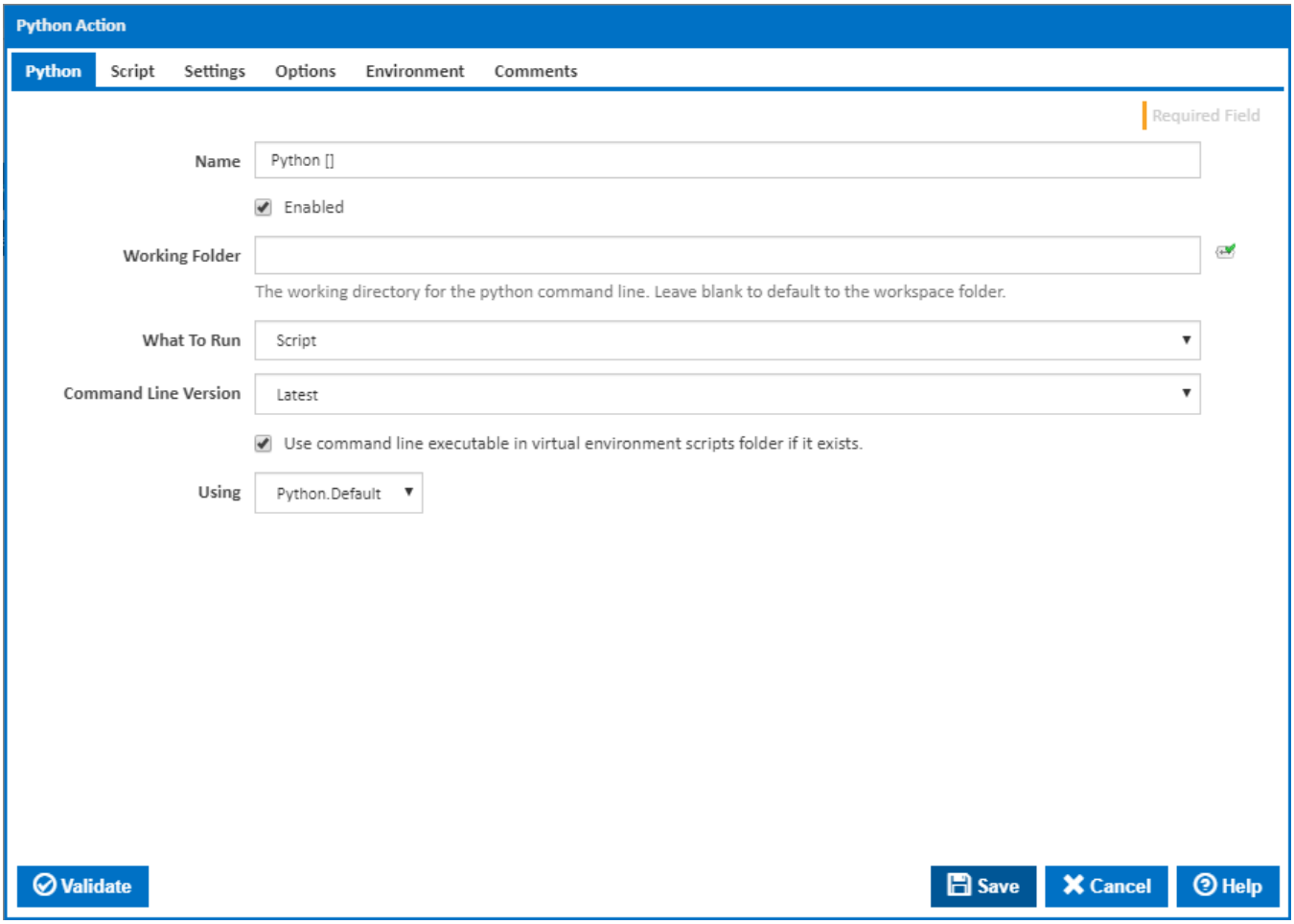
Python Action

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

Python action is used to run python scripts, modules or commands.

Python



Working Folder

The working directory for the python command line. Leave blank to default to the workspace folder.

What To Run

Select what to run. The relevant fields are located in their respective tabs.

