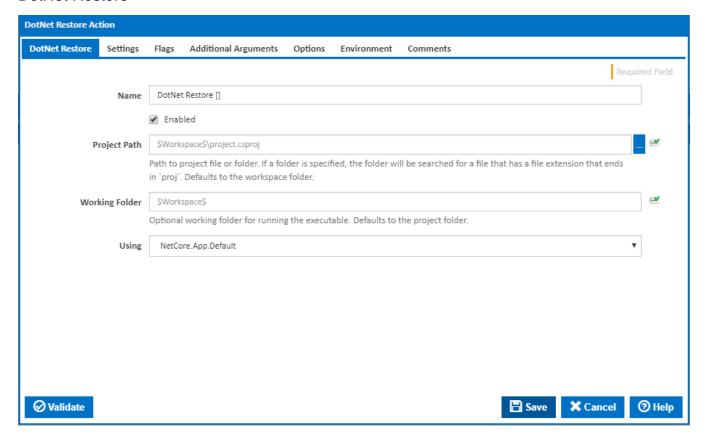
DotNet Restore Action



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

The DotNet Restore action is used to restores the dependencies and tools of a .Net Core project.

DotNet Restore



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.

Project Path

Path to project file or folder. 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

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

Using

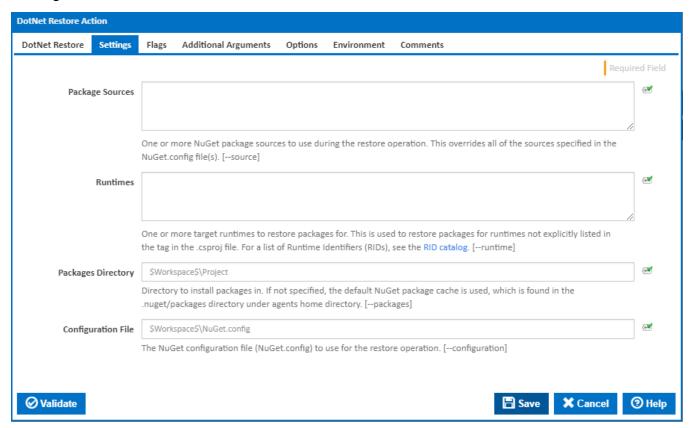
The Using drop down is populated with any property collector whose namespace matches the pattern defined by the DotNet CLI actions. The pattern for this action is ^DotNet.Cli.*

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 Restore**".

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.

Settings



Package Sources

One or more NuGet package sources to use during the restore operation. This overrides all of the sources specified in the NuGet.config file(s). [--source]

Runtimes

One or more target runtimes to restore packages for. This is used to restore packages for runtimes not explicitly listed in the tag in the .csproj file. For a list of Runtime Identifiers (RIDs), see the RID catalog. [--runtime]

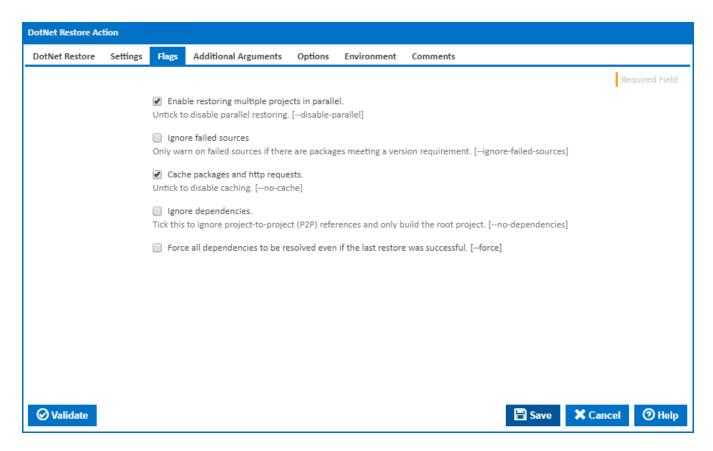
Packages Directory

Directory to install packages in. If not specified, the default NuGet package cache is used, which is found in the .nuget/packages directory in the user's home directory/. [--packages]

Configuration File

The NuGet configuration file (NuGet.config) to use for the restore operation. [--configuration]

