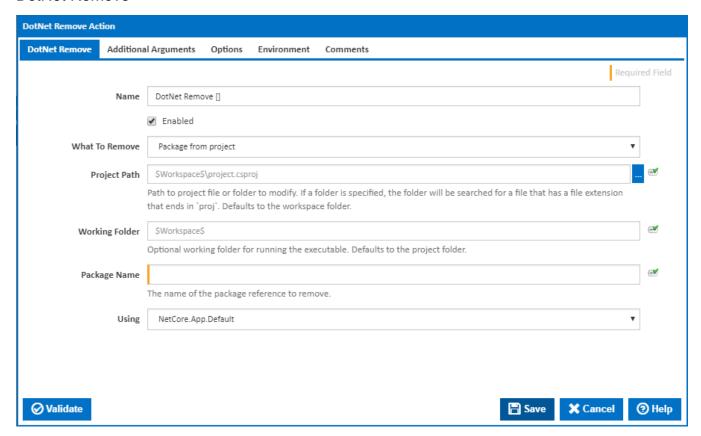# DotNet Remove

> (i) The DotNet Remove action in Continua CI is a wrapper around the .Net Core command line tools. If you're having trouble using the DotNet Build action, please refer to the .NET Core Command Line Tools documentation.

The DotNet Remove action can be used to remove a reference or package from a .Net project or a project from a .Net solution file.

DotNet Remove

DotNet Remove Action

| **DotNet Remove** | Additional Arguments | Options | Environment | Comments |

Required Field

Name | DotNet Remove []

☑ Enabled

What To Remove | Package from project ▾

Project Path | $Workspace$\project.csproj ...

Path to project file or folder to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in `proj`. Defaults to the workspace folder.

Working Folder | $Workspace$

Optional working folder for running the executable. Defaults to the project folder.

Package Name |

The name of the package reference to remove.

Using | NetCore.App.Default ▾

⊘ Validate   💾 Save   ✖ Cancel   ⊙ Help

Name

A friendly name for this action (will be displayed in the actions workflow area).

Enabled

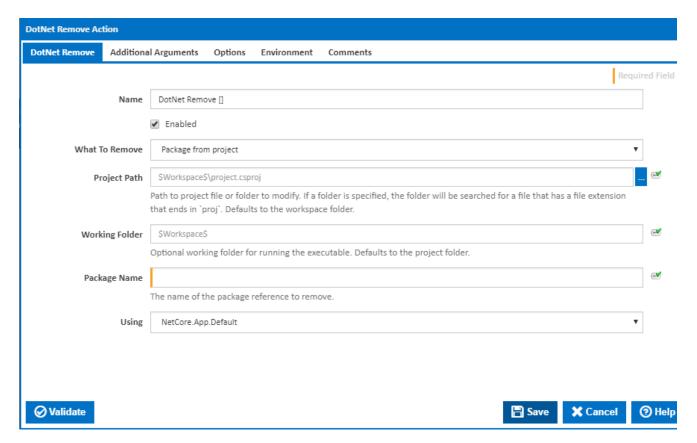Determines if this action will be run within the relevant stage.

What to Remove

Select what to want to remove from what. You can choose to remove one of the following:

- Package from project
- Reference from project
- Project from solution

Package from project

This option uses the "dotnet remove package" command line to remove a NuGet package reference from a project file.

DotNet Remove Action

DotNet Remove Additional Arguments Options Environment Comments

| Required Field

Name | DotNet Remove []

☑ Enabled

What To Remove | Package from project ▼

Project Path | $Workspace$\project.csproj
Path to project file or folder to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in `proj`. Defaults to the workspace folder.

Working Folder | $Workspace$
Optional working folder for running the executable. Defaults to the project folder.

Package Name |
The name of the package reference to remove.

Using | NetCore.App.Default ▼

⊘ Validate   💾 Save   ✖ Cancel   ⑦ Help

The following fields are shown when this option is selected.

