

Jaipur 2018.1 (build 57605) EAP1 Release Notes

- User interface changes
- Inherited build steps configuration improvements
 - Ability to redefine inherited build step settings
 - Ability to have pre- and post-steps in a template
- Enforced settings
- PowerShell Core support
- Docker plugin improvements
- Re-run build action improvements
- Shared resources in composite builds
- Other improvements

User interface changes

The user interface update that we started in the previous version continues in the future 2018.1 version.

In this EAP build, the build line has been redesigned a bit and the agent history page now looks similar to the All Builds page (which was introduced in 2017.2.x):

| Build ID | Build Name | Status | Changes | Start Time | Duration |
|----------|--|---------------------------------|---------------------|-----------------|------------|
| #135 | TeamCity / Indora-2017.2.x / Integration Tests / VCS Tests / TFS Plugin Tests / IntegrationBuild TFS .NET SDK (Windows) | Tests passed: 1811, ignored: 15 | Changes (2) | 29 Mar 18 06:11 | 19m 22s |
| #51101 | TeamCity / Indora-2017.2.x / Integration Tests / Plugins Integration Tests / Perfmon / Perfmon Integration Tests (Windows) | Tests passed: 3 | No changes | 29 Mar 18 06:09 | 1m 22s |
| #414 | TeamCity / Indora-2017.2.x / Integration Tests / DSL Converters Tests | Tests passed: 8 | No changes | 29 Mar 18 07:03 | 5m 23s |
| #1441 | TeamCity / Trunk / Integration Tests / NuGet plugin / Prepare for NuGet support integration tests | Success | No changes | 29 Mar 18 06:16 | 6s |
| #71609 | TeamCity / Trunk / Compile | Success | Filipp Riabchun (1) | 29 Mar 18 03:20 | 2h:36m:07s |
| #2471 | TeamCity / Trunk / Integration Tests / Build Runner Tests / Ant Runner Tests | Tests passed: 915 | Pavel Sher (1) | 28 Mar 18 23:55 | 1h:11m:24s |
| #1468 | TeamCity / Trunk / IDEA Plugin Tests / IntegrationBuild IDEA plugin (IDEA 2017.2) | Tests passed: 73 | Changes (3) | 28 Mar 18 23:51 | 3m:10s |

We're working on other pages where lists of builds are shown. Expect more changes in the next EAP build.

