

Indore 10 EAP1 (build 40941) Release Notes

- Cross-platform TFS integration
- Versioned project settings in TFS
- Default auto-checkout on the agent
- Feature branches support for Perforce streams
- Java 8
 - Minimum server Java
 - Recommended agent Java
- Cloud support
 - VMware vSphere
 - Amazon EC2
- Better VCS Hosting Services Integrations
 - GitHub Issue tracker
 - Commit Status Publisher
- Notifications enhancements
- New Projects popup
- Web UI
 - Performance improvements
 - Console view in Build Log
 - Other UI enhancements
- New build configuration settings features
- New server health reports
- REST API enhancements
- Bundled Plugins
- Deprecated features
- Other Improvements

Cross-platform TFS integration

Starting from this version, TeamCity supports cross-platform TFS integration, which works on Linux, Mac OS X, and Windows platforms. Without installing additional software, TeamCity servers and build agents can interact with Team Foundation Servers (from 2010 to 2015) and Visual Studio Team Services.

The built-in TFS support used to depend on Microsoft Team Explorer, which works only on Windows machines. Now the built-in plugin can work in two modes: the default and cross-platform. The working mode is based on the availability of Team Explorer: if it is not present, the plugin falls back from the default to cross-platform mode.

Versioned project settings in TFS

Starting from this EAP, TeamCity allows storing the project configuration settings in a TFVC repository. The user specified in the TeamCity TFS root must have permissions to check in changes to the repository.

Default auto-checkout on the agent

Aiming at freeing up the server resources, we changed the default setting for the VCS checkout mode: now TeamCity will prefer to check out the sources on the agent before the build. If the agent-side checkout is not possible, TeamCity will display a corresponding health report item and will use the server-side checkout.

It only applies to newly created build configurations; all the existing one will work as configured before.

Feature branches support for Perforce streams

For VCS roots where Perforce stream is specified, you can now enable experimental support for feature branches.



When feature branches support is enabled in Perforce VCS root, TeamCity will monitor all streams matched by branch specification:

- Those streams will be descendant for the main stream, specified on the VCS Root page (and under the same stream root)
- Stream name should match the format from the branch specification. The default format is +:*(i.e. all streams). To limit streams to some pattern, use something like +:(task*). Or, you can set branches explicitly, one-per-line.

As a result you'll see picture similar to the one you get for Git and Mercurial:



If you have a VCS trigger for a build configuration with such VCS Root, TeamCity will trigger builds for all matching streams. But, you can use an additional filter in the VCS trigger to trigger builds only for specified branches.

TeamCity should switch agent working copy between streams without clean checkout (when agent-side checkout is used).

TeamCity also allows to start a personal build (remote run/delayed commit) in an alternative stream. When there are no explicit configurations targeted to the workspace stream, suitable configurations dialog for remote run should show stream-based configurations with the same stream root as in workspace stream. TeamCity should also switch working copy to this stream when running the personal build (the stream name is indicated in the build log).

Known limitations:

- Task streams are supported only for configurations using agent-side checkout. Development streams work with server-side checkout as well
- Due to Perforce limitations, TeamCity is not able to reliably detect a branch point in the VCS history, as a result TeamCity cannot construct a proper graph of changes on change log
- Merge points are not detected yet

Java 8

Minimum server Java

Starting from TeamCity 10, the TeamCity server requires a Java 8 JRE/JDK (included in the Windows .exe distribution).

Recommended agent Java

Starting from the next TeamCity version, the minimum requirement for the Java on the agent will be Java 1.8 (included in the Windows .exe distribution). If an earlier version is detected, TeamCity displays a [health item](#) suggesting that you now consider upgrading the agent Java.

Cloud support

There is a number of improvements in TeamCity agent cloud support:

- You can now run a build on any agent of a certain cloud image (TeamCity will also attempt start a new one, if no available cloud agents of that type exist) from the Run Custom Build dialog and from the IDE's remote run customization dialog

VMware vSphere

- When configuring a cloud profile, you can now limit a number of instances across all images and set the limit per image.
- TeamCity detects cloud profile changes immediately, and forces shutdown of the agents started prior to these changes once the agents finish the current build.

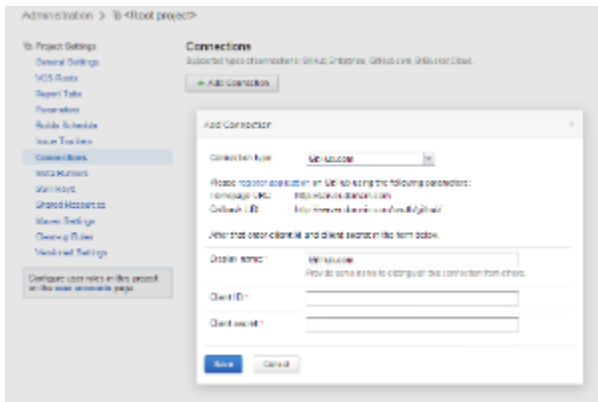
Amazon EC2

- If a user removes an image from a profile, TeamCity displays a warning that the instances already started by TeamCity will not be automatically stopped.
- Amazon EC2 images are now configured via a more convenient web interface, similar to the one in VMware vSphere plugin

Better VCS Hosting Services Integrations

Now TeamCity can be connected to GitHub, GitHub Enterprise or BitBucket Cloud. Once a connection is configured, it becomes easier to setup projects which use repositories on GitHub or BitBucket Cloud.

