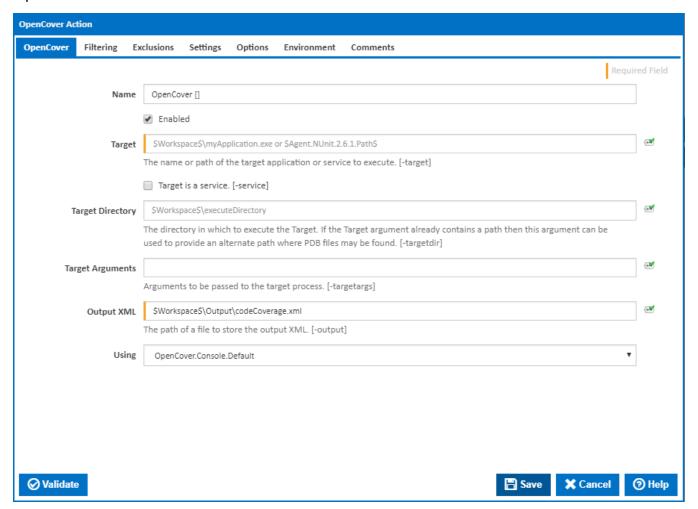# OpenCover Action

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

OpenCover is an open source code coverage utility built for .NET 2 and higher. It allows you to run code coverage tests over your .NET processes and specify filters regarding which assemblies, files or classes should be covered. Currently OpenCover only supports XML report output. Reports can be generated within Continua CI with either the 'Generate ReportGenerator Report' OpenCover option, or using the ReportGenerator action.

OpenCover

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.

Target

The name of the target application or service that will be started; this can also be a path to the target application. [-target:]

This can either be used to point to your application or service, or to your testing software executable, such as NUnit or MSTest.

When executing your testing software you can use expressions and refer to your software through continua. For example, you can call your NUnit executable with '$Agent.NUnit.2.6.1.Path$'.

Target is a service

If this is ticked, the value provided in the '**Target'** field is the name of a service rather than a name of a process. [-service]

Target Directory

The directory in which to execute the target application. If the directory is provided above, then this can be an alternate path where PDB files may be found. [-targetdir:]

Target Arguments

Arguments to be passed to the target process. [-targetargs:]

Output XML

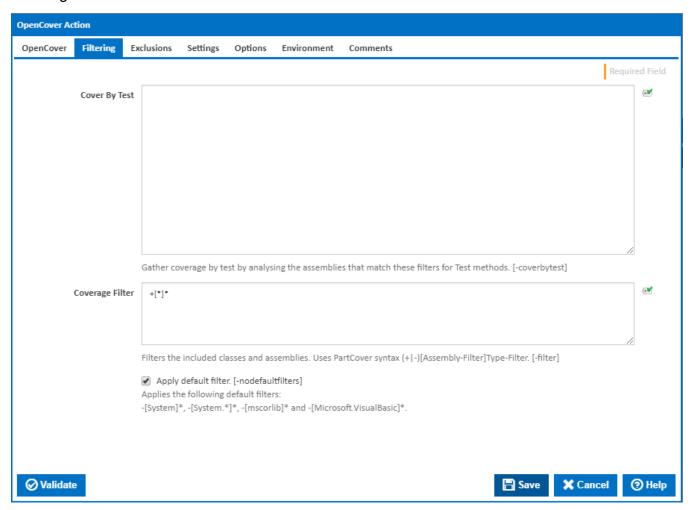The location and name of the output XML file. [-output]

Using

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

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 "**OpenCover**".

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.

Filtering

Cover By Test

Gather coverage by test by analysing the assemblies that match these filters for Test methods. [-coverbytest:]

Currently OpenCover only supports MSTest and NUnit tests for Cover by Test.

Coverage Filter

A list of filters to apply to selectively include or exclude assemblies and classes from coverage results. [-filter:]

Coverage Filter uses the following PartCover syntax: **±[module-filter]class-filter**. (*) can be used as a wildcard character. An exclusion filter (-) takes precedence over an inclusion filter (+).

Each filter should appear on a new line.