Flags



Enable restoring multiple projects in parallel

If this is ticked, multiple restore operations are run concurrently. [--disable-parallel]

Ignore failed sources

If this is ticked, DotNet Restore will only warn on failed sources if there are packages meeting a version requirement. [--ignore-failed-sources]

Cache packages and HTTP requests

If this is ticked, packages and HTTP requests are cached. [--no-cache]

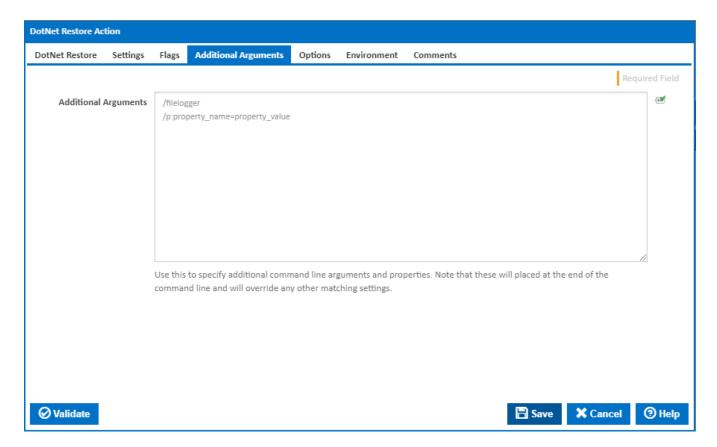
Ignore dependencies

Tick this to ignore project-to-project (P2P) references and only build the root project. [--no-dependencies]

Force all Dependencies to be resolved even if the last restore was successful.

Set this flag to force all dependencies to be resolved even if the last restore was successful. It is the same as deleting the project.assets.json file. [--force]

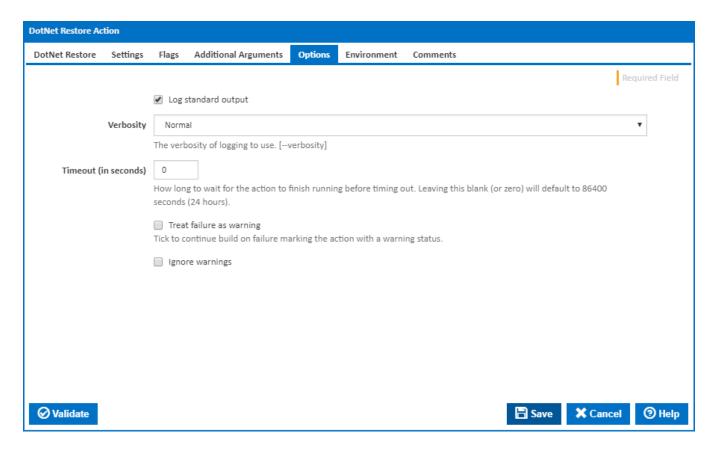
Additional Arguments



Additional Arguments

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

Options



Log standard output

If this is ticked, the command line output is written to the build log.

Verbosity

The amount of information detail to display in the build log. [--verbosity]

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).

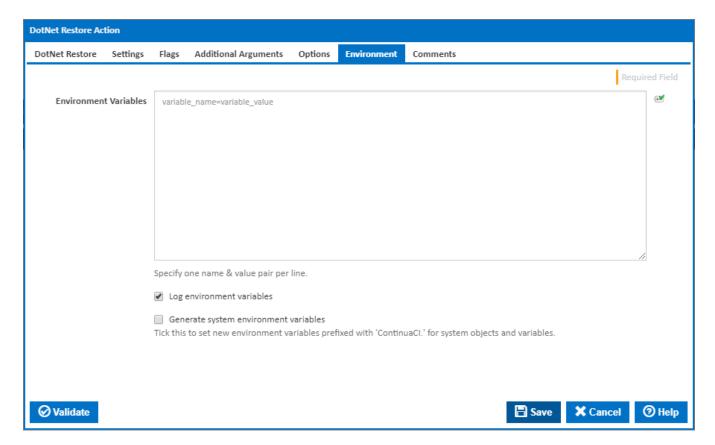
Treat failure as warning

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.