Variables

Before reading this page, it is highly recommended you read the Variables, Objects & Expressions page.

Variables

Variables store dynamic values which can be used during the build process. A full description and further conceptual information can be found on the Variables, Objects & Expressions concept page.

This page discusses information regarding the hands-on use of variables.

Additional Variable Information

- Variable Types
- Variable Prompt Types

Variable Types

Each variable must be assigned a type which limits what time of information can be stored in a variable. Note that all variable types, except expression variables, can have their values changed during the build process.

A full list of variable types can be found on the Variable Types page.

Variable Prompt Types

All configuration variables can be assigned a prompt type which enables users to set a variable's value when queuing a manual build.

Configuration Variables can be given the following prompt types:

- Text
- Memo
- Password
- Numeric
- Date Time
- Time Span
- Date
- Time
- Selection
- Multiple Selection
- Boolean

Read the Variable Prompt Types page for more information on these prompt types.

Variable Namespaces

Variable namespaces control where a variable exists within the Continua CI environment. Namespaces also provide variables with a scope which limits what each build can access. For example, if you create a configuration variable then only builds from that configuration can access that variable.

Continua uses a namespace hierarchy (as shown below) to determine which variable value should be returned when no namespace has been defined. Read the **Variable Namespace Hierarchy section** on the Variables, Objects & Expressions concept page for a closer look at how the Continua hierarchy works.

blocked URL

Build Variables

Build variables belong to a specific build however they cannot be created manually. They are used to override any of the variables listed below. See the next section for more information on how build variables work. Build variables can be accessed with the **%MyVariableName%** syntax (Note that you cannot use the Build namespace prefix).

When a build is created, it automatically creates build variables for every configuration variable that has been defined. It automatically sets the build variable's value to the same value as the configuration variable unless the value is overridden automatically with Triggers or manually through the queue dialog.

Configuration Variables

Configuration variables are created on a specific configuration and they can only be accessed by builds that belong to that configuration. Configuration Variables can be created, modified and deleted through the Configuration Wizard.

Configuration variables are server variables and can only be updated during the build by the Set Variable action set to update server variables or the Persist Build Variable build event handler. A write lock is required to update server variables during the build. The initial value of a configuration variable at the start of the build is copied to an agent variable and can be accessed with the **%Configuration.MyVariableName%** syntax. Agent variables can only be updated using the options to synchronise agent variables in the Set Variable and Load Variable actions.

Project Variables

Project variables are created on a specific project and they can only be accessed by configurations and builds that belong to that project. Project Variables can be created, modified and deleted through the Project Wizard.

Project variables are server variables and can only be updated during the build by the Set Variable action set to update server variables or the Persist Build Variable build event handler. A write lock is required to update server variables during the build. The initial value on a project variable at the start of the build is copied to an agent variable and can be accessed with the **%Project.MyVariableName%** syntax. Agent variables can only be updated using the options to synchronise agent variables in the Set Variable and Load Variable actions.

Application Variables

Application variables are system wide variables that every build, configuration and project can access. This means that any variable defined in the application namespace can be used anywhere in Continua CI. These are the highest variables that the user can define. Application Variables can be created, modified and deleted through the Variables page in the administration section.

Application variables are server variables and can only be updated during the build by the Set Variable action set to update server variables or the Persist Build Variable build event handler. A write lock is required to update server variables during the build. The initial value on a application variable at the start of the build is copied to an agent variable and can be accessed with the **%Application.MyVariableName%** syntax. Agent variables can only be updated using the options to synchronise agent variables in the Set Variable and Load Variable actions.