- Script
- Module
- Command

Command Line Version

Select the version of the Python command line that is installed on the agent. Some other settings and options may be unavailable depending which the command line version is selected.

Use command line executable in virtual environment scripts folder if it exists.

If this is ticked and exists, the python executable in the virtual environment scripts folder will be used instead.

Using

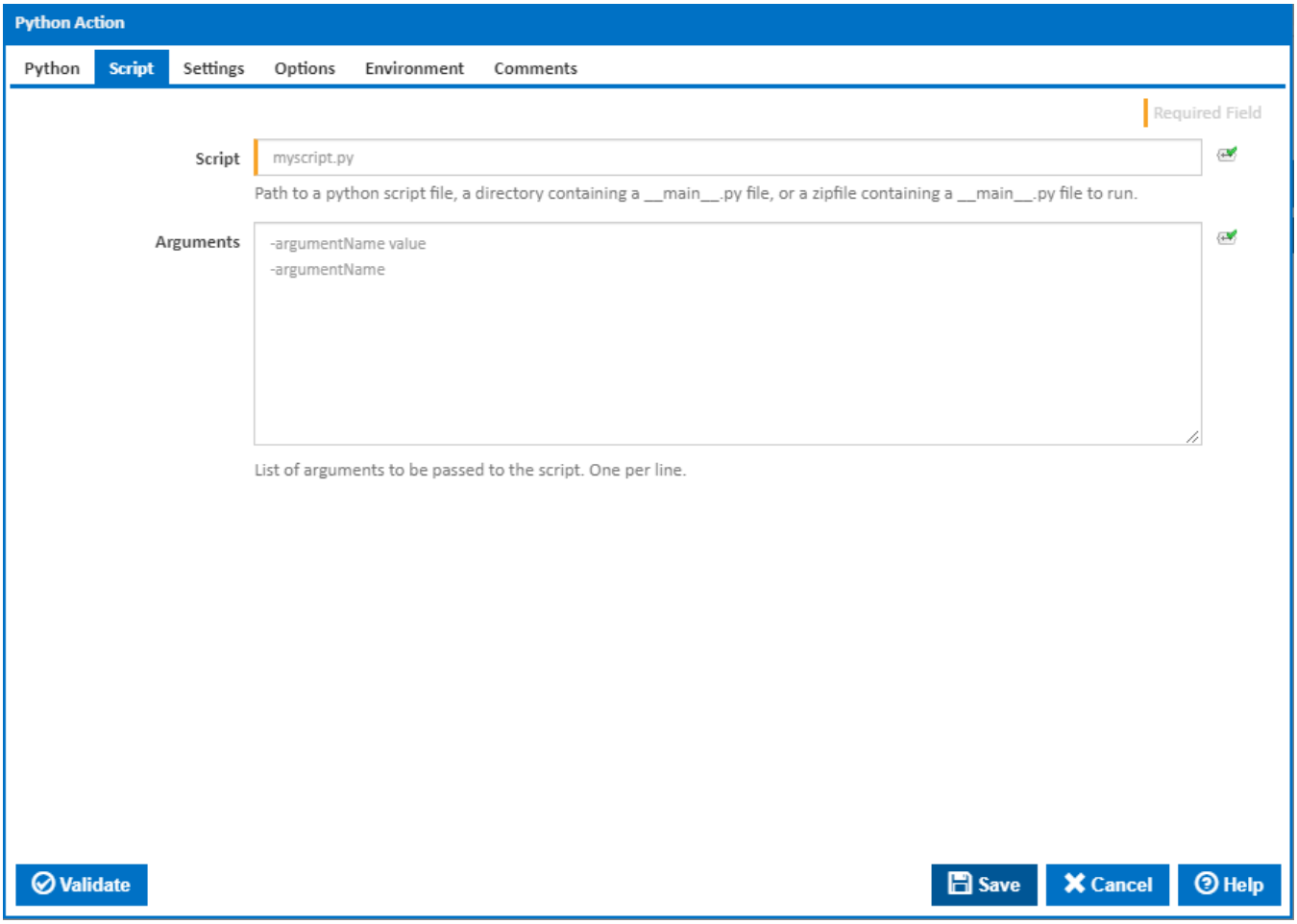
The Using drop down is populated by any property collector properties whose namespace matches the pattern defined by the Python action. The pattern for this action is `^Python\..*`. The default property collector searches the environment path for "python.exe".

