

# Code coverage exclude: local OK, Azure DevOps => not OK #938

🔒 Closed **BloodSeaker** opened this issue on Aug 27, 2020 · 6 comments

Labels

waiting for customer

**BloodSeaker** commented on Aug 27, 2020 • edited ▼

Hi,

I am writing a .NET Core 3.1 web api and have a unit test project to run my unit test.  
Everything is being done in Visual Studio Professional 2019 (latest update).

In my unit test project (xUnit) I have following references to have coverlet working

```
<PackageReference Include="Microsoft.NET.Test.Sdk" Version="16.6.1" />
  <PackageReference Include="coverlet.msbuild" Version="2.9.0">
    <PrivateAssets>all</PrivateAssets>
    <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
  </PackageReference>
  <PackageReference Include="Microsoft.TestPlatform.ObjectModel" Version="16.6.1" />
```

When I run locally, my excludes do work (I want to exclude everything with Models in the namespace)  
dotnet test /p:CollectCoverage=true /p:CoverletOutputFormat=cobertura /p:CoverletOutput=../MyCoverage/ /p:Exclude="[\*]\*.Models\*"

.Business.Cache.Keys.E  
API.Business.Cache.Keys.E  
API.Business.Cache.Keys.E  
API.Business.Cache.Keys.M  
API.Business.Cache.Keys.O  
API.Business.Cache.Keys.O  
API.Business.Cache.Keys.S  
API.Business.Helpers.Bounc  
API.Business.Helpers.Emplc  
API.Business.Helpers.Medic  
API.Business.Helpers.Quest  
API.Business.Helpers.Surve  
API.Business.HttpClients.Ab  
API.Business.HttpClients.Ch  
API.Business.HttpClients.Cu  
API.Business.HttpClients.Cu  
API.Business.HttpClients.En  
API.Business.HttpClients.En  
API.Business.HttpClients.En  
API.Business.HttpClients.En  
API.Business.HttpClients.Ha  
API.Business.HttpClients.Ke  
API.Business.HttpClients.Me  
API.Business.HttpClients.Pa  
API.Business.Managers.Che  
API.Business.Managers.Cor  
API.Business.Managers.Col  
API.Business.Managers.EFc  
API.Business.Managers.Em  
API.Business.Managers.Em  
API.Business.Managers.Em  
API.Business.Managers.Exa  
API.Business.Managers.Exa  
API.Business.Managers.Key  
API.Business.Managers.Not

API.Business.Managers.Sur  
API.Business.Managers.Sur  
API.Business.Managers.Tra  
API.Business.Managers.Use

Now, when I include it in my build pipeline, with following step (and of course a step publishing it as well)

```
- task: DotNetCoreCLI@2
  displayName: Run unit tests
  inputs:
    command: test
    projects: '**/*Test/*.csproj'
    arguments: '/p:CollectCoverage=true /p:CoverletOutputFormat=cobertura /p:CoverletOutput=./MyCoverage/ /p:Exclude="[*]*.Models*"'
    publishTestResults: true
```

The coverage is calculated, report is available, but....

[API.Business.Managers.En](#)  
[API.Business.Managers.En](#)  
[API.Business.Managers.En](#)  
[API.Business.Managers.Ki](#)  
[API.Business.Managers.N](#)  
[API.Business.Managers.St](#)  
[API.Business.Managers.St](#)  
[API.Business.Managers.Tr](#)  
[API.Business.Managers.U](#)  
[API.Business.Models.Acc](#)  
[API.Business.Models.Acc](#)  
[API.Business.Models.Add](#)  
[API.Business.Models.Ansv](#)  
[API.Business.Models.Ansv](#)  
[API.Business.Models.Auth](#)  
[API.Business.Models.Con](#)  
[API.Business.Models.Cou](#)  
[API.Business.Models.Cust](#)  
[API.Business.Models.Doc](#)  
[API.Business.Models.Efoc](#)  
[API.Business.Models.Emp](#)

As you may see, it keeps the coverage on model in the report. Since the commands are the same, I am running out of ideas what the issue is here to be honest :) It's not like I can debug locally, as it works over there :)