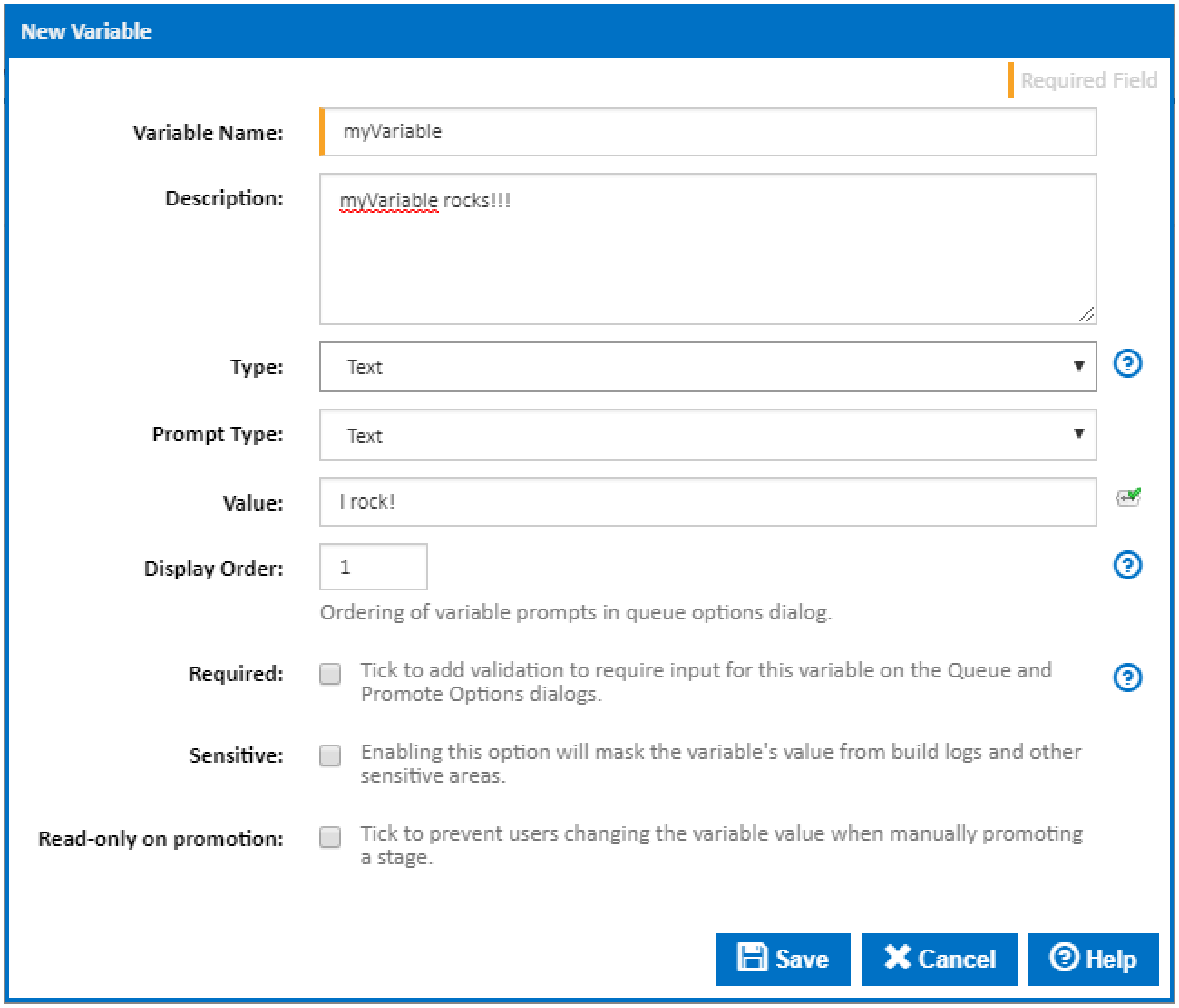
Environment Variables

Environment variables are created from the server's environment variables automatically and cannot be changed. These variables can be accessed system wide by every build. Environment variables are always read-only and they can be accessed with the **%Environment.MyVariableName%** syntax.

Creating and Editing Variables

Configuration, Project and Application variables are all created in the same way through their respective pages. So configuration variables are created through the Configuration Wizard, project variables are created through the Project Wizard and application variables are created through the variables page in the administration section.

All variables, regardless of their namespace, are created in the same way. The only exception are Configuration variables which can also be assigned a prompt type.



Variable Name: myVariable

Description: myVariable rocks!!!

Type: Text

Prompt Type: Text

Value: I rock!

Display Order: 1

Ordering of variable prompts in queue options dialog.

Required: Tick to add validation to require input for this variable on the Queue and Promote Options dialogs.

Sensitive: Enabling this option will mask the variable's value from build logs and other sensitive areas.

Read-only on promotion: Tick to prevent users changing the variable value when manually promoting a stage.

When defining a variable, you can specify the following properties:

Name

The name is what you will reference whenever you call this variable. In the screenshot above, I would reference this variable with the following syntax: **%myVariable%**

Description

This description is shown on all variables pages and it is also displayed on the Queue Build dialog as help text.

Type

Variables can be set to several different types that affect which values can be stored in the variable. The Variable Types page has a full list of types and their definitions.

Prompt Type - Configuration Variables Only

Prompt types determine what values the user can specify when a build is queued manually. The Variable Prompt Types page has a full list of prompt type and their definitions.

Value

This is the actual value that the variable will store. Note that all variables (except Build variables) will never change their value during the build process. Instead, if you change a variables value, it will create a build variable with the same name and the new value.

Sensitive

If this is ticked, the variable's value from build logs and other sensitive areas is masked.

Additional fields if a prompt type is selected:

Display Order

The order of variable prompts in the queue options dialog.

Required

Tick to add validation to require input for this variable on the Queue and Promote Options dialogs.

Read-only promotion

Tick to prevent users changing the variable value when manually promoting a stage.

Options - Dropdown select or Checkbox select only

Use a separate line for each selection option. Blank lines will be ignored.