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 "**Python**".

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.

Script



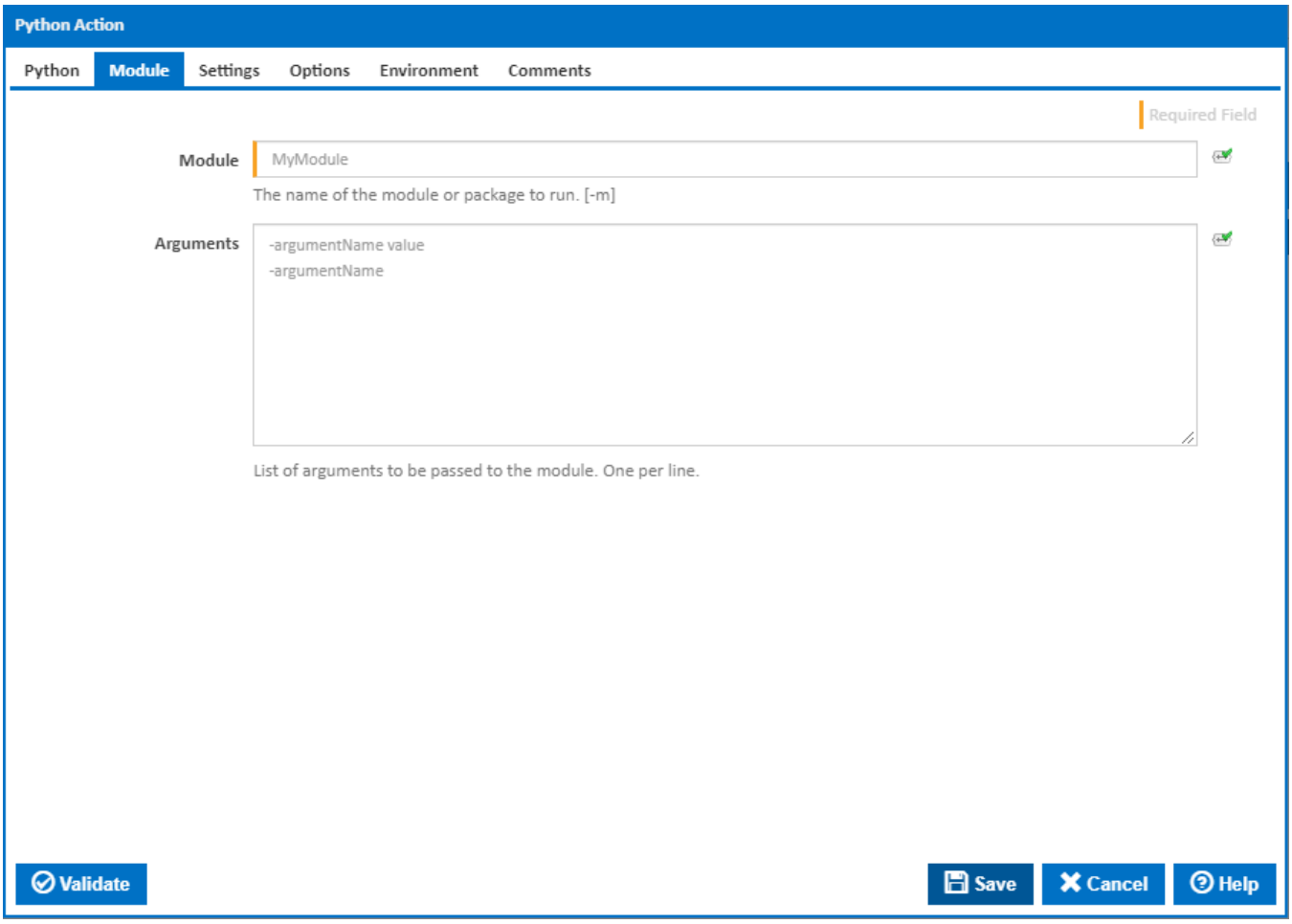
Script

Path to a python script file, a directory containing a __main__.py file, or a zipfile containing a __main__.py file to run.

