

# 1 System.AttributeUsageAttribute Class

2  
3

```
4 [ILASM]  
5 .class public sealed serializable AttributeUsageAttribute  
6 extends System.Attribute  
  
7 [C#]  
8 public sealed class AttributeUsageAttribute: Attribute
```

## 9 Assembly Info:

- 10 • Name: mscorlib
- 11 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 12 • Version: 1.0.x.x
- 13 • Attributes:
  - 14 ○ CLSCompliantAttribute(true)

## 15 Type Attributes:

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

## 18 Summary

19

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

## 22 Inherits From: System.Attribute

23

24 Library: BCL

25

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

28

## 29 Description

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

39

# 1 AttributeUsageAttribute(System.Attribute 2 Targets) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void  
5 .ctor(valuetype System.AttributeTargets validOn)  
6 [C#]  
7 public AttributeUsageAttribute(AttributeTargets validOn)
```

## 8 Summary

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

## 11 Parameters

12  
13

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 <b>System.AttributeTargets</b> enumeration values.

14

## 15 Description

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

21

# 1 AttributeUsageAttribute.AllowMultiple 2 Property

```
3 [ILASM]  
4 .property bool AllowMultiple { public hidebysig specialname  
5 instance bool get_AllowMultiple() public hidebysig  
6 specialname instance void set_AllowMultiple(bool value) }  
  
7 [C#]  
8 public bool AllowMultiple { get; set; }
```

## 9 Summary

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

## 13 Property Value

14  
15 A **System.Boolean** where **true** indicates more than one instance of  
16 the attribute is permitted to be applied; otherwise, **false**. The default  
17 is **false**.

## 18 Description

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

## 21 Example

### 23 Example #1:

24  
25 The following example demonstrates the use of  
26 **System.AttributeUsageAttribute.AllowMultiple**. If **AllowMultiple**  
27 for an attribute is set to **true**, more than one of those attributes may  
28 be assigned to any given program element.

```
29 [C#]  
30  
31 using System;  
32  
33 [AttributeUsageAttribute(AttributeTargets.Class |  
34 AttributeTargets.Struct,  
35 AllowMultiple = true)]  
36 public class Author: Attribute {  
37  
38     public Author(string name) { this.name = name; }  
39     public string name;  
40 }
```

```
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

# 1 AttributeUsageAttribute.Inherited 2 Property

```
3 [ILASM]  
4 .property bool Inherited { public hidebysig specialname  
5 instance bool get_Inherited() public hidebysig specialname  
6 instance void set_Inherited(bool value) }  
  
7 [C#]  
8 public bool Inherited { get; set; }
```

## 9 Summary

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

## 13 Property Value

14

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

## 17 Description

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

25

# 1 AttributeUsageAttribute.ValidOn Property

```
2 [ILASM]
3 .property valuetype System.AttributeTargets ValidOn {
4 public hidebysig specialname instance valuetype
5 System.AttributeTargets get_ValidOn() }
6
7 [C#]
8 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

13

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