If no Coverage Filter is provided then the **+[*]*** (include all assemblies and classes) filter is used.
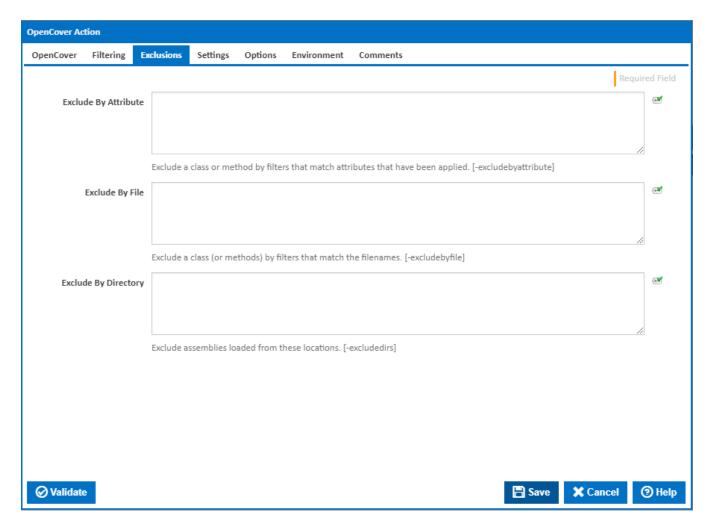
Apply default filter

Applies the following default OpenCover filters:

-[mscorlib]*

-[mscorlib.*]*

-[System]*

-[System.*]*

-[Microsoft.VisualBasic]*

These filters are applied by default and should be applied for most applications. Untick to pass the flag -nodefaultfilters.

Exclusions

OpenCover   Filtering   **Exclusions**   Settings   Options   Environment   Comments

Required Field

Exclude By Attribute

Exclude a class or method by filters that match attributes that have been applied. [-excludebyattribute]

Exclude By File

Exclude a class (or methods) by filters that match the filenames. [-excludebyfile]

Exclude By Directory

Exclude assemblies loaded from these locations. [-excludedirs]

⊘ Validate   💾 Save   ✖ Cancel   ⦸ Help

Exclude by Attribute

Exclude a class or method by filter(s) that match attributes that have been applied. An * can be used as a wildcard. [-excludebyattribute:]

Note that this field **does not use the PartCover Syntax**.

Each exclusion should appear on a new line.

Exclude by File

Exclude a class (or methods) by filter(s) that match the provided file names. An * can be used as a wildcard. [-excludebyfile:]

Note that this field **does not use the PartCover Syntax**.
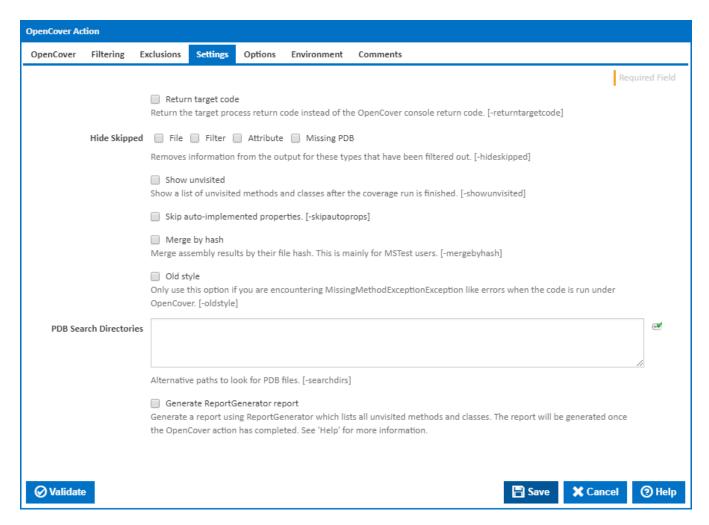
Each exclusion should appear on a new line.

Exclude by Directory

Exclude assemblies loaded from these locations. An * can be used as a wildcard. [-excludedirs:]

Note that this field **does not use the PartCover Syntax**.

Each exclusion should appear on a new line.