Arguments

List of arguments to be passed to the script. One per line.

Module



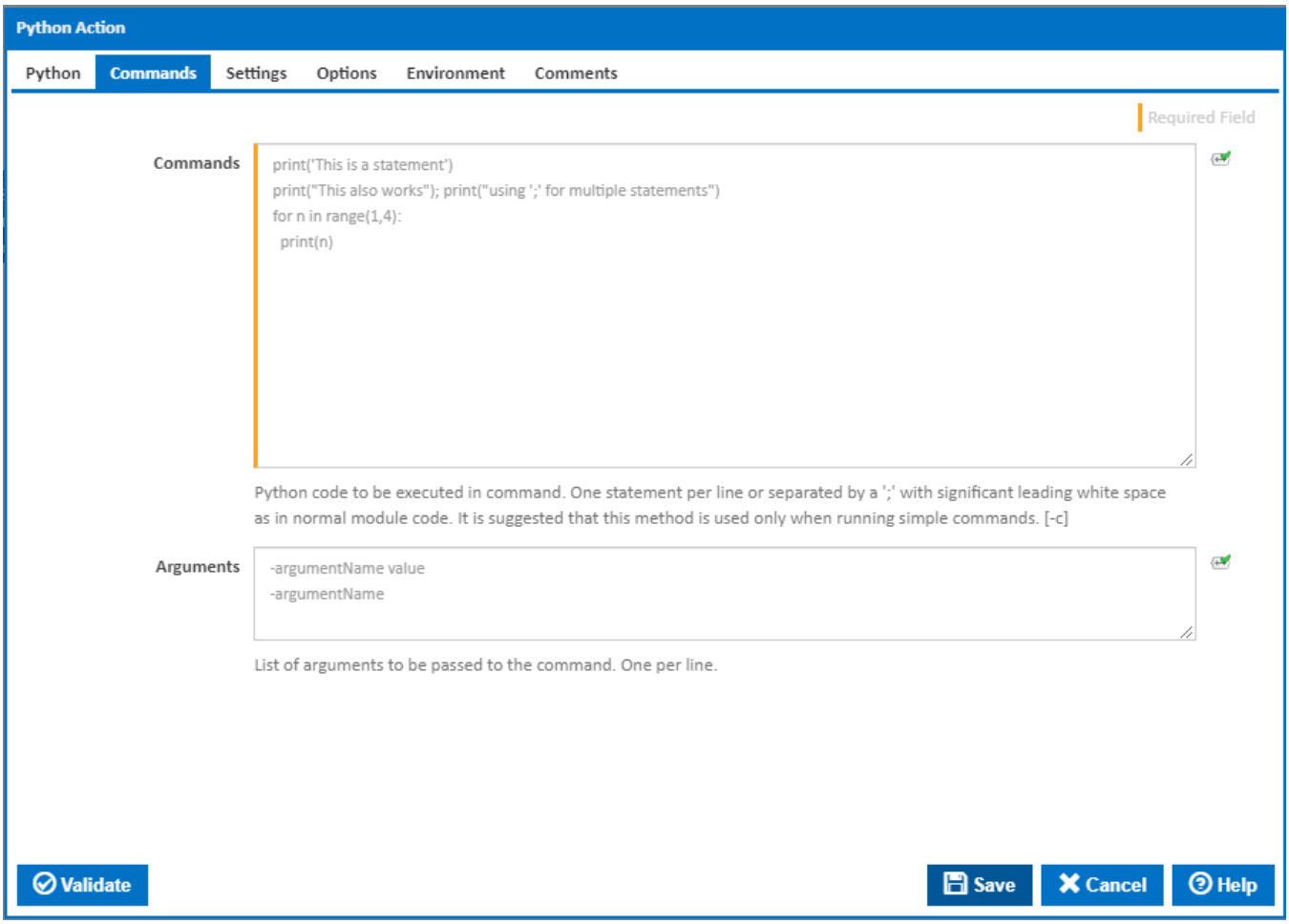
Module

The name of the module or package to run. [-m]

Arguments

List of arguments to be passed to the module. One per line.

