

DotNet Run Action



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

The DotNet Run action provides a convenient way to run a .Net Core application from the source code with one command. It compiles source code, generates an output program and then runs that program.

DotNet Run

DotNet Run Action

DotNet Run Settings Flags Additional Arguments Options Environment Comments

Name

DotNet Run []

☒ Enabled

Project Path

\$Workspace\$\project.csproj

Path to project file or folder to build and run. 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. [--project]

Working Folder

\$Workspace\$

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

Using

NetCore.App.Default

Required Field

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 to build and run. 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. [--project]

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 Run**".

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 Run Action

DotNet Run

Settings

Flags

Additional Arguments

Options

Environment

Comments

Required Field

Configuration

Release

Configuration under which to build. Defaults to Debug if left empty. [--configuration]

Framework

Framework to build for. The target framework has to be specified in the project file. [--framework]

Runtime

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

☐ Ignore launch profile

Tick to skip reading a launch profile from launchSettings.json. [--no-launch-profile]

Launch Profile

The name of the launch profile (if any) to use when launching the application. Launch profiles are defined in the launchSettings.json file and are typically called Development, Staging and Production. [--launch-profile]

Application Arguments

Arguments to be passed to the application being run. [--]

Validate

Save

Cancel

Help

Configuration

The configuration under which to build. Defaults to Release 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 build for. For a list of Runtime Identifiers (RIDs), see the [RID catalog](#). [--runtime]

Ignore launch profile

Tick to skip reading a launch profile from launchSettings.json. [--no-launch-profile]

Launch Profile

This field is visible only if the '**Ignore launch profile**' checkbox is not ticked. The name of the launch profile (if any) to use when launching the application. Launch profiles are defined in the launchSettings.json file and are typically called Development, Staging and Production. [--launch-profile]

Application Arguments

Arguments to pass to the application being run. [--]

Flags

DotNet Run Action

DotNet Run

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]

Validate

Save

Cancel

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]

Additional Arguments

DotNet Run Action

DotNet Run

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 be 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 be placed at the end of the command line and will override any other matching settings.

Options

DotNet Run Action

DotNet Run

Settings

Flags

Additional Arguments

Options

Environment

Comments

Required Field

☒ 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 Run Action

DotNet Run

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 'ContinuaCI.' 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 '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.