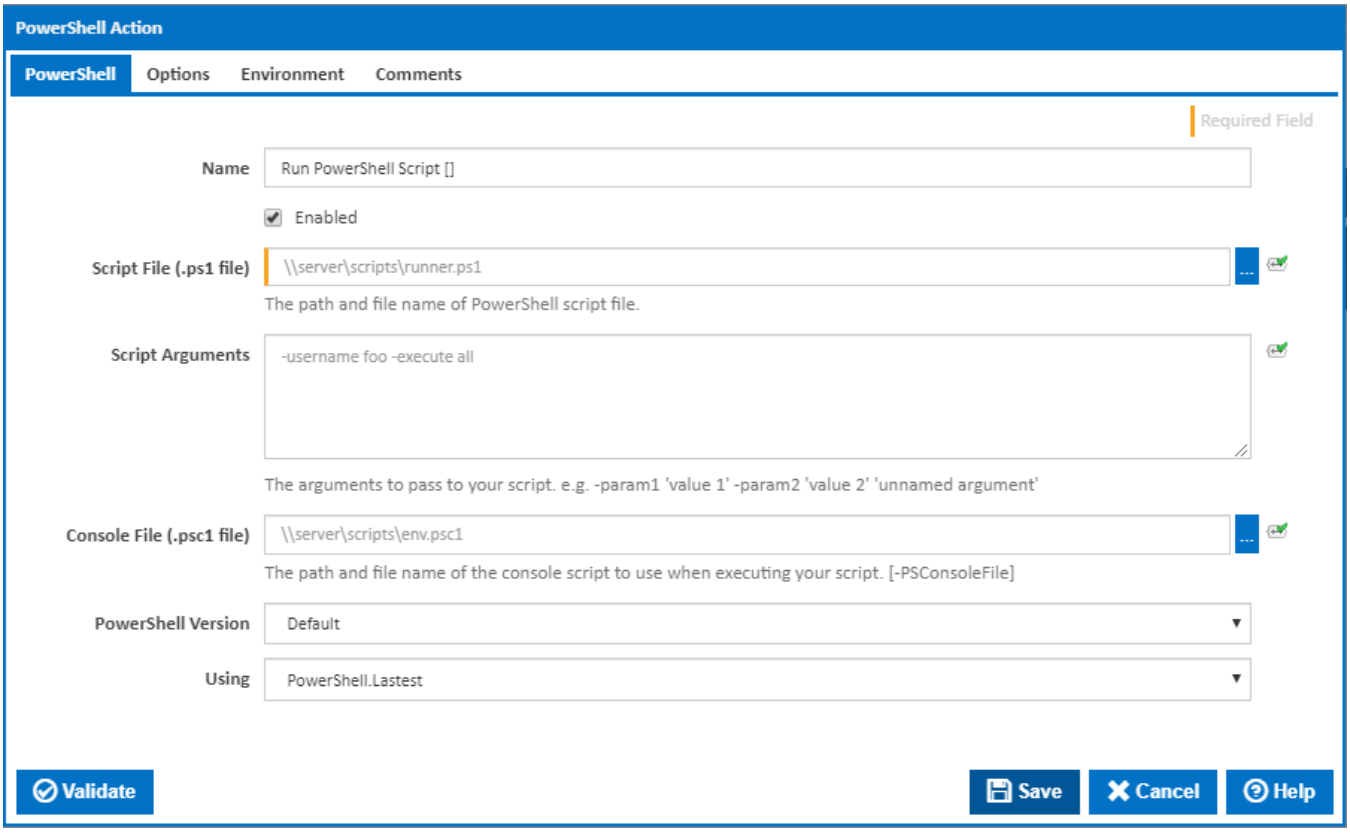
PowerShell Action

The PowerShell action in Continua is a wrapper around the PowerShell command line. If you're having trouble using the PowerShell action, please refer to the Command Line Reference.

PowerShell



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.

Script File (.ps1 file)

The path and file name of the PowerShell script you want to execute. [-Command]

Script Arguments

The arguments you want to pass to your script. e.g. -username fflintstone -workspace '$Workspace'.

Note: Parameter values containing spaces (or expanding to values containing spaces) must be delimited with single quotes.

