

System.AttributeUsageAttribute Class

```
[ILASM]
.class public sealed serializable AttributeUsageAttribute
extends System.Attribute

[C#]
public sealed class AttributeUsageAttribute: Attribute
```

Assembly Info:

- Name: mscorlib
- Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- Version: 1.0.x.x
- Attributes:
 - CLSCompliantAttribute(true)

Type Attributes:

- AttributeUsageAttribute(AttributeTargets.Class, AllowMultiple=false, Inherited=true)

Summary

Specifies the behavior of a custom attribute when that attribute is defined.

Inherits From: System.Attribute

Library: BCL

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

[Note: Custom attributes may be applied to various application ("target") elements, such as classes, parameters, and structures (see **System.AttributeTargets** for the full list). The **System.AttributeUsageAttribute** class contains three properties that govern custom attribute behavior: the kinds of application elements the attribute may be associated with; whether the attribute may or may not be inherited by derived elements; and whether multiple instances of the attribute may or may not be allowed on the same target element.]

AttributeUsageAttribute(System.AttributeTargets) Constructor

```
[ILASM]
public rtspecialname specialname instance void
.ctor(valuetype System.AttributeTargets validOn)

[C#]
public AttributeUsageAttribute(AttributeTargets validOn)
```

Summary

Constructs and initializes a new instance of the **System.AttributeUsageAttribute** class.

Parameters

Parameter	Description
<i>validOn</i>	The set of application elements to which the attribute will be applied. When indicating multiple application elements, <i>validOn</i> is a bitwise OR combination of System.AttributeTargets enumeration values.

Description

The new instance will be constructed with the specified value of *validOn* and the properties **System.AttributeUsageAttribute.AllowMultiple** and **System.AttributeUsageAttribute.Inherited** set to their default values (**false** and **true** respectively).

AttributeUsageAttribute.AllowMultiple Property

```
[ILASM]
.property bool AllowMultiple { public hidebysig specialname
instance bool get_AllowMultiple() public hidebysig
specialname instance void set_AllowMultiple(bool value) }

[C#]
public bool AllowMultiple { get; set; }
```

Summary

Gets or sets a value indicating whether more than one instance of a specified attribute is permitted to be applied to any given program element.

Property Value

A **System.Boolean** where **true** indicates more than one instance of the attribute is permitted to be applied; otherwise, **false**. The default is **false**.

Description

[Note: It is expected that compilers will validate this property; this property is not validated during execution.]

Example

Example #1:

The following example demonstrates the use of **System.AttributeUsageAttribute.AllowMultiple**. If **AllowMultiple** for an attribute is set to **true**, more than one of those attributes may be assigned to any given program element.

```
[C#]

using System;

[AttributeUsageAttribute(AttributeTargets.Class |
                        AttributeTargets.Struct,
                        AllowMultiple = true)]
public class Author: Attribute {

    public Author(string name) { this.name = name; }
    public string name;
}
```

```

1
2     [Author("John Doe")]
3     [Author("John Q Public")]
4     class JohnsClass {
5
6         public static void Main() {}
7     }

```

8 **Example #2:**

9
10 The following example demonstrates an error that is expected to be
11 caught by compilers: the sample attempts to assign multiple instances
12 of an attribute for which **AllowMultiple** was set to **false**.

13
14 [C#]

```

15     using System;
16
17     [AttributeUsageAttribute(AttributeTargets.Class |
18                             AttributeTargets.Struct,
19                             AllowMultiple = false)]
20     public class Author: Attribute {
21
22         public Author(string name) { this.name = name; }
23         public string name;
24     }
25
26     [Author("John Doe")]
27     [Author("John Q Public")]
28     class JohnsClass {
29
30         public static void Main() {}
31     }

```

32 This should throw an error similar to:
33
34 error CS0579: Duplicate 'Author' attribute

35

AttributeUsageAttribute.Inherited Property

```
[ILASM]
.property bool Inherited { public hidebysig specialname
instance bool get_Inherited() public hidebysig specialname
instance void set_Inherited(bool value) }

[C#]
public bool Inherited { get; set; }
```

Summary

Gets or sets a **System.Boolean** value indicating whether the attribute can be inherited by subclasses of the class to which the attribute is applied.

Property Value

true indicates the attribute is inherited by subclasses; otherwise, **false**. The default is **true**.

Description

Information on an inherited attribute will be included in the metadata for the class on which it is applied, but will not be included in the metadata for classes that derive from it. A metadata consumer (such as reflection) is required therefore to traverse up the inheritance chain of a class if that consumer is interested in **System.Attribute** data that is marked inherited, but applied to an ancestor class. There is nothing for the compiler to validate at compile time.

1 AttributeUsageAttribute.ValidOn Property

```
2 [ILASM]  
3 .property valuetype System.AttributeTargets ValidOn {  
4 public hidebysig specialname instance valuetype  
5 System.AttributeTargets get_ValidOn() }  
  
6 [C#]  
7 public AttributeTargets ValidOn { get; }
```

8 Summary

9 Gets the set of values sent to the **System.AttributeUsageAttribute**
10 constructor that indicate to which targets the custom attribute may be
11 applied.

12 Property Value

14 One or more of the **System.AttributeTargets** values sent to the
15 constructor, combined by a bitwise OR operation.

16 Description

17 This property is read-only.
18