Commands



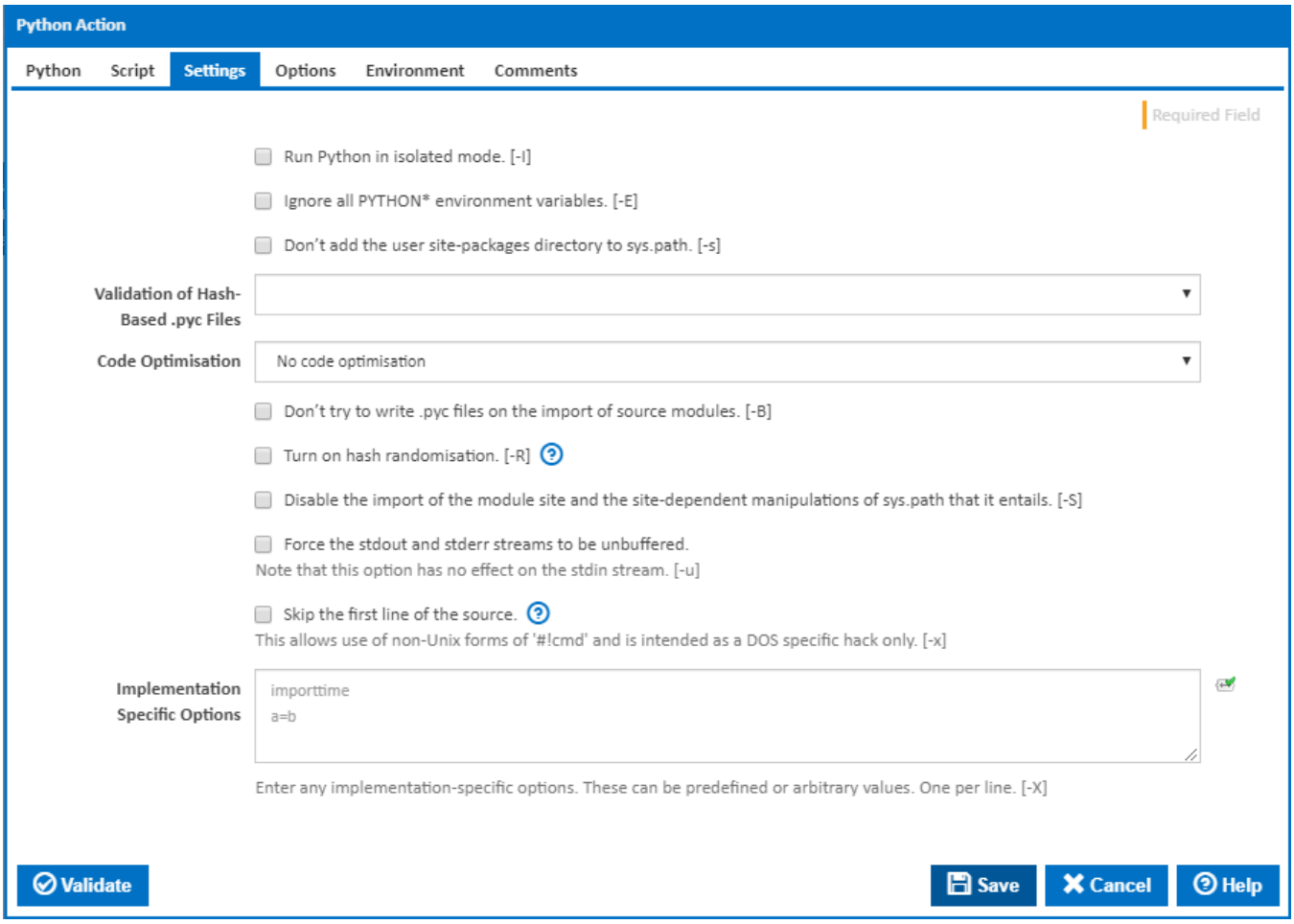
Commands

Python code to be executed in command. One statement per line or separated by a ';' with significant leading white space as in normal module code. It is suggested that this method is used only when running simple commands. [-c]

Arguments

List of arguments to be passed to the command. One per line.

