DotNet Publish Action

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

The DotNet Publish action is used to packs the .Net Core application and all of its dependencies into a folder getting it ready for publishing.

DotNet Publish

DotNet Publish Action

| DotNet Publish | Settings | Flags | Additional Arguments | Options | Environment | Comments |

Required Field

Name DotNet Publish []

☑ 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 published artifacts. [--output]

Manifests

Specify one or more paths of target manifest files to use to trim the set of packages published with the app. [--manifest]

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 published artifacts. If left empty, it will default to ./bin/[configuration]/[framework]/ for a framework-dependent deployment or ./bin/[configuration]/[framework]/[runtime] for a self-contained deployment. [--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 Publish**".

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 Publish Action

| DotNet Publish | **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]

Framework

Specify the framework to publish for. The framework needs to be defined in the project file. [--framework]

Runtime

Target runtime to publish for. This is used when creating a self-contained deployment (SCD). For a list of Runtime Identifiers (RIDs), see the RID catalog. Default is to publish a framework-dependent deployment (FDD). [--runtime]

Version Suffix

Defines what * should be replaced with in the version field in the project file. The format follows NuGet's version guidelines. [--version-suffix]

⊘ Validate   💾 Save   ✕ Cancel   ⑦ Help

Configuration

The configuration to use when building the project. This defaults to "Debug" if left empty. [--configuration]
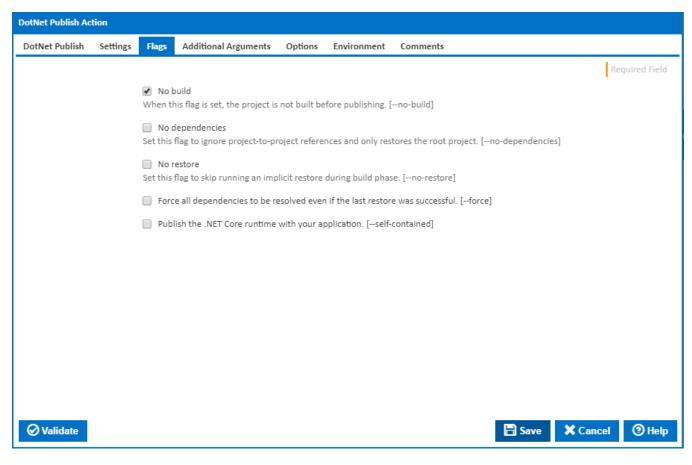
Framework

The name of the framework to compile for. The framework must also be defined in the project.json file. [--framework]

Runtime

Target runtime to publish for. This is used when creating a self-contained deployment (SCD). For a list of Runtime Identifiers (RIDs), see the RID catalog. Default is to publish a framework-dependent deployment (FDD). [--runtime]

Version Suffix

This can be used to replace a wildcard * in the version field in the project.json file. The format follows NuGet's version guidelines. [--version-suffix]

Flags

DotNet Publish   Settings   **Flags**   Additional Arguments   Options   Environment   Comments

Required Field

☑ No build
When this flag is set, the project is not built before publishing. [--no-build]

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

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

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

☐ Publish the .NET Core runtime with your application. [--self-contained]

⊘ Validate   💾 Save   ✖ Cancel   ⑦ Help

No Build

When this flag is set, the project is not built before publishing. [--no-build]

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]

Publish the .NET Core runtime with your application

Set this flag to publish the .NET Core runtime with your application so the runtime doesn't need to be installed on the target machine. For more information about the different deployment types, see .NET Core application deployment. [--self-contained]

Additional Arguments

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.

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.