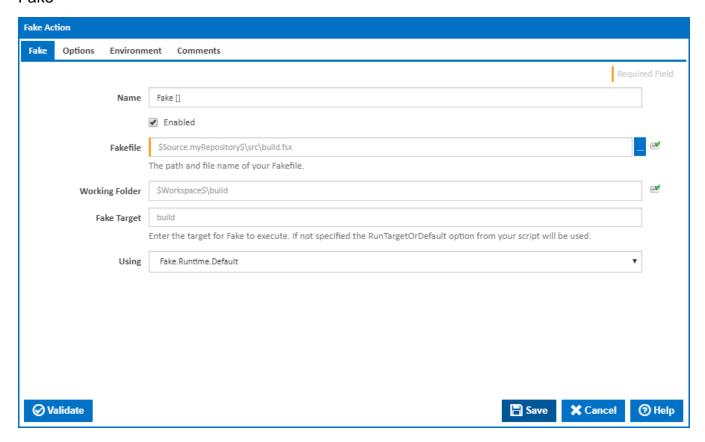# Fake Action

> (i) The Fake action in Continua is a wrapper around the fake.exe command line. If you're having trouble using the Fake action, please refer to the [Command Line Reference](#).

The Fake action runner allows users to extend the functionality of Continua by providing integration with FAKE. "FAKE - F# Make" is a build automation system with capabilities which are similar to make and rake. Fake uses an easy domain-specific language (DSL) so that you can start using it without specifically learning F#.

For more information about the project and documentation please visit the [FAKE Project Website](#).

Fake

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.

Fake File

Optional - Specify the path and name of the Fake file to be used. If no file is specified, then Fake will search the working folder for a file named build.sfx.

Working Folder

Optional - Specify the working folder (relative to the build workspace). If not specified, then the build workspace folder is used. This folder should generally be where the Fake file lives, unless the Fake file is specified.

Fake Target

Optional - The build target which this action should run against. If no target is specified the target designated within your "RunTargetOrDefault" parameter (located within your script file) will be used.
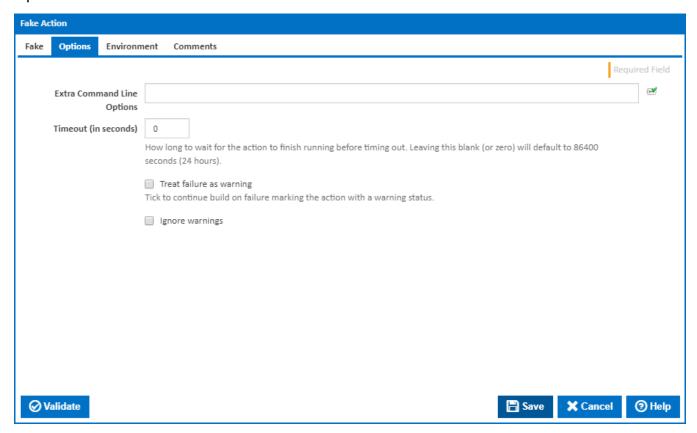
Using

The Using drop down is populated by any property collector properties whose namespace matches the pattern defined by the Fake action. The pattern for this action is `^Fake\..*`. The default property collector searches the environment path for "fake.exe".

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 "**Fake**".

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.

Options

Fake Action

| Fake | **Options** | Environment | Comments |

Required Field

Extra Command Line Options

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

Extra Command Line Options

Optional - Additional command line arguments which will be passed to Fake.

Timeout (in seconds)

How many seconds to wait for the action to finish before timing out. The default is 86400 seconds (24 hours).
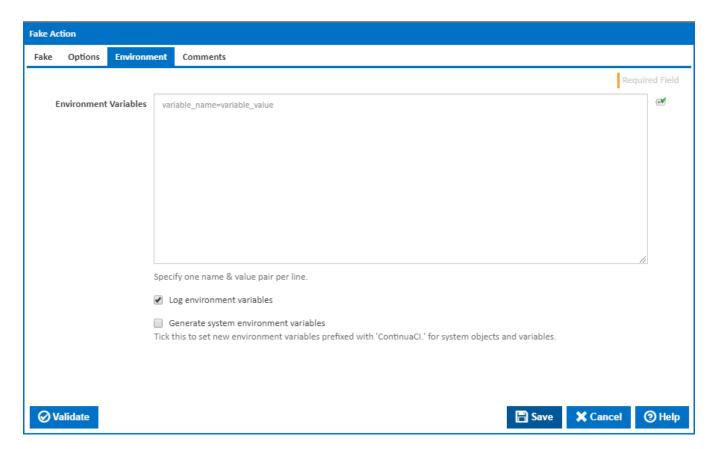
Treat failure as warning

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.

An example Fake File

build.fsx

```fsharp
// include Fake lib
#r @"C:\Program Files (x86)\fake\tools\FakeLib.dll"

open Fake
// Default target
Target "Default" (fun _ ->
    trace "Hello World from FAKE"
)
// Default target
Target "superhappy test" (fun _ ->
    trace "super happy test from FAKE"
)
// start build
RunTargetOrDefault "Default"
```