Settings

Return target code

Return the target process return code instead of the OpenCover console return code. [-returntargetcode]

Return Code Offset

Offset the OpenCover return code to a value outside the range returned by the target process.

Hide Skipped

Removes information from the Output XML that relates to classes/modules that have been skipped (filtered) through the Coverage Filter, Exclude By Attribute or Exclude By File fields. [-hideskipped]

Multiple values can be selected.

Show unvisited

Show a list of unvisited methods and classes after the coverage run is finished. [-showunvisited]

Skip auto-implemented properties

Tick to skip auto-implemented properties. [-skipautoprops]

Merge by hash

Under some scenarios e.g. using MSTest, an assembly may be loaded many times from different locations. This option is used to merge the coverage results for an assembly regardless of where it was loaded assuming the assembly has the same file-hash in each location. [-mergebyhash]

Old style

OpenCover will use old style instrumentation.

This instrumentation is not Silverlight friendly and is provided to support environments where NSCorLib instrumentation is not working. [-oldstyle]

PDB Search Directories
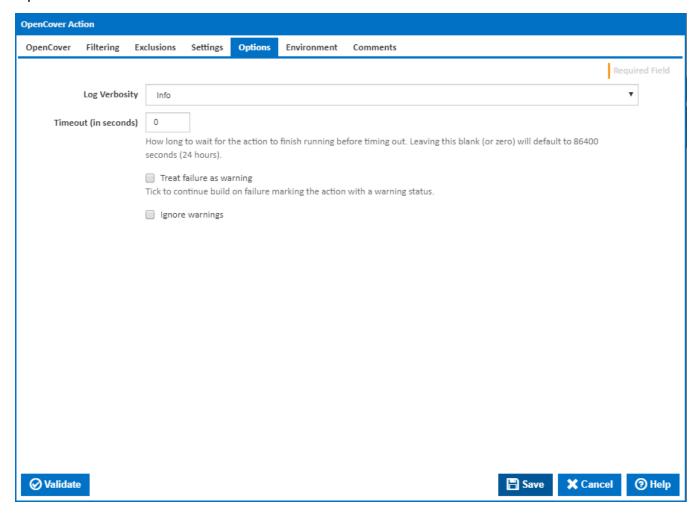
Alternative paths to look for PDB files. [-seearchdirs]

Generate ReportGenerator report

This generates a Code Coverage report using ReportGenerator. To generate this report, the ReportGenerator.exe must be accessible through the 'ReportGenerator.Default' property collector.

This will generate a HTML report to your 'Output XML' directory in a sub-folder called 'CoverageReport'. open the 'index.htm' file to view your report

This option is designed to get you a Code Coverage report quickly and easily. If the report needs to be further configured then uncheck this option and create a new ReportGenerator action after the OpenCover action and have it point to your OpenCover Output XML.

Options

Log Verbosity

The amount of information detail to display in the build log. [-log:]

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).
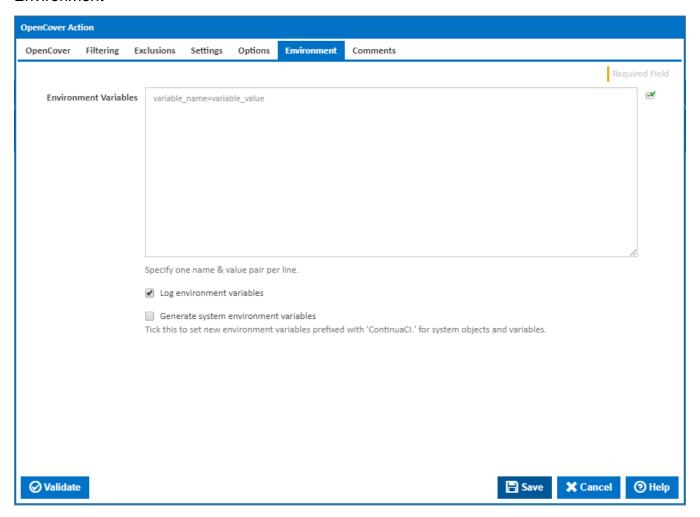
Treat failure as warning

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.