.Configuring+the+Test+Scope+v10.4.0

In this section:

Overview

The test scope refers to the file or set of files for testing. Use the `-resource` switch followed by a path in the solution to define the scope. Do not use file system paths to define the scope. Use the Visual Studio Solution Explorer path instead.

If you are running analysis from your IDE, a source file that is open in the active editor has higher priority than resources defined with Solution Explorer and only this file will be analyzed.

Testing a Single Project in a Solution

```
dottestcli.exe -solution "C:\Devel\FooSolution\FooSolution.sln"
-resource "FooSolution/QuxProject"
-config "builtin://Demo" -report "C:\Report"
```

Testing a Single Directory of Files in a Project

```
dottestcli.exe -solution "C:\Devel\FooSolution\FooSolution.sln"
-resource "FooSolution/BarProject/QuxDirectory"
-config "builtin://Demo"
```

Testing a Single Source File

```
dottestcli.exe -solution "C:\Devel\FooSolution\FooSolution.sln"
-resource "FooSolution/BarProject/QuxDirectory/BazFile.cs"
-config "builtin://Demo"
```

Testing a Single Project Under a Solution Folder

```
dottestcli.exe -solution "C:\Devel\FooSolution\FooSolution.sln"
-resource "FooSolution/BarSolutionFolder/QuxProject"
-config "builtin://Demo" -report "C:\Report"
```

Testing a Single Source File When No Solution is Provided

Because the name of the solution is unknown, the solution path should start with `/`:

```
dottestcli.exe -project "C:\Devel\FooSolution\FooProject.csproj"
-resource "/FooProject/BarDirectory/QuxFile.cs"
-config "builtin://Demo" -report "C:\Report"
```

Fine-tuning the Scope

Use the -include and -exclude switches to apply additional filters to the scope.

- `-include` instructs dotTEST to test only the files that match the file system path; all other files are skipped.
- `-exclude` instructs dotTEST to test all files except for those that match the file system path.

If both switches are specified, then all files that match `-include`, but not those that match `-exclude` patterns are tested.

These switches accept file system paths and ANT-style wildcards. This is in contrast to the `-resource` switch, which accepts the solution path and ANT-style wildcards.

The following example shows how to exclude all files under directories `*.Tests`:

```
dottestcli.exe -solution "C:\Devel\FooSolution\FooSolution.sln"
-exclude "C:\Devel\FooSolution\*.Tests\**\*.*"
-config "builtin://Demo" -report "C:\Report"
```

You can specify a file system path to a list file (*.lst) to include or exclude files in bulk. Each item in the *.lst file is treated as a separate entry.

Specifying Additional Assemblies

Use the `-reference` switch to specify a path to additional assemblies needed to resolve dependencies of the analyzed projects. ANT-style wildcards and relative paths to the current working directory are accepted.

Examples

```
-reference C:\MySolution\ExternalAssemblies\*.dll
-reference C:\MySolution\ExternalAssemblies\*.exe
-reference C:\MySolution\ExternalAssemblies\**\*.dll
-reference C:\MySolution\ExternalAssemblies\**\*.dll
```

Use the `-reference` switch if you receive an "Unable to find reference assembly" message.