Inherited build steps configuration improvements

Ability to redefine inherited build step settings

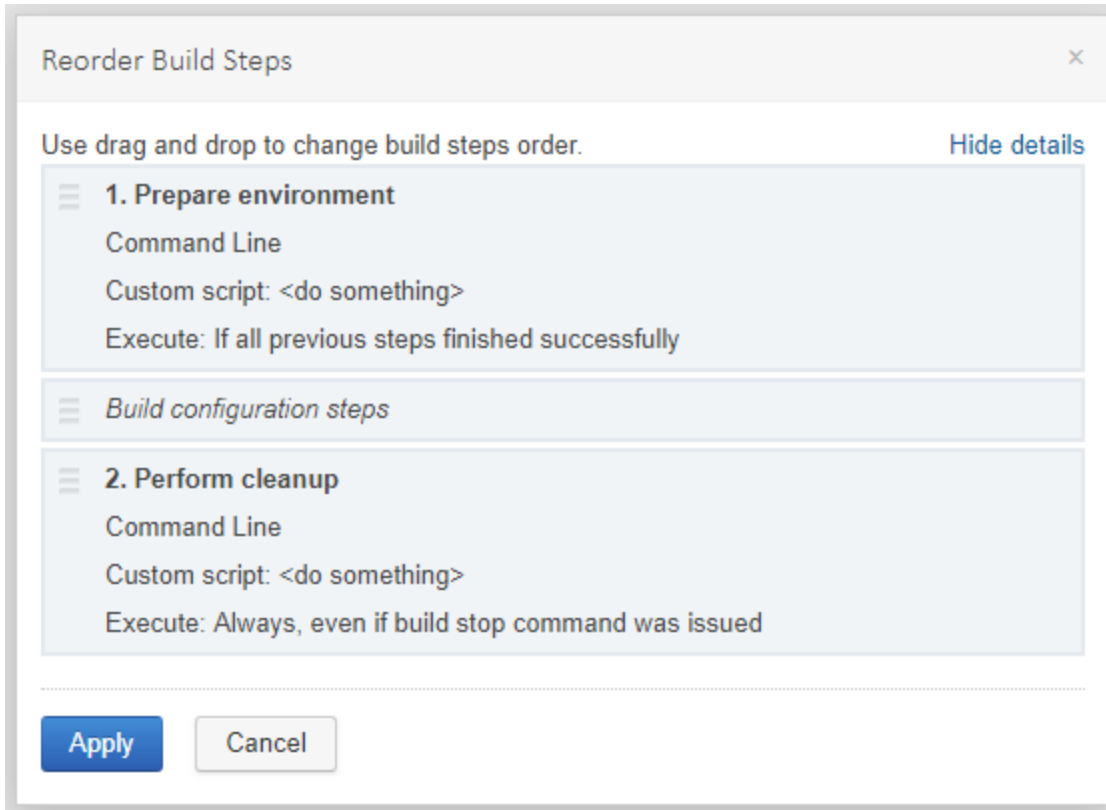
In the current TeamCity version, if a build step is defined in a template, it is shown as read-only in a build configuration inheriting this build step. If some customization is required, then a parameter reference has to be defined for some settings in the template, and a specific value should be set for this parameter in the build configuration. However, this workaround works for text fields only.

In this EAP it is finally possible to customize an inherited build step, i.e. it is possible to change every setting of the inherited build step without the need to introduce parameters.

Ability to have pre- and post-steps in a template

There is sometimes a need to define a common build step in a template, so that this step will be executed either before all build configuration steps or after them.

In this EAP build, for a given template it is possible to define such steps and then define their placement in respect to the build configuration steps. All build configuration steps are represented as a placeholder in the Reorder Build Steps dialog. The template steps can be placed before or after this placeholder.

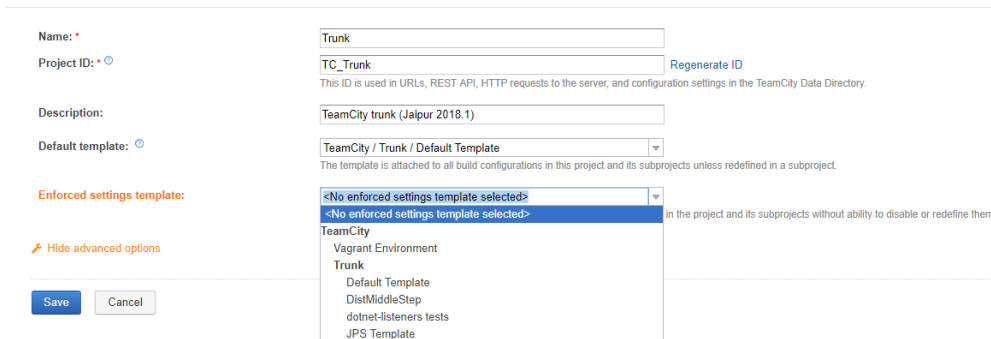


Note: you still can have a completely custom order of steps in a build configuration inherited from a template.

Enforced settings

Starting with this EAP, TeamCity provides the ability to enforce settings for all of the build configurations in a project hierarchy. For instance, with help of this feature it is possible to enforce agent side checkout everywhere, or make sure that all build configurations have some strict execution timeout.