Connections are configured per project, see the new Connections menu item on the project administration page. The configured connection is accessible in this project and in all of its subprojects. In case of global VCS hosting services like GitHub or BitBucket Cloud, it makes sense to configure a single connection for the Root project.

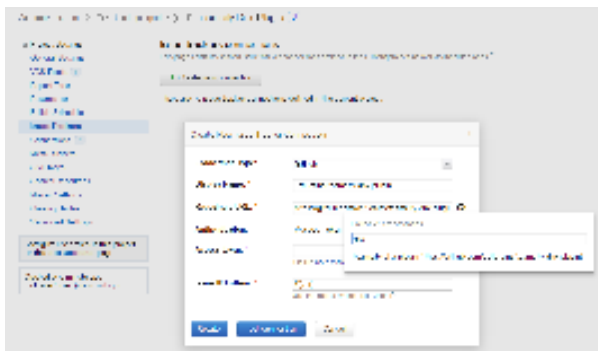


After that a small GitHub or BitBucket icon will be shown in several places where repository URL can be specified: create project from URL, create VCS root from URL, create Git or Mercurial VCS root, create GitHub issue tracker. Clicking this icon shows GitHub or BitBucket repositories available to the current user:



GitHub Issue tracker

TeamCity comes with built-in support for many of issue trackers, which now include [GitHub](#). If a GitHub connection is configured, connecting TeamCity to GitHub issues can be done easily:



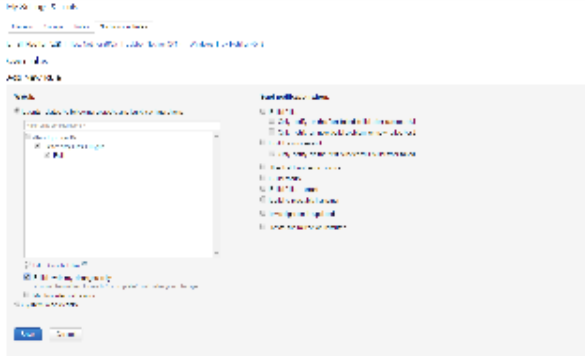
If you are using the TeamCity-GitHub [third-party plugin](#), you can safely remove it: the built-in TeamCity integration will detect the existing connection to GitHub issue tracker and pick up your settings automatically.

Commit Status Publisher

Commit status publisher plugin is now bundled with TeamCity. Recently we updated this plugin with support for GitHub and BitBucket Cloud (in addition to already supported Gerrit and BitBucket Server).

Notifications enhancements

Previously it was not possible to configure notifications to watch for builds with your changes in specified build configurations. In this EAP we reworked how notifications are configured in an attempt to make them more flexible and more obvious:



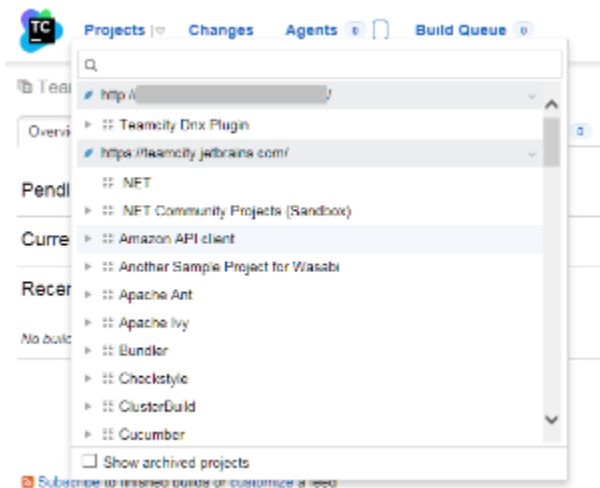
New Projects popup

In this EAP a new version of TeamCity Projects popup is available. This popup was written from scratch and provides better performance, besides it allows browsing projects and build configurations on multiple servers.

The pop-up is disabled by default and can be enabled by setting the `teamcity.restProjectsPopup.enabled=true` [internal property](#).

There is no web interface for specifying multiple servers yet, their list can be defined via another [internal property](#): `teamcity.federation.attachedServers=<comma separated list of TeamCity server URLs>`.

Note that the new pop-up fetches available projects using REST API, so [CORS](#) must be enabled on the target server for the pop-up to work. By the way popup is compatible with TeamCity 9.1.x REST API, so you should be able to connect this EAP build with your existing 9.1.x servers.