Settings



Run Python in isolated mode.

If this is ticked, Python is run in isolated mode. [-I]

Ignore all PYTHON* environment variables.

Visible only if the checkbox '**Run Python in isolated mode**' is NOT ticked.

If this is ticked, all PYTHON* environment variables are ignored. This is implied if python is run in isolated mode. [-E]

Don't add the user site-packages directory to sys.path.

Visible only if the checkbox '**Run Python in isolated mode**' is NOT ticked.

If this is ticked, the user site-packages directory is not added to the sys.path. This is implied if python is run in isolated mode. [-s]

Division Control Semantics

Control the semantics of division. [-Q]

- **old** - division of int/int and long/long return an int or long (default)
- **new** - new division semantics, i.e. division of int/int and long/long returns a float
- **warn** - old division semantics with a warning for int/int and long/long
- **warnall** - old division semantics with a warning for all uses of the division operator

Note that this option is only available in Python version 2.x.

Validation of Hash-Based .pyc Files

Control the validation behavior of hash-based .pyc files. [----check-hash-based-pycs]

- **Default** - checked and unchecked hash-based bytecode cache files will be validated according to their default semantics.
- **Always** - all hash-based .pyc files, whether checked or unchecked, will be validated against their corresponding source file.
- **Never** - hash-based .pyc files will be validated against their corresponding source files.

Code Optimisation

Select the level of code optimisation.

- No code optimisation
- Remove assert statements and any code conditional on the value of __debug__ [-O]
- Remove assert statements, any code conditional on the value of __debug__ and docstrings [-OO]

Don't try to write .pyc files on the import of source modules.

If this is ticked, Python won't try to write .pyc files on the import of source modules. [-B]

Turn on hash randomisation.

If this is ticked, hash randomisation is turned on. [-R]

Note that this option only has an effect if the PYTHONHASHSEED environment variable is set to 0, since hash randomisation is enabled by default.

Disable the import of the module site and the site-dependent manipulations of the sys.path that it entails.

If this is ticked, the import of the module site and the site-dependent manipulations of sys.path that it entails is disabled. Also, the manipulations are disabled if site is explicitly imported later. [-S]

Force the stdout and stderr streams to be unbuffered.

If this is ticked, the stdout and stderr streams are forced to be unbuffered. [-u]

Skip the first line of the source.

