

# Build Configuration Template

Build Configuration Templates allow you to eliminate duplication of build configuration settings. If you want to have several similar (not necessarily identical) build configurations and be able to modify their common settings in one place without having to edit each configuration, create a build configuration template with those settings. Modifying template settings affects all build configurations associated with this template.

On this page:

- [How can I create build configuration template?](#)
- [Associating build configurations with templates](#)
- [Detaching build configurations from template](#)
- [Redefining settings inherited from template](#)
  - [Modifying Settings](#)
  - [Using parameter reference](#)
    - [Example of configuration parameters usage](#)



Build configuration templates support project hierarchy: you can use the templates from the current project and its parents. On copying a project or a build configuration, the templates that do not belong to the target project or one of its parents are automatically copied. Now you can associate a build configuration to a template only if the template belongs to the current project or one of its parents.

If you upgraded from versions earlier than TeamCity 8.0, your configurations from one project might be associated with templates from an unrelated project. After upgrading to TeamCity 8+, such templates can become inaccessible in out-of-order projects during the settings editing or new build configuration creation. To reuse build configuration templates from an unrelated project, it is recommended to manually move the templates into the common parent project (or the [Root project](#) if you want them to be globally available).

## How can I create build configuration template?

- Manually, like a [regular build configuration](#).
- Extract from an existing build configuration: there is the Extract template option available from the Actions button at top right corner of the screen. Note that if you extract a template from a build configuration, the original configuration automatically becomes [associated](#) with the newly created template.

## Associating build configurations with templates

- You can create new build configurations based on a template.
- You can associate any number of existing build configurations with a template: there's the Associate with Template option available from the Actions button at top right corner of the screen.



When you associate an existing build configuration with a template, the build configuration inherits all the settings defined in the template, and if there's a conflict, the template settings supersede the settings of the build configuration (except dependencies, parameters and requirements). The settings inherited from a template [can be overridden](#).

You can associate a build configuration to a template only if the template belongs to the current project or one of its parents.

A template which has at least one associated build configuration cannot be deleted, the associated build configurations need to be detached first.

## Detaching build configurations from template

When you detach a build configuration from a template using the Detach from template option available from the Actions button at top right corner of the build configuration settings screen, all settings from the template will be copied to the build configuration and enabled for editing.

## Redefining settings inherited from template

Although a build configuration associated with a template inherits all its settings (marked as inherited in the UI), it is still possible to redefine a number settings in an associated build configuration. Modifying settings in the template will influence all configurations associated with this template.

 Inherited settings cannot be deleted from an associated build configuration.

The following settings can be overridden in a build configuration inherited from a template:

- build number format
- artifact paths
- build options (hanging builds detection, triggering personal builds, status widget, number of simultaneously running builds)
- VCS checkout mode
- checkout directory
- clean all files before build
- show changes from snapshot dependencies
- execution timeout
- all common build failure conditions, including execution timeout
- Parameters
- Agent Requirements
- Snapshot Dependencies

 Note that if you rename an inherited parameter or an agent requirement in a build configuration, this parameter/ agent requirement is actually disconnected from the template and is treated as a new parameter created in the build configuration. For example, if you rename a parameter in an associated build configuration, and then rename the same parameter in the template, you will end up with two parameters in the build configuration: the new one inherited from the template, and the old one which was redefined in the build configuration.

Besides, you can redefine settings configured via parameter references or add new items to lists.

## Modifying Settings

### Text field settings

When you specify some fixed value in a text field of a template, it is inherited as is and cannot be changed in an associated build configuration.

However, in most of the text fields of your template settings (except names (build configuration, parameter, build step), descriptions, agent requirements, typed parameters definitions), you can use a [reference](#) to a [build parameter](#) instead of the actual value. Thus you can define the actual value of this parameter in each particular associated build configuration separately.

See [below](#) for an example of configuration parameters usage.

Other settings: drop-downs, lists, check boxes, password fields, etc.

These settings are inherited from a template as is and cannot be redefined in an associated build configuration.

### Collections

A collection of settings, such as parameters, build steps, VCS roots, or build triggers can be extended in an inherited configuration.

- You can add new element to a collection, for example one more VCS root.
- In some cases you can reorder the collection elements in the inherited configuration, for example, build steps.

 Modified settings are highlighted with a yellow border and the Reset button appears on the right of the modified settings enabling you to revert the changes to the original settings of the template.

## Using parameter reference

To introduce a configuration parameter reference, use the `%ParameterName%` syntax in the template text field. Once introduced, this parameter appears on the Parameters page of the build configuration template marked as requiring a value.

You can either specify the parameter's default value or leave it without any value. You can then define the actual value for the parameter in a build configuration associated with the template.

See also [Configuring Build Parameters](#) and [Defining and Using Build Parameters in Build Configuration#Using Build Parameters in Build Configuration Settings](#).

### Example of configuration parameters usage

Assume that you have two similar build configurations that differ only by checkout rules. For instance, checkout rules for the first configuration should contain `+:release_1_0 => .'`, and for the second `+:trunk => .'`. All other settings are equal. It would be useful to have one template to associate with both build configurations, but this means changing the checkout rules in each build configuration separately.

To do so, perform the following steps:

1. Extract a template from one of those configurations.
2. In the template settings, navigate to Version Control Settings, open the Checkout rules dialog for the VCS root and enter there: `%checkout.rules%`

3. For the inherited build configuration, open the configuration settings page and on the Parameters page specify the actual value for the `checkout.rules` configuration parameter.
4. For the second build configuration, use the Associate with template option from Actions and choose the template. Specify an appropriate value for the `checkout.rules` parameter right in the "Associate with Template" dialog. Click "Associate".

As a result, you'll have two build configurations with different checkout rules, but associated with one template.

This way you can create a configuration parameter and then reference it from any build configuration, which has a text field.

See also:

[Administrator's Guide: Creating and Editing Build Configurations | Configuring Build Parameters](#)