Project Path

The path to project file or folder to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in "proj". If left empty, this will default to the workspace folder.
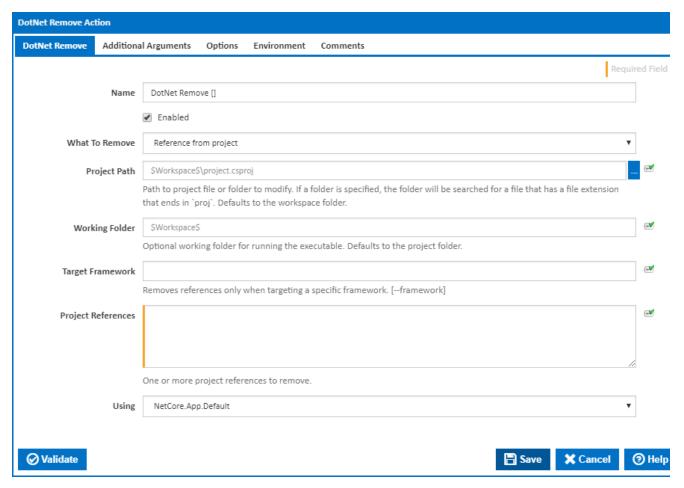
Working Folder

An optional working folder for running the executable. If left empty, this will default to the Project Path.

Package Name

The name of the package reference to remove.

Reference from project

This option uses the "dotnet remove reference" command line to remove project references from a project file.

The following fields are shown when this option is selected.

Project Path

The path to project file or folder to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in "proj". If left empty, this will default to the workspace folder.

Working Folder

An optional working folder for running the executable. If left empty, this will default to the Project Path.
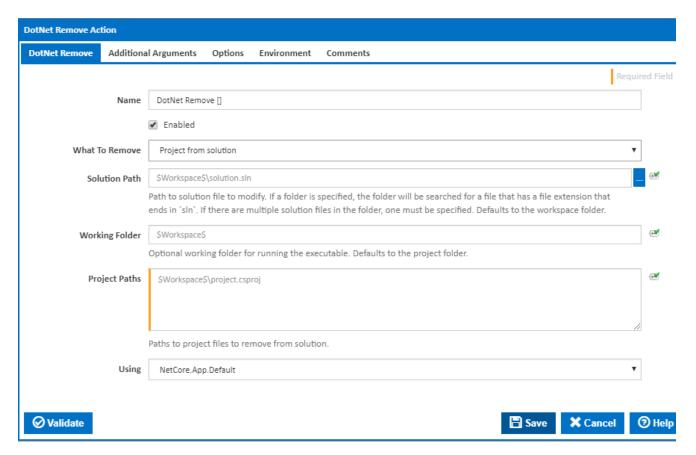
Target Framework

If the target framework name is entered, the project reference will only be removed from a project targeting the specified framework. [--framework]

Project References

Enter the path to one or more project references - one reference per line. These should generally be entered as relative paths - relative to the Working Folder.

Project from solution

This option uses the "dotnet sln .. remove" command line to remove project references from a solution file.

The following fields are shown when this option is selected.

Solution Path

The path to the solution file to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in `sln`. If there are multiple solution files in the folder, one must be specified. If left empty, this will default to the workspace folder.

Working Folder

An optional working folder for running the executable. If left empty, this will default to the Solution Path.

Project Paths

Add the path to one or more project files - one file per line. This should generally be entered as relative paths - relative to the Working Folder.
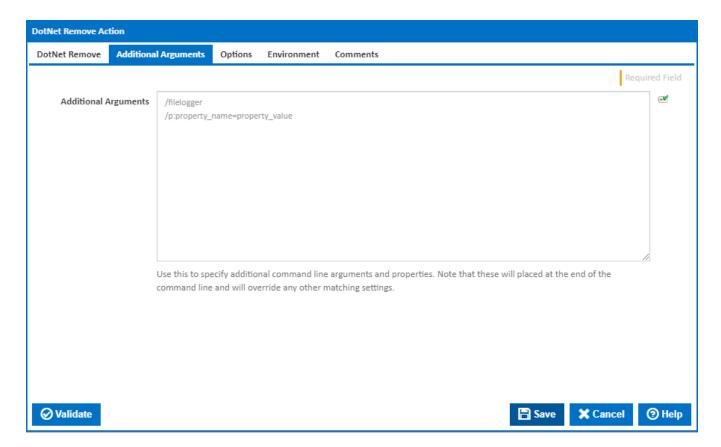
Using

The Using drop down is populated with any property collector whose namespace matches the pattern defined by the DotNet CLI actions. The pattern for this action is `^DotNet.Cli.*`

If you create a property collector for this action, make sure you select the **Path Finder PlugIn** type and give it a name that will match the pattern above in blue. Example names listed here, search the table's Plugin column for "**DotNet Remove**".