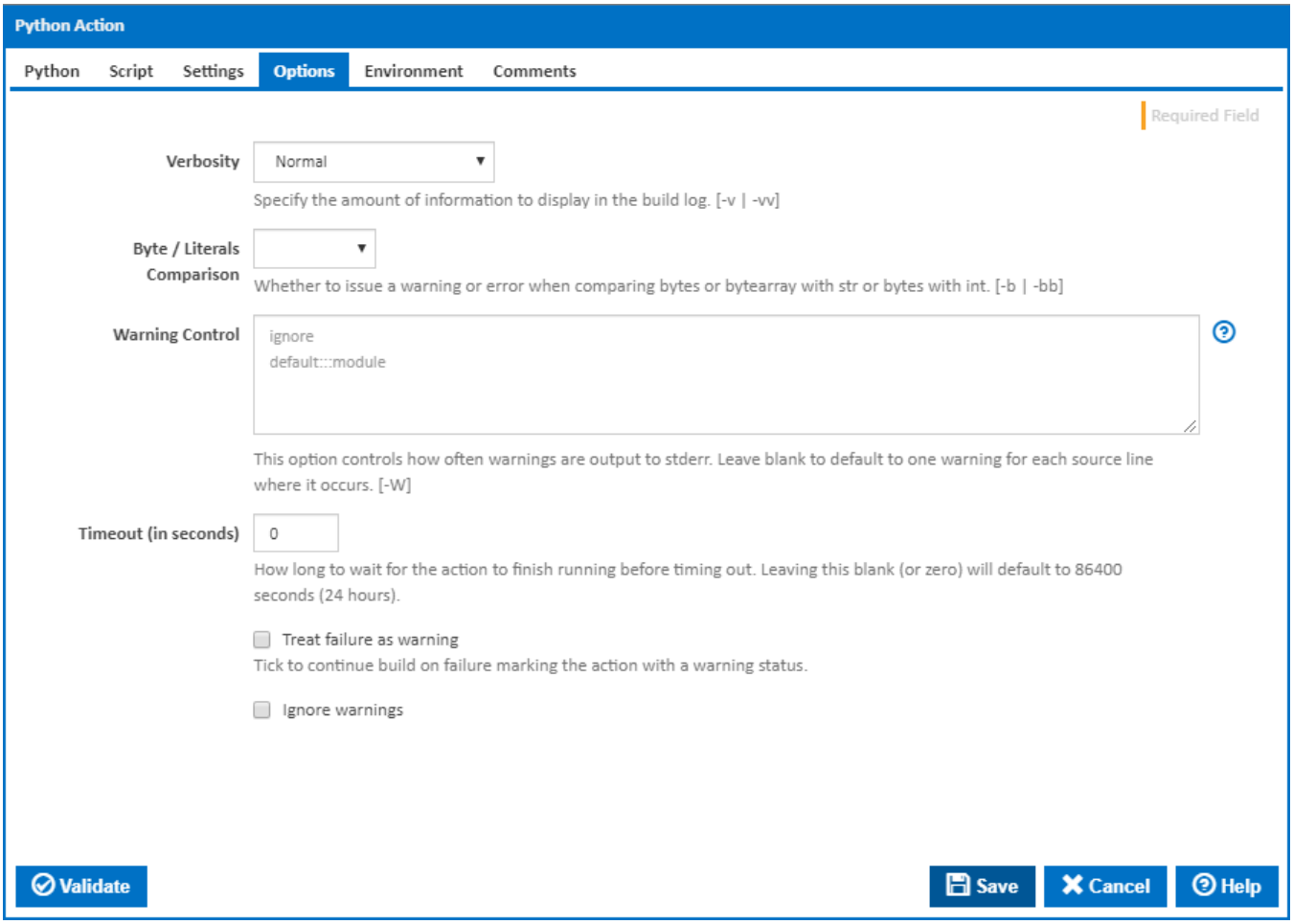
If this is ticked, the first line of the source is skipped. This allows the use of non-Unix forms of #!cmd. This is intended for a DOS specific hack only. [-x]

Note that, with this option ticked, line numbers in error messages will be off by one.

Implementation Specific Options

Enter any implementation-specific options. These can be predefined or arbitrary values. One per line. [-X]

Options



Verbosity

The amount of information detail to display in the build log. [-v | -vv]

Byte / Literals Comparison

Whether to issue a warning or error when comparing bytes or bytearray with str or bytes with int. [-b | -bb]

Indentation Issues

Select whether to issue a warning or error when a source file mixes tabs and spaces for indentation in a way that makes it depend on the worth of a tab expressed in spaces. [-t | -tt]

Note that this option is only available in Python version 2.x

Warning Control

This option controls how often warnings are output to stderr. Leave blank to default to one warning for each source line where it occurs. [-W]

Multiple options may be given; when a warning matches more than one option, the action for the last matching option is performed. Invalid options are ignored (though, a warning message is printed about invalid options when the first warning is issued).

The simplest form of argument is one of the following action strings:

- **ignore** - Ignore all warnings
- **default** - Explicitly request the default behaviour (printing each warning once per source line)
- **all** - Print a warning each time it occurs (this may generate many messages if a warning is triggered repeatedly for the same source line, such as inside a loop)
- **module** - Print each warning only the first time it occurs in each module
- **once** - Print each warning only the first time it occurs in the program.
- **error** - Raise an exception instead of printing a warning message.

The full form of argument is: `action:message:category:module:line`

For more information, please refer to the warnings module in the python documentation.

Warn about Python 3.x possible incompatibilities

If this is ticked, a 'DeprecationWarning' is emitted for features that are removed or significantly changed in Python 3 and can't be detected using static code analysis. [-3]

Note that this option is only available in Python versions 2.6 and 2.7.

