

# Hajipur 9.0 EAP2 (build 31717) Release Notes

- [Server Clean-up in Background](#)
- [Initial work on Perforce Stream Support](#)
  - [Known Limitations](#)
- [Settings in Version Control](#)
- [Per-project Configuration of Issue Tracker Integration](#)
- [WebSockets for Server Events](#)
- [Customizable Behavior on Snapshot Dependency Failure](#)
- [Floating Point Numbers Support for Statistic Values](#)
- [Gradle Project in Inspections \(IntelliJ IDEA\) and Duplicates Finder \(Java\)](#)
- [Mercurial VCS Root Enhancements](#)
- [Other Improvements](#)

## Server Clean-up in Background

Now the build history clean-up is run as a background process, which means that now there is no server down-time and your TeamCity is available 24/7.

One thing should be noted though: if you are using the HSQL database, there is still a period of server unavailability when the HSQL database is being compacted. One more reason to switch to external database!

## Initial work on Perforce Stream Support

Starting from this EAP, you can configure a Perforce VCS Root to monitor Perforce streams ([TW-20040](#)). The work has just been started, but checkout on an agent and on the server are already supported.

## Known Limitations

- The switch between streams is not efficient yet and will result in a full checkout.
- In case of checkout on agent, exclude and include checkout rules are not supported. The only supported rule is: `. => <some dir>`.

## Settings in Version Control

We continue improving [settings synchronization with the version control](#) in this EAP build:

- now, you can store your settings not only in Git, but also in Mercurial
- changes in VCS settings are only shown for build configurations affected by these changes
- a number of critical problems have been fixed making this feature much more stable

## Per-project Configuration of Issue Tracker Integration

The configuration of [integration with issue trackers](#) has moved to the project level, and now users with the Project Administrator permissions can access this feature in the Project Settings. The integration now is more enterprise friendly: if you have multiple projects on both the TeamCity and the issue tracker server or if you are using different issue-trackers for different projects, you will definitely benefit from this improvement.

Enabling integration for the project also enables it for all its subprojects; if the configuration settings are different in a subproject, its settings have priority over the project's settings.

## WebSockets for Server Events

TeamCity user interface has always been quite responsive. We are trying to avoid the whole page refresh as much as possible. As such we need some way to deliver events occurring on the server to browsers. For this we traditionally used HTTP polling. While being quite reliable, it still has disadvantages:

- it increases the load on the server especially in case of a large number of users
- its responsiveness is limited by the poll interval

While responsiveness is a nice addition, a high server load is still a bigger problem. We attempted to solve these issues in this TeamCity EAP. Now it uses the [WebSocket](#) protocol for events, running builds updates and statistics counters (the number of builds in the queue, the number of agents, etc). Now every client establishes a single WebSocket connection with the TeamCity

server, and the server pushes information about events to the browsers once they occur. Then it's up to the client code to decide what and how should be updated on the page.

The WebSocket connection is a new technology and comes with some limitations:

- At the moment TeamCity requires Tomcat version 7.0.43 and above for WebSockets to work; TeamCity will switch to polling automatically if this is not the case.
- The WebSocket connection is not available for Internet Explorer versions below 10 and Safari; for them polling will be used.
- If the TeamCity server is behind a proxy, additional [configuration](#) is required for the WebSocket connection to work.

## Customizable Behavior on Snapshot Dependency Failure

Previously TeamCity offered limited control over dependent build status if dependency fails. Now more options are available for more flexible configuration.

For each failed or failed to start dependency you can select one of the four options:

- Run build, but add problem: the dependent build will be run and the problem will be added to it, changing its status to failed (if problem was not muted earlier)
- Run build, but do not add problem: the dependent build will be run and no problems will be added
- Make build failed to start: the dependent build will not run and will be marked as "Failed to start"
- Cancel build: the dependent build will not run and will be marked as "Canceled"

## Floating Point Numbers Support for Statistic Values

Previously TeamCity only supported integer numbers as statistics values. This caused problems with metrics like coverage or with custom values for percentage. For instance, if a statistic value changed by a fraction of a per cent, it was hard to configure a build failure condition to spot these little variations. Now TeamCity supports fractional numbers with up to 6 decimal points for all statistic values, finally eliminating this problem.

## Gradle Project in Inspections (IntelliJ IDEA) and Duplicates Finder (Java)

For quite a long time TeamCity inspections and duplicate code analysis tasks were available for IntelliJ IDEA and Maven projects only. Starting with this EAP you can now use Gradle projects as well.

## Mercurial VCS Root Enhancements

TeamCity now supports configuration of Mercurial options per repository leaving behind the old approach which implied editing global mercurial configuration files on the server and agents. Now it is possible to [enable some mercurial extensions](#) only in the repositories where they are required.

## Other Improvements

- Now the Maven artifact dependency trigger can use authorization configured using [the new advanced settings](#) of the trigger.
- REST API: added support for tagging queued and running builds, added ability to download build's artifacts in an archive
- Location of memory dumps produced by TeamCity has changed from the `<TeamCity installation directory>/bin` to `<TeamCity installation directory>/logs/memoryDumps/ directory`.
- Bundled Maven 3.2.x
- NTLM authentication can be used for Subversion repositories
- [fixed issues](#)