NuGet Pack Action

> ⓘ The NuGet actions in Continua are a wrapper around the nuget command line. If you're having trouble using any of the NuGet actions, please refer to the Command Line Reference.

The NuGet Pack action is used to create a NuGet Package File (.nupkg).

NuGet Pack

NuGet Pack Action

| NuGet Pack | Build | Properties | Filtering | Settings | Options | Environment | Comments |

Required Field

Name NuGet Pack []

☑ Enabled

Project/NuSpec File $Source.YourRepo$\nuget\myproj.nuspec

The path to the project or NuSpec file.

Output Directory $Workspace$\nuget\

Specify the directory for the created NuGet package file. [-OutputDirectory]

Base Path $Source.YourRepo$\nuget\

The base path of the files defined in the nuspec file. [-BasePath]

Version 2.0

Overrides the version number from the nuspec file. [-Version]

Command Line Version Latest ▼

Using NuGet.Default ▼

⊘ Validate   🖺 Save   ✖ Cancel   ⑦ 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/NuSpec File

The location of the .nuspec file. For further information on creating these files, please see the NuSpec documentation.

Output directory

The location to place the resultant .nupkg files. [-OutputDirectory]

Base Path

The base path of the files defined in the nuspec file (optional). [-BasePath]

Version

The version of the package. This value will overwrite the version number provided in the nuspec file. This can be very useful for allocating version numbers based off build variables (for example $Build.Version$). [-Version]

Command Line Version

Select the version of the NuGet command line that is installed on the agent. Some other settings and options may be unavailable depending which the command line version is selected.

Using

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

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 "**NuGet 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.

Build

Build project

Determines if the project should be built before building the package. [-Build]

Configuration

The configuration to use when packaging. This option is relevant when building a .proj file. [-Properties Configuration=]

MSBuild Version

The version of MSBuild to use with this command. Leave blank to default to the MSBuild executable in your path, or otherwise the highest installed version of MSBuild. [-MSBuildVersion].

MSBuild Path

The path to the MSBuild executable to use with the command. This takes precedence over the MSBuild Version. [-MSBuildPath].

No package analysis

Skip package analysis (running of package rules). At the time of writing this includes things like checking for: Invalid Framework Folders, Misplaced Assemblies, Misplaced Script Files, Missing package descriptions etc. [-NoPackageAnalysis]

Properties

Properties

A list of properties that override values in the project file. [-Properties]

Filtering

NuGet Pack Action

NuGet Pack   Build   Properties   **Filtering**   Settings   Options   Environment   Comments

Exclude Patterns

```
*.exe
*.bat
```

Specify one or more wildcard patterns, one per line, to match files to exclude when creating the package. [-Exclude]

☐ Prevent default exclusions

Prevent default exclusion of NuGet package files and files and folders starting with a dot, such as .svn and .gitignore. [-NoDefaultExcludes]

☐ Exclude empty directories

Prevent inclusion of empty directories when building the package. [-ExcludeEmptyDirectories]

⊘ Validate   💾 Save   ✕ Cancel   ⊙ Help

Exclude Patterns

Specify one or more wildcard patterns, one per line, to match files to exclude when creating the package. [-Exclude]

Prevent default exclusions

Prevent default exclusion of NuGet package files and files and folders starting with a dot, such as .svn and .gitignore. [-NoDefaultExcludes]

Exclude empty directories

Prevent inclusion of empty directories when building the package. [-ExcludeEmptyDirectories]

Settings

Configuration File

A specific NuGet configuration file to use. Leave blank to use %AppData%\NuGet\nuget.config. [-ConfigFile]

Version Suffix

A suffix to append to the internally generated version number. This is typically used for appending build or other pre-release identifiers. [-Suffix].

Min Client Version

Specify the minimum version of the NuGet client that can install the created package. [-MinClientVersion]

The package contains sources and symbols

Specifies that the package contains sources and symbols. [-Symbols]

Symbol Package Format

This field is visible only if the '**The package contains sources and symbols**' checkbox is selected.

Select the symbol package format. Defaults to legacy symbol package format 'symbols.nupkg'. [-SymbolPackageFormat]

Use the tool folder

Determines if the output files of the project should be put in the tool folder. [-Tool]

Include referenced projects

Include referenced projects either as dependencies or as part of the package. If a referenced project has a corresponding nuspec file that has the same name as the project, then that referenced project is added as a dependency. Otherwise, the referenced project is added as part of the package. [-IncludeReferencedProjects]

Install the package to the output directory

Prepare the package output directory to support share as feed. [-InstallPackageToOutputPath]

Output filenames without version

Prepare the package output name without the version. [-OutputFileNamesWithoutVersion]

Options

Verbosity

How much information should the command line output? [-Verbosity]

Force English output

Forces the command line to run using an invariant, English-based culture. [-ForceEnglishOutput]

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.