To enforce some settings in the project hierarchy, a template with these settings must be created. After that, a system administrator can set this template as the enforced settings template in the project:



To some extent, the enforced settings template works similarly to the default template - i.e. all of its settings are inherited in build configurations of the project hierarchy. The difference is that these inherited settings cannot be disabled or overridden anyhow. In addition to that, only system administrator can associate a project with a specific enforced settings template - project administrator permissions are not sufficient. On the other hand, the template itself can be edited by a project administrator who can administer the project where the template is defined.

If the enforced settings template is specified in a project and another template is assigned as the enforced settings in a subproject, the subproject's template will have the higher priority.

Limitations: Dependencies, triggers, and build steps are currently not enforced. We're not convinced there are useful use cases for them, please let us know if you have some cases in mind.

PowerShell Core support

- Cross-platform PowerShell ([PowerShell Core](#)) is now supported on Windows, MacOS and Linux
- Side-by-side installations of PowerShell Desktop and PowerShell Core is supported under Windows

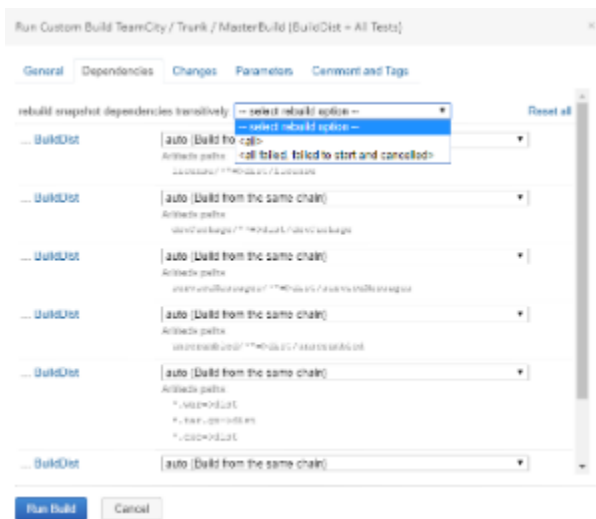
Docker plugin improvements

- Docker wrapper now forms build step environment more correctly and passes the relevant environment variables to the container. [TW-53498](#)
- Docker wrapper can now use Gradle and Maven provided by the Docker images in the corresponding build steps. [TW-54066](#), [TW-51179](#)

Re-run build action improvements

Now when re-running a build, TeamCity preserves all the custom parameters of this build and its dependencies as of the time of the original run.

Besides that, on the dependencies tab, there is now an additional option to rerun the failed/failed to start or cancelled dependencies of the build:



Shared resources in composite builds

Since this EAP shared resources can be locked not only for regular builds, but for composite builds as well. A lock on the specified resource will be acquired when a composite build starts (when the first build of the composite build chain starts); the lock will be released when the composite build finishes (the last build in its chain is finished).

The locks acquired on composite builds affect only these composite builds and are not propagated to their individual parts. For example, if a resource has a quota of N, then N composite builds that have a read lock on this resource can be run concurrently. The number of concurrent individual builds inside these composite builds will not be affected by the resource quota.

Adding a lock on a composite build is done the same way it is done for a regular build: it can be done by adding the [Shared Resources](#) build feature.

Other improvements

- Browsing of artifacts inside archives is now supported for all external artifacts storages too (previously this feature was available only if artifacts were stored in the internal artifacts storage).
- Java 9 and Java 10 can now be used to run an agent.
- Gerrit Commit Status Publisher now allows an admin to configure custom values for success and failure instead of +1 and -1 respectively. This label is also configurable. Note that since TeamCity 2018.1 Gerrit 2.6. is the minimal version supported by Commit Status Publisher.
- Build chains page has got new grouping option: it groups together in a single node all of the build configurations which are not related to the current context.

- Duplicates finder (.NET) and Inspections (.NET) runners have been renamed to Duplicates finder (ReSharper) and Inspections (ReSharper) respectively. You can now select the platform bitness for `InspectCode` when using Inspections (ReSharper).
- [All fixed issues](#)