

Project Wizard

- Details
 - Name
 - Slug
 - Description
 - Versioning
 - Version Counter
 - Reuse Build Number
 - Version Format String
 - Overrides
- Repositories
- Variables
- Security
- Cleanup
- Finished

The Project Wizard is where all properties associated with a project are set. The Project Wizard has many sub pages or steps that allow you to edit and create various Continua elements and attach them to the current project.



The various sections can be quickly accessed through the Project Wizard Menu as seen above. This menu can be found at the top of every Project Wizard page.

Details

Name

Dev Project

The name of your project may contain up to 128 characters. Your project's name can contain letters, numbers, spaces, dashes, periods or underscores.

Slug

dev_project

The project's slug is used in the url when linking to a project. It may contain up to 128 characters but it must start with a letter or a number and can only contain letters, numbers, underscores and hyphens.

Description

Versioning

☐ Set project-wide versioning

[✕ Exit Wizard Without Saving](#) [✓ Save & Complete Wizard](#) [➤ Save & Continue](#)

The Details step of the Project Wizard is where a project's basic settings can be input.

Name

This is used to identify each project. Each project must have a unique name.

Slug


The Slug is used in Continua URLs when linking to a project or item within a project. By default, this will be the project's name with all special characters replaced with underscores. Each project must have a unique slug.

Description

This can contain any additional information relating to the current project.

Versioning

Versioning can be applied either at project level or configuration level.

 Note that project-wide versioning is only available from version 1.7

If project-wide versioning is set then the same version counter is incremented for each build across all project configurations. This can be overridden allowing a project configuration to increment its own version counter or use it's own version formatting for each build.

If project-wide versioning is not set then all configurations have their own version counters and formatting.

Versioning

☒ Set project-wide versioning

Version Counter

0

The version counter is used to populate `$Build.BuildNumber$` and specifies the starting number for your incremental build count.

☐ Reuse build number if previous build was discarded, stopped or failed while initialising

Version Format String

1.0.0.\$Build.BuildNumber\$

The version format string can contain **expressions** which allows this field to use dynamic values.

The following configurations are overriding these settings

- Dev Test Configuration

Clear All Overrides

Version Counter

The version counter is a numeric value which is incremented by one every time a project configuration is built.

This can be considered an offset for any builds created under this project. For example, if you set the Version Counter to 50 then the first build will be version 1.0.0.50, the second build will be 1.0.0.51, etc.

By default this is set to 0.

Reuse Build Number

Ticking the "Reuse build number if previous build was discarded, stopped or failed while initialising" checkbox will cause the version counter to be decremented if a build was discarded by a configuration condition, or was stopped or failed while initialising.

The build number will not be decremented if the build failed or stopped while it was queued or running. It will also not be decremented if the version counter has been incremented by another build for this project which has started in the meantime.

Version Format String

The Version Format String sets how each build for configurations under this project should display its version. For example, the default Version Format String is set to 1.0.0.\$Build.BuildNumber\$ (The **\$Build.BuildNumber\$** expression will be substituted for the **Version Counter** value at runtime). This means that all builds for this project will have the following versioning:

- 1.0.0.1
- 1.0.0.2
- ...
- 1.0.0.7548, etc.

Say you are about to start work on version 1.1 of your project. All you would need to do is set your Version Format String to 1.1.0.\$Build.BuildNumber\$ which will then produce the following versioning:

- 1.1.0.7549
- 1.1.0.7550, etc.

[Expressions and Variables](#) can be used in the Version Format String. This allows you to include dynamic values into your versioning.

Overrides

Any configurations which are overriding these versioning settings will be listed at the bottom of the versioning section. A button is displayed allowing you to clear the overridden settings.

Repositories

The Repositories step allows you to link Version Control Systems to the current project. Head over to the [Repositories](#) page for more information on creating and editing repositories.

Variables

The Variables step allows you to create and edit project level variables. The [Variables](#) page includes an in depth guide to variables.

Security

The Security step allows you to specify project wide security settings. Any security settings that are defined at the project level will override global settings. For example, if a user has project read access but you grant them project edit status, they will be able to edit the current project but only view any other projects. Note that configuration security settings will override project security settings.

More information on creating and editing user permissions can be found in the [Access Control](#) section.

Note that when you set explicit permissions on the project you will only be able to set project, configuration and build permissions. All administration permissions are project independent and can only be set on global security settings.

Cleanup

The Cleanup step allows you to apply a build cleanup policy for all configurations in the project. For more information on cleanup policies, check out the [Cleanup](#) page.

Finished

The Finished step allows you to review the configurations that are attached to the project. From this page you can modify existing configurations that belong to this project, create a new configuration or create another project.

Projects

[✓ Go to New Project's Homepage](#)[✓ Create Another Project](#)

Configurations

[✓ Create a Configuration](#)

Choose an existing Configuration

Name	Actions
bitbucketHg	[Queue] [Edit] [Clone] [Delete]
CodeplexGit	[Queue] [Edit] [Clone] [Delete]
CodeplexHg	[Queue] [Edit] [Clone] [Delete]
github	[Queue] [Edit] [Clone] [Delete]
KilnTest	[Queue] [Edit] [Clone] [Delete]
kilnTestGit	[Queue] [Edit] [Clone] [Delete]