UPDATE: Maybe good to know, the log in the build pipeline reports

Module	Line	Branch	Method
WNP.BFF.API.Business	74.03%	66.77%	71.24%
WNP.BFF.API.Data	91.66%	77.77%	88.88%
WNP.BFF.API	65.35%	58.1%	55.26%
WNP.BFF.API.Shared	25.86%	50%	26.31%

Whereas the report itself display for the API.Data project 66.6% line coverage & 70% for branch coverage

**MarcoRossignoli** commented on Aug 27, 2020

Can you try to remove ` from argument in yaml? Similar to our test task

<https://github.com/coverlet-coverage/coverlet/blob/master/eng/build.yml#L21>

 **MarcoRossignoli** added the **waiting for customer** label on Aug 27, 2020

**BloodSeaker** commented on Aug 27, 2020

Hi thanks for your quick reply

I just did another test and now I noticed something weird


Locally, I am using reportgenerator version 4 something.

Azure Devops is loading in a version 1.x :/ Stupid Azure DevOps.

I am manually adding ReportGenerator task now, that should do the trick



1

👍  **BloodSeaker** closed this as completed on Aug 27, 2020

**MarcoRossignoli** commented on Aug 27, 2020

Ok feel free to reopen if needed!

**ranouf** commented on Apr 7

Hi,

So following @**MarcoRossignoli** 's suggestion, I added in test task on Azure Dev Ops `/p:Exclude="[coverlet.core.tests.samples.netstandard]%"` `[coverlet.tests.xunit.extensions]"`

Please note that I use classic editor to configure the build pipeline and used yaml option to copy paste the configuration:

```
- task: DotNetCoreCLI@2
  displayName: 'dotnet test'
  inputs:
    command: test
    projects: '**/*Tests/*.csproj'
    arguments: '--configuration $(BuildConfiguration) --collect "XPlat Code coverage" -- RunConfiguration.DisableAppDomain=true --settings tests.runsettings /p:Exclude="[coverlet.core.tests.samples.netstandard]%" [coverlet.tests.xunit.extensions]"'
    testRunTitle: 'MyProject-Backend-Tests'
```

Here is the error:

```
One or more runsettings provided contain invalid token
##[error]Error: The process 'C:\hostedtoolcache\windows\dotnet\dotnet.exe' failed with exit code 1
##[warning]No test result files were found.
##[warning].NET 5 has some compatibility issues with older Nuget versions(<=5.7), so if you are using an older Nuget version(and not dotnet cli) to restore, then the dotnet cli commands (e.g. dotnet build) which rely on such restored packages might fail. To mitigate such error, you
```

can either: (1) - Use dotnet cli to restore, (2) - Use Nuget version 5.8 to restore, (3) - Use global.json using an older sdk version(<=3) to build

Info: Azure Pipelines hosted agents have been updated and now contain .Net 5.x SDK/Runtime along with the older .Net Core version which are currently lts. Unless you have locked down a SDK version for your project(s), 5.x SDK might be picked up which might have breaking behavior as compared to previous versions. You can learn more about the breaking changes here: <https://docs.microsoft.com/en-us/dotnet/core/tools/> and <https://docs.microsoft.com/en-us/dotnet/core/compatibility/> . To learn about more such changes and troubleshoot, refer here: <https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/build/dotnet-core-cli?view=azure-devops#troubleshooting>

```
##[error]Dotnet command failed with non-zero exit code on the following projects : D:\a\1\s\src\tests\MyProject.Tests\MyProject.Tests.csproj
Finishing: dotnet test
```

Thanks for your help

**MarcoRossignoli** commented on Apr 8 • edited ▼

**@ranouf** you should move `-- RunConfiguration.DisableAppDomain=true` after the main command. can you try:

```
arguments: '--configuration $(BuildConfiguration) --collect "XPlat Code coverage" --settings tests.runsettings /p:Exclude="[coverlet.core.tests.samples.netstandard]*%2c[coverlet.tests.xunit.extensions]*" -- RunConfiguration.DisableAppDomain=true'
```

**ranouf** commented on Apr 8

**@MarcoRossignoli** Thanks a lot, it works