For more in-depth explanations on property collectors see Property Collectors.

Alternatively, you can select the **Custom** option from the Using drop down list and specify a path in the resulting input field that will be displayed. Please read Why it's a good idea to use a property collector before using this option.
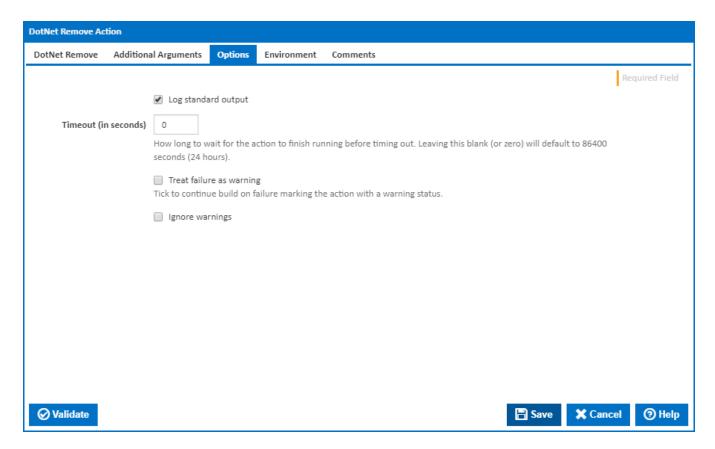
Additional Arguments

DotNet Remove | **Additional Arguments** | Options | Environment | Comments

Required Field

Additional Arguments

```
/filelogger
/p:property_name=property_value
```

Use this to specify additional command line arguments and properties. Note that these will placed at the end of the command line and will override any other matching settings.

⊘ Validate   🖫 Save   ✖ Cancel   ⊙ Help

Additional Arguments

Use this to specify additional MSBuild command line arguments and properties. Note that these will placed at the end of the command line and will override any other matching settings.

Options

Log standard output

Tick to log standard output.

Timeout (in seconds)

How long to wait for the action to finish running before timing out. Leaving this blank (or zero) will default to 86400 seconds (24 hours).
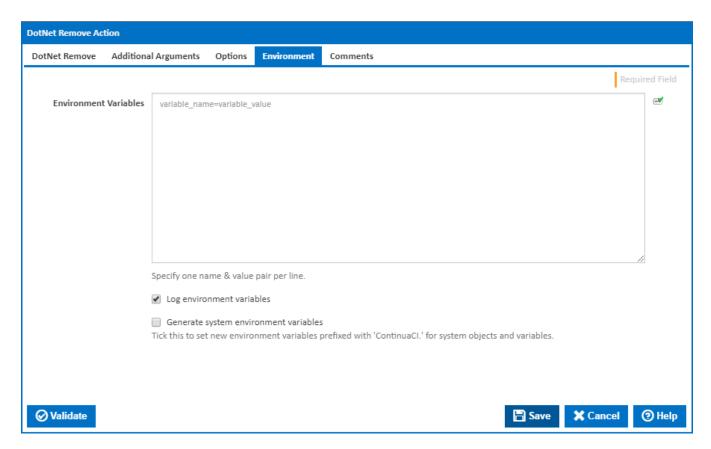
Treat failure as warning

Tick to continue build on failure, marking the action with a warning status.

Ignore warnings

If this is ticked, any warnings logged will not mark the action with a warning status.

Environment

Environment Variables

Multiple environment variables can be defined - one per line. These are set before the command line is run.

Log environment variables

If this is ticked, environment variable values are written to the build log.

Generate system environment variables

Tick this checkbox to set up a list of new environment variables prefixed with 'ContinuaCI.' for all current system expression objects and variables.

Mask sensitive variable values in system environment variables

This checkbox is visible only if the '**Generate system environment variables**' checkbox is ticked.

If this is ticked, the values of any variables marked as sensitive will be masked with **** when setting system environment variables. Clear this to expose the values.