

# DotNet Add



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

The DotNet Add action can be used to add a reference or package to a .Net project or a project to a .Net solution file.

## DotNet Add

**DotNet Add Action**

DotNet Add

Settings

Additional Arguments

Options

Environment

Comments

Required Field

Name

DotNet Add []

☒ Enabled

What To Add

Package to project

Project Path

\$Workspace\$\project.csproj

Path to project file or folder to modify. 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

\$Workspace\$

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

Target Framework

Adds references only when targeting a specific framework. [--framework]

Package Name

The name of the package reference to add.

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.

### What to Add

Select what to want to add to what. You can choose to add one of the following:

- Package to project
- Reference to project
- Project to solution

#### Package to project

This option uses the "dotnet add package" command line to add a package reference to a project file. The command starts with a compatibility check to ensure the package is compatible with the frameworks in the project. If the check passes, a <PackageReference> element is added to the project file and [dotnet restore](#) is run.

DotNet Add Action

DotNet Add

Settings

Additional Arguments

Options

Environment

Comments

Required Field

Name

DotNet Add []

☒ Enabled

What To Add

Package to project

Project Path

\$Workspace\$\project.csproj

...

Path to project file or folder to modify. 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

\$Workspace\$

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

Target Framework

Adds references only when targeting a specific framework. [--framework]

Package Name

The name of the package reference to add.

Using

NetCore.App.Default

Validate

Save

Cancel

Help

The following fields are shown when this option is selected.

### Project Path

The path to project file or folder to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in ".proj". If left empty, this will default to the workspace folder.

### Working Folder

An optional working folder for running the executable. If left empty, this will default to the Project Path.

### Target Framework

If the target framework name is entered, the package reference will only be added to a project targeting the specified framework. [--framework]

### Package Name

The name of the package reference to add.

### Reference to project

This option uses the "dotnet add reference" command line to add project references to a project file. The command adds one or more <ProjectReference> elements to the project file.

DotNet Add Action

DotNet Add

Additional Arguments

Options

Environment

Comments

Required Field

Name

DotNet Add []

☒ Enabled

What To Add

Reference to project

Project Path

\$Workspace\$\project.csproj

Path to project file or folder to modify. 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

\$Workspace\$

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

Target Framework

Adds references only when targeting a specific framework. [--framework]

Project References

One or more project references to add.

Using

NetCore.App.Default

Validate

Save

Cancel

Help

The following fields are shown when this option is selected.

### Project Path

The path to project file or folder to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in ".proj". If left empty, this will default to the workspace folder.

### Working Folder

An optional working folder for running the executable. If left empty, this will default to the Project Path.

### Target Framework

If the target framework name is entered, the project reference will only be added to a project targeting the specified framework. [--framework]

### Project References

Enter the path to one or more project references - one reference per line. These should generally be entered as relative paths - relative to the Working Folder.

### Project to solution

This option uses the "dotnet sln .. add" command line to add project references to a solution file.

DotNet Add Action

DotNet Add
Additional Arguments
Options
Environment
Comments

Required Field

Name
DotNet Add []

☒ Enabled

What To Add
Project to solution

Solution Path
\$Workspace\$\solution.sln

Path to solution file to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in `sln`. If there are multiple solution files in the folder, one must be specified. Defaults to the workspace folder.

Working Folder
\$Workspace\$

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

Project Paths
\$Workspace\$\project.csproj

Paths to project files to add to solution.

Using
NetCore.App.Default

Validate
Save
Cancel
Help

The following fields are shown when this option is selected.

## Solution Path

The path to the solution file to modify. If a folder is specified, the folder will be searched for a file that has a file extension that ends in `sln`. If there are multiple solution files in the folder, one must be specified. If left empty, this will default to the workspace folder.

## Working Folder

An optional working folder for running the executable. If left empty, this will default to the Solution Path.

## Project Paths

Add the path to one or more project files - one file per line. This should generally be entered as relative paths - relative to the Working 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 Add".

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

This tab is only visible if 'Package to project' is selected in the 'What To Add' field within the 'DotNet Add' tab.

DotNet Add Action

DotNet Add

Settings

Additional Arguments

Options

Environment

Comments

Required Field

Package Version

Release

The version of the package to add. [--version]

Package Source

The NuGet package source to use during the restore operation. [--source]

☐ Skip restore preview and compatibility check. [--no-restore]

Package Directory

Directory to restore the package to. [--package-directory]

Validate

Save

Cancel

Help

### Package Version

The version of the package to add. [--version]

### Package Source

The NuGet package source to use during the restore operation. [--source]

### Skip restore preview and compatibility check

Tick to add the package without performing a restore preview and compatibility check. [--no-restore]

### Package Directory

A directory to restore the package to. [--package-directory]

## Additional Arguments

DotNet Add Action

DotNet Add

Settings

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 Add Action

DotNet Add

Settings

Additional Arguments

Options

Environment

Comments

Log standard output

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

Required Field

Validate

Save

Cancel

Help

### Log standard output

Tick to log standard output.

### 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 Add Action

DotNet Add

Settings

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.