Web UI

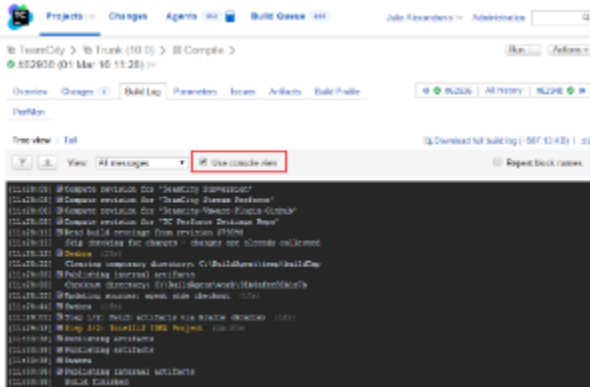
Performance improvements

We moved calculation of changes, artifacts and some other build-related data into background, which resulted in faster loading

of pages; this is especially noticeable on the Projects overview and build configuration home page; information about changes is loaded lazily after the main page is loaded.

Console view in Build Log

You can now enable the dark theme in the build log by selecting the appropriate check-box:



Other UI enhancements

TeamCity web interface has become even convenient with the following enhancements:

- The Agent pools page has been rewritten to provide better performance in case of a large number of pools and agents; the page now has the default agent pool at the top, collapsible agent pools, and the option to hide archived projects from display.
- Branch filter field in build triggers, build features, notification rules, artifact dependencies, and other places now has icon which can show available branches and simplify defining filter rules.
- Health items icons are now shown for projects too; in addition, the icons are now displayed in breadcrumbs which should make them more noticeable.

New build configuration settings features

- Artifact dependencies and agent requirements can now be disabled to preserve them in settings without affecting the build (similarly to build features, triggers and build steps).
- Multiple agent requirements for a single parameter can be added now. The conditions are treated as 'and' to determine compatible agents.
- An experimental feature to disable editing of project or build configuration settings via the web UI is added. This can be useful if you generate the settings (e.g. via REST API) and want to forbid changing the settings via the UI. To turn it on, add the `teamcity.ui.settings.readOnly=true` configuration parameter to the build configuration or project.

New server health reports

- agents not running the recommended Java: this report shows all of the agents running under Java version < 1.8
- report for large server side patches with recommendation to switch to agent side checkout
- inefficient artifacts publishing: TeamCity detects a build publishing many small artifact files and suggests publishing them as a single zip archive
- if a project or build configuration has secured parameters and is configured to build GitHub pull requests, this report will raise the warning, because malicious code submitted via the pull request can obtain these secured parameters
- health report suggesting configuring GitHub issue tracker if a VCS root points to GitHub

REST API enhancements

- Now you can get information about a changed file action: added, edited, removed, copied or unchanged reported via the `changeType` attribute for the files listed by `<teamcity URL:port>/httpAuth/app/rest/changes/id:<changeid>`
- To get the build status icon for several builds, use the `{<teamcity URL:port>/httpAuth/app/rest/builds/aggregated/<build locator> request and strob build locator dimension.`
- Now you can set a comment when disabling a build agent via REST API.
- You can fetch hanging builds via REST API now.

Bundled Plugins

TeamCity 10.x now includes the following plugins, and you no longer need to install them separately:

- [Deployer plugin](#) - open source plugin, containing a number of additional runners to upload files to FTP, SSH and deploy wars into Apache Tomcat servlet container
- [Commit status publisher](#) - open source plugin providing ability to submit status of a build into different VCS hosting services: GitHub and GitHub Enterprise, BitBucket Cloud and BitBucket Server and Gerrit
- [VMware vSphere plugin](#) - open source plugin providing support for cloud agents on VMware vSphere

Deprecated features

The Legacy version of TeamCity VS Add-in (available for Visual Studio versions from 2005 to 2013 and compatible with JetBrains .NET tools prior to ReSharper 9.0, dotCover 3.0 and dotTrace 6.0.) is no longer available starting from TeamCity 10.x.

The TeamCity Visual Studio Add-in is shipped as a part of ReSharper Ultimate products bundle. After installation, the TeamCity Add-in will be available under the RESHARPER menu in Visual studio.

Note that the installer will remove the pre-bundle products versions: TeamCity and ReSharper versions prior to 9.0, dotCover prior to 3.0, dotTrace prior to 6.0.

ReSharper Ultimate does not support the Visual Studio versions 2005 and 2008.

Other Improvements

- The Administration | Projects Import page now allows uploading the backup archive for import.
- Improved search: you can now find builds with the specified build problem.
- When using NUnit 3 in the NUnit test runner, you can instruct TeamCity to run some tests before others and reduce test failure feedback time.
- When Perforce VCS mapping is edited, clean checkout is no longer forced. Note that [clean checkout will be triggered](#) for all Perforce vcs roots after upgrade to TeamCity 10.
- Output of successful tests in the Gradle runner can be suppressed by adding the `-Dteamcity.ignoreTestStdOut=true` property to the Additional command line parameters field of the runner.
- .NET Inspections runner supports .NET 4.6.x as target framework.
- Code coverage percentage is now published.
- [fixed issues](#)