Typed Parameters

When adding a build parameter (system property, environment variable or configuration parameter), you can extend its definition with a specification that will regulate parameter's control presentation and validation.
This specification is the parameter's "meta" information that is used to display the parameter in the Run Custom Build dialog. It allows making a custom build run more user-friendly and usable by non-developers.
Consider a simple example. You have a build configuration in which you have a monstrous-looking build parameter that regulates if a build has to include a license or not; can be either true or false; and by default is false. It may be clear for a build engineer, which build parameter regulates license generation and which value it is to have, but it may not be obvious to a regular user.

Using the build parameter's specification you can make your parameters more readable in the Run Custom Build dialog.

On this page:

Adding Parameter Specification

To add specification to a build parameter, click the Edit button in the Spec area when editing/adding a build parameter.

All parameters specifications support a number of common properties, such as:

- Label: some text that is shown near the control in the Run Custom Build dialog.
- Description: some text that is shown below the control containing an explanatory note of the control use.
- Display: If hidden is specified, the parameter will not be shown in the Run Custom Build dialog, but will be sent to a build; if prompt is specified, TeamCity will always require a review of the parameter value when clicking the Run button (won't require the parameter if build is triggered automatically); if normal is selected, the parameter will be shown as usual.
- Type : Currently you can present parameters in following forms:
  - a simple text field with the ability to validate its value using regular expression;
  - a checkbox;
  - a select control;
  - a password field.

The table below provides more details on each control type.

Type	Description
Text	The default. Represents a usual text string without any extra handling
Checkbox	True/false option represented by a check box
Select	"Select one" or "select many" control to set the value to one of predefined settings
Password	This is designed to store passwords or other secure data in TeamCity settings. TeamCity makes the value of the password parameter never appear in the TeamCity Web UI: it affects the settings screens and the Run Custom Build dialog where password fields appear. Also, the value is replaced in the build's Parameters tab and build log. The value is stored scrambled in the configuration files under TeamCity Data Directory. Please note that build log value hiding is implemented with simple search-and-replace, so if you have a trivial password of "123", all occurrences of "123" will be replaced, potentially exposing the password. Setting the parameter to type password does not guarantee that the raw value cannot be retrieved. Any project administrator can retrieve it and also any developer who can change the build script can in theory write malicious code to get the password.

Depending on the specification's type, there are additional settings.

Text	Allowed value - choose the allowed value. For the Regex option, specify Pattern, a Java-style regular expression to validate the field value, as well as a validation message.
Checkbox	Checked value/Unchecked value: Specify values for the parameter to have depending on the checkbox's state.
Select	Check the Allow multiple box to enable multiple selection. In the Items field specify a newline-separated list of items. Use following syntax `label => value` or `value`.

Manually Configuring Parameter Specification

Alternatively, you can manually configure a specification using specially formatted string with syntax similar to the one used in service messages (`typeName key='value'`).
For example, for text: `text label='some label' regex='some pattern'`.

Copying Parameter Specification

If you start editing a parameter that has a specification, you can see a link to its raw value in the "Edit parameter" dialog. Click it to view the specification in its raw form (in the service message format). To use this specification in another build configuration, just copy it from here, and paste in another configuration.

Modifying Parameter Specification via REST API

You can also view/edit typed parameters specification via REST API.

See also:

Administrator's Guide: Configuring Build Parameters | Defining and Using Build Parameters in Build Configuration | Predefined Build Parameters