Console File (.psc1 file)

The path and file name of the console script you want pass to PowerShell. This setting is optional. [-PSConsoleFile]

PowerShell Version

Adds the -Version argument to the PowerShell command line. [-Version]

Using

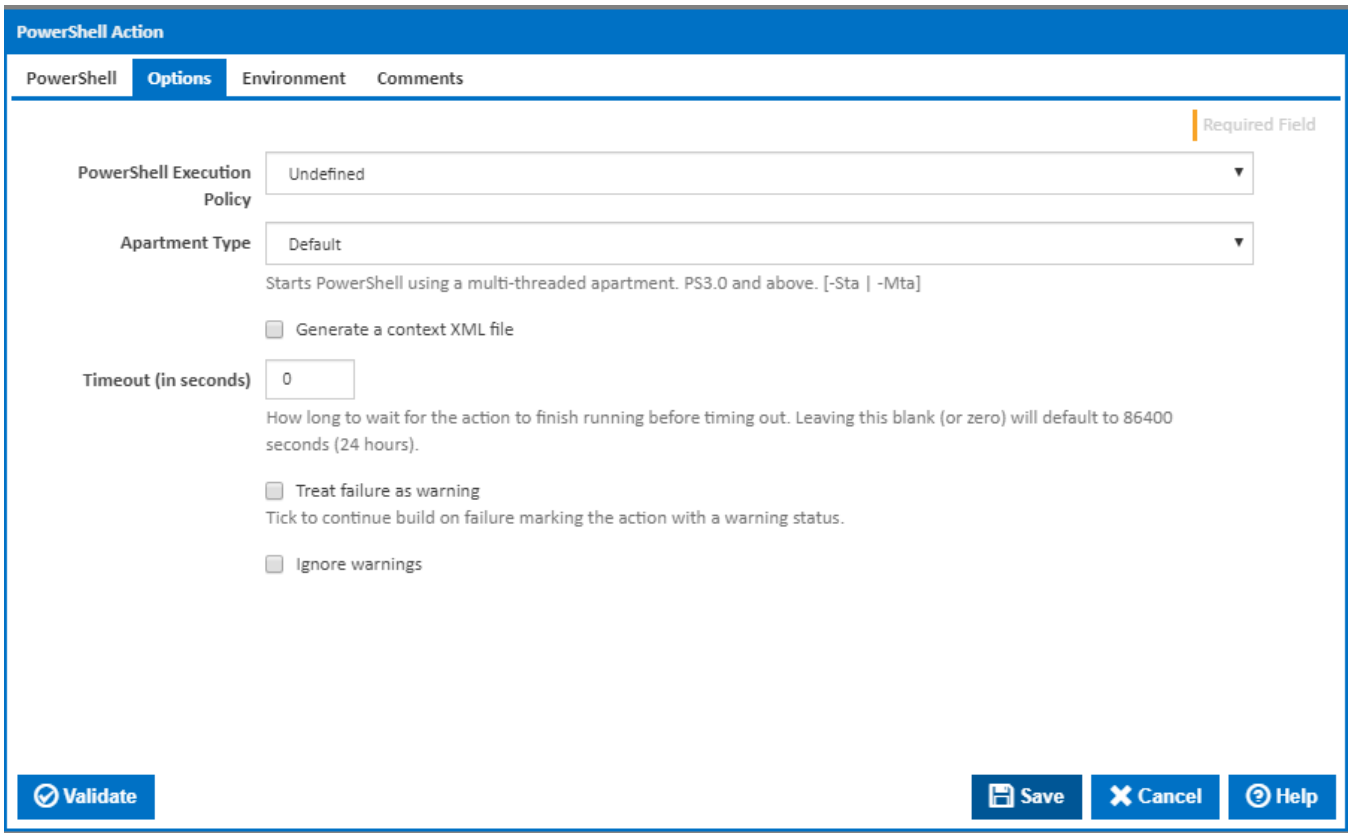
The Using drop down is populated with any property collector whose namespace matches the pattern defined by the PowerShell action. The pattern for this action is `^PowerShell.*`

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 "**PowerShell**".

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



PowerShell Execution Policy

Add an -ExecutionPolicy argument to the PowerShell command line. Options are:

- **Restricted**: Does not load configuration files or run scripts. Restricted is the default execution policy.
- **All signed**: Requires that all scripts and configuration files be signed by a trusted publisher, including scripts that you write on the local computer**.**
- **Remote signed**: Requires that all scripts and configuration files downloaded from the Internet be signed by a trusted publisher.
- **Unrestricted**: Loads all configuration files and runs all scripts. If you run an unsigned script that was downloaded from the Internet, you are prompted for permission before it runs.
- **Bypass**: Nothing is blocked and there are no warnings or prompts.
- **Undefined**: No -ExecutionPolicy argument is added. Default execution policy is used.

