

DotNet Pack Action



The DotNet Pack action in Continuum CI is a wrapper around the .Net Core command line tools. If you're having trouble using the DotNet Pack action, please refer to the [.NET Core Command Line Tools documentation](#).

The DotNet Pack action is used to create NuGet packages from .Net Core projects.

DotNet Pack

DotNet Pack Action

DotNet Pack

Settings

Flags

Additional Arguments

Options

Environment

Comments

Required Field

Name

DotNet Pack []

☒

Enabled

Project Path

\$Workspace\$\project.csproj

...

Path to project file or folder. If a folder is specified, it 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.

Output Directory

Directory in which to place the built packages. [--output]

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.

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.

Output Directory

Directory in which to place the built packages. [--output]

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 Pack"**.

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

DotNet Pack Action

DotNet Pack

Settings

Flags

Additional Arguments

Options

Environment

Comments

Required Field

Configuration

Debug

Configuration to use when building the project. Defaults to "Debug" if left empty. [--configuration]

Version Suffix

The version suffix for an asterisk (*) in the version field of the project file. The format follows NuGet's version guidelines. [--version-suffix]

Runtime

Target runtime to restore packages for. For a list of Runtime Identifiers (RIDs), see the [RID catalog](#). [--runtime]

Validate

Save

Cancel

Help

Configuration

The configuration under which to build. This defaults to "Debug" if left empty. [--configuration]

Version Suffix

The version suffix for an asterisk (*) in the version field of the project file. The format follows NuGet's version guidelines. [--version-suffix]

Runtime

Target runtime to build for. For a list of Runtime Identifiers (RIDs), see the [RID catalog](#). [--runtime]

Flags

DotNet Pack Action

DotNet Pack

Settings

Flags

Additional Arguments

Options

Environment

Comments

Required Field

☒ No build

When this flag is set, the build phase of the packing process is skipped. [--no-build]

☐ Include symbols

Set this flag to generate symbol files in the NuGet package. [--include-symbols]

☐ Include source

Set this flag to include the source files in the NuGet package. The sources files are included in the src folder within the package. [--include-source]

☐ Serviceable

Set the serviceable flag in the package. [--serviceable]

☐ No dependencies

Set this flag to ignore project-to-project references and only restore the root project. [--no-dependencies]

☐ No restore

Set this flag to skip running an implicit restore during build. [--no-restore]

☐ Force all dependencies to be resolved even if the last restore was successful. [--force]

Validate

Save

Cancel

Help

No build

Set this flag to skip the building phase of the packing process. [--no-build]

Include Symbols

Set this flag to generate symbol files in the NuGet package. [--include-symbols]

Include Source

Set this flag to include the source files in the NuGet package. The sources files are included in the src folder within the package. [--include-source]

Serviceable

If this is ticked, the package is marked as serviceable. [--serviceable]

No dependencies

Set this flag to ignore project-to-project references and only build the root project. [--no-dependencies]

No Restore

Set this flag to skip running an implicit restore during build. [--no-restore]

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

DotNet Pack Action

DotNet Pack

Settings

Flags

Additional Arguments

Options

Environment

Comments

Additional Arguments

Required Field

/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

DotNet Pack Action

DotNet Pack

Settings

Flags

Additional Arguments

Options

Environment

Comments

Log standard output

Verbosity

Normal

The verbosity of logging to use. [--verbosity]

Timeout (in seconds)

0

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

Validate

Save

Cancel

Help

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

DotNet Pack Action

DotNet Pack

Settings

Flags

Additional Arguments

Options

Environment

Comments

Environment Variables

Required Field

variable_name=variable_value

Specify one name & value pair per line.

☒ Log environment variables

☐ Generate system environment variables

Tick this to set new environment variables prefixed with 'ContinuaCl.' for system objects and variables.

Validate

Save

Cancel

Help

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 'ContinuaCl.' 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.