all files in the checkout directory before a build. If applied to snapshot dependencies, all the builds of the build chain will be forced to use clean checkout.

The option also enables rebuilding all dependencies (unless custom dependencies are provided via the custom build dialog or the schedule trigger promotes a build).

Other Improvements

- **Modifiable Build Queue Optimization:** Prior to this version, TeamCity optimized the build queue by automatically replacing queued builds with an earlier started build or a more recent queued build if the builds used the same change set and the same custom properties. This default behavior can now be manually disabled via the corresponding option in the VCS Build Trigger and Schedule Build Trigger.
- TeamCity plugin for IntelliJ-based products supports remote run from TFS now.
- The maintainDB tool now has a new option, `-I` (which stands for 'internal'). It is to be used in combination with `-D` ('d atabase') option to restore TeamCity into a new instance of the embedded HSQL database.
- It is now possible to override build triggers defined in a template.
- TeamCity now detects problems with agents upgrade and shows health report.
- You can test the connection to your VCS when configuring Commit Status Publisher.
- TeamCity users can now retrieve forgotten passwords.
- There is a handy way to quickly copy a failed test exception stacktrace to the clipboard.
- The test name character limit has been increased to 1024.
- CSRF protection has been implemented.
- TeamCity-JIRA issue tracker integration now supports automatic project ids synchronization: when configuring the connection, all project ids can be loaded from a JIRA server automatically. If a new project is added to this JIRA server, TeamCity will detect it and automatically synchronize the list of projects.
- IntelliJ IDEA Project runner supports 'Build Artifacts' tasks specified in the 'Before launch' list of IDEA run configurations.
- When the "Show changes from snapshot dependencies" option is used, TeamCity shows which dependent build the changes come from. On hovering over the  icon (in the Changes pop-up on the Build configuration overview page and on the Changes tab for a build) the number of the dependent build is displayed; clicking the link opens the Changes tab of the dependent build.
- [Fixed issues](#)