

## How to use Custom Attributes with {smartassembly}?

**PROFESSIONAL AND ENTERPRISE EDITIONS ONLY**

### Defining the Attributes

First, you need to define the Attributes. You can do it by either:

- Adding a reference to the *SmartAssembly.Attributes.dll* (located in the {smartassembly}\Attributes folder) in your project,
- Or adding the source code file to your project (*SmartAssembly.Attributes.cs* for a C# project, or *SmartAssembly.Attributes.vb* for a VB.NET project). Both files are located in the {smartassembly}\Attributes folder. If you choose this option, you can change the namespace to match your own and thus simplify the use of these Attributes.

### Using the Attributes

#### **DoNotObfuscateAttribute**

*Usage:* This attribute can be added to assembly, class, delegate, enum, field, interface, method, module and struct.

*Effect:* This will exclude the member from the obfuscation.

#### **DoNotObfuscateTypeAttribute**

*Usage:* This attribute can be added to any type (class, enum, interface and struct).

*Effect:* This will exclude the type definition as well as all type's members from the obfuscation.

#### **DoNotPruneAttribute**

*Usage:* This attribute can be added to assembly, class, constructor, delegate, enum, event, field, interface, method, module, parameter, property and struct.

*Effect:* This will exclude the member from the pruning.

#### **DoNotPruneTypeAttribute**

*Usage:* This attribute can be added to any type (class, enum, interface and struct).

*Effect:* This will exclude the type definition as well as all type's members from the pruning.

#### **DoNotSealTypeAttribute**

*Usage:* This attribute can be added to any type (class, enum, interface and struct).

*Effect:* This type will not be sealed, even if you set the option to automatically seal all possible classes.

#### **ReportExceptionAttribute**

*Usage:* This attribute can be added to any method.

*Effect:* Any unhandled exception which occurs in this method will be reported, even if the caller catches it.

#### **ObfuscateControlFlowAttribute**

*Usage:* This attribute can be added to any class, method and struct.

*Effect:* This will add stronger control flow obfuscation to the method, or to the methods of the type.

## C# Code Sample

```
[DoNotSealType()]
public class MyClass
{
    [DoNotPrune()]
    [DoNotObfuscate()]
    public void CalledOnlyByReflection()
    {
    }

    [DoNotPrune()]
    public string MyPropertyCalledByReflection
    {
        get
        {
            return "hello world";
        }
    }

    public void DoSomething([DoNotPrune()] string myString)
    {
    }

    [ReportException()]
    [ObfuscateControlFlow()]
    public void DoSomethingDangerous()
    {
    }

    public MyClass()
    {
    }

    [DoNotPrune()]
    public MyClass(string[] args)
    {
        //Called only by Reflection
    }
}

[DoNotPruneType()]
[DoNotObfuscateType()]
public class OtherClass
{
}
```

## VB.NET Code Sample

```
<DoNotSealType()> _
Public Class Class1

    <DoNotPrune(), DoNotObfuscate()> _
    Public Sub CalledOnlyByReflection()

        End Sub

    <DoNotPrune()> _
    Public ReadOnly Property MyPropertyCalledByReflection()
        Get
            Return "hello world"
        End Get
    End Property

    Public Sub DoSomething(<DoNotPrune()> ByVal MyString As String)

        End Sub

    <ReportException()> _
    <ObfuscateControlFlow()> _
    Public Sub DoSomethingDangerous()

        End Sub

    Public Sub New()

        End Sub

    <DoNotPrune()> _
    Public Sub New(ByVal args As String())
        REM Called only by Reflection
    End Sub
End Class

<DoNotPruneType(), DoNotObfuscateType()> _
Public Class OtherClass

End Class
```