Apartment Type

Adds the -Sta or -Mta parameters to the PowerShell command line to run the script using a single-threaded or multi-threaded apartment.

Generate a context XML file

Generate an XML file containing all the available expressions and build variables. You can read this file using PowerShell commands such as : `[xml]$Xml Document = Get-Content ($workspace + "/ContinuaContext.xml")`

Context XML File Path

Visible only if the '**Generate a context XML file**' checkbox is ticked.

The path to the context XML file.

Remove context XML file on completion

Visible only if the '**Generate a context XML file**' checkbox is ticked.

If this ticked, the context XML file would be removed on completion.

Save sensitive variable values to context XML file

Visible only if the '**Generate a context XML file**' checkbox is ticked.

All build variables are written to the context file. By default, the values for variables marked as sensitive are masked with asterisks. If this option is ticked then all variable values are stored in the XML file as plain text.

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).

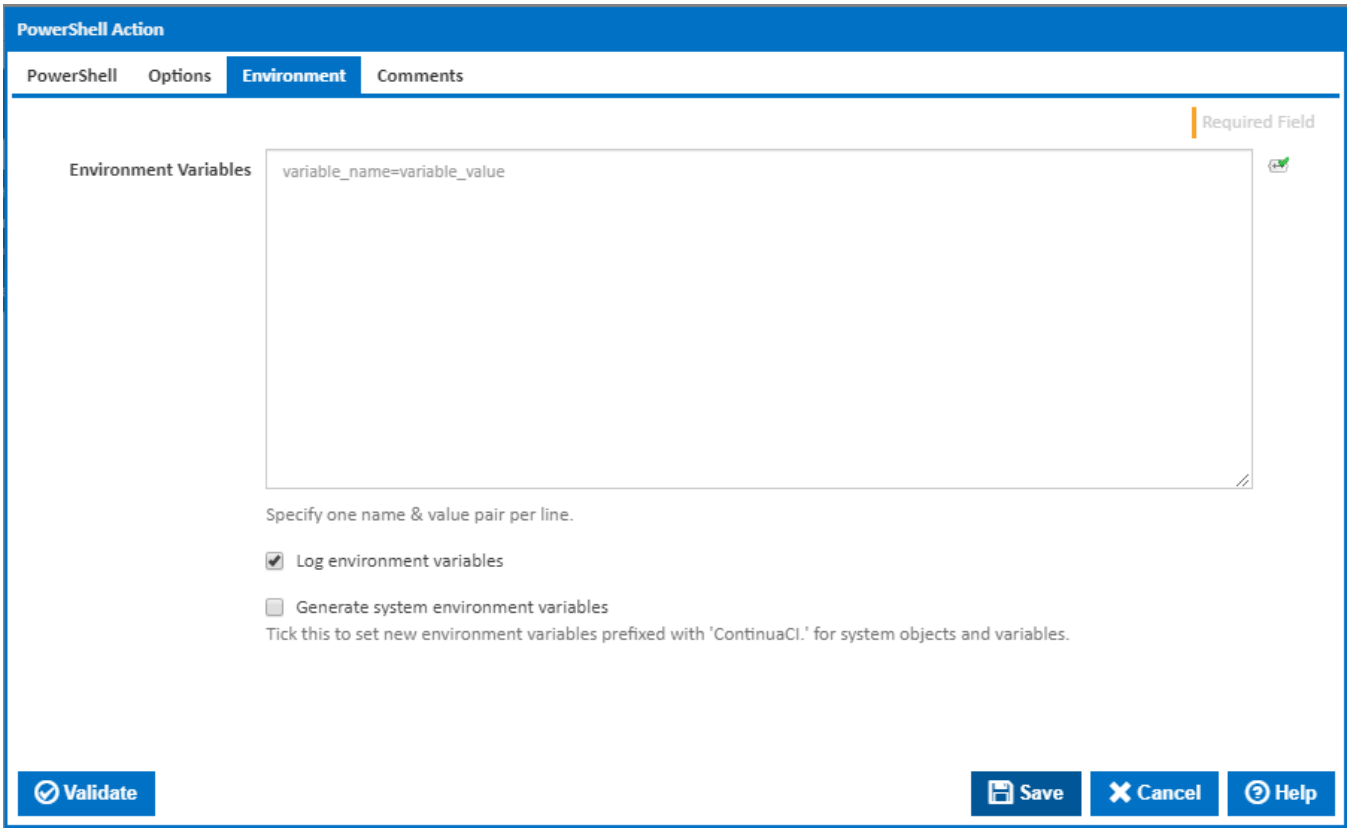
Treat failure as warning

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.