Timeout (in seconds)

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

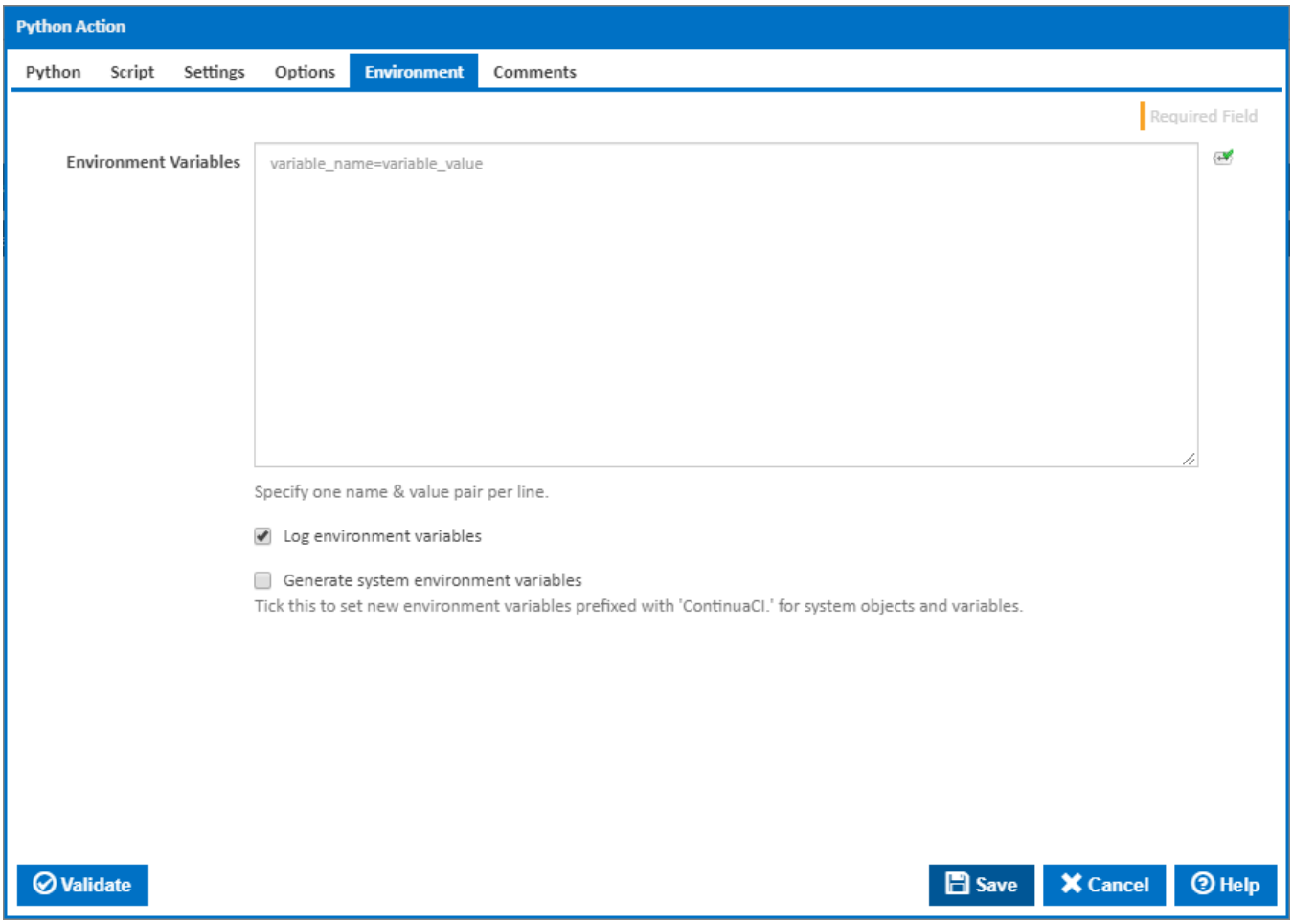
Treat failure as warning

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. [-W]

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.