Unit Test Access to Internals

Problem

Sometimes, you write a unit test that needs access to a method in the assembly/library under test.
But, it's visibility is set to private or protected, and not visible.

This is because, a unit test project works the same as a consuming library/process, where it only has access to the publicly visible, methods of the library under test.

> NOTE: We will talk in terms of making method calls internal, so that a unit test can reach them.
> But, this same technique can be applied to classes and properties, as well.

Solution

To get around this, while keeping things invisible, you can borrow a trick from C++.
In C++, you can make friend classes, so that there is a deeper visibility between them.
In C#, this can be approximated by assigning `Internal` visibility to a method.
That makes it public within the library/assembly, but private outside it.
And, we can add an attribute to the library/assembly, that allows internals to be visible to the specific unit test project.

So. The solution is in two-parts:

1. Assign Internal visibility to the method that your unit test needs to touch, directly.
2. Update the library, to allow your unit test to see internals.

The first step is trivial to do.
Just change the method visibility to internal, for what the unit test needs to reach.

> NOTE: This has a side effect of allowing anything in the library to see the method.

The second step can be done in one of two places:

In the Project File

>

> NOTE: Be sure to set the "MyProject.Tests" to the assembly name of the unit test project.

In the project file (.csproj), add an item group directly in modern SDK-style projects:

```
<ItemGroup>
  <InternalsVisibleTo Include="MyProject.Tests" />
</ItemGroup>
```

In the Assembly Definition

In the AssemblyInfo.cs, declare the attribute at the assembly level:

```
[assembly: System.Runtime.CompilerServices.InternalsVisibleTo("MyProject.Tests")]
```

Caveat

This is not a perfect fix, as you're still making a method publicly visible, within its library.
But, it is a way to prevent that public visibility from bleeding out to consuming libraries, while allowing the unit test to reach it.

Revision #1
Created 12 August 2026 17:42:49 by glwhite
Updated 12 August 2026 18:07:04 by glwhite