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 00]
- *Version:* 2.0.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 can 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 can be associated with; whether the attribute can or cannot be inherited by derived elements; and whether multiple instances of the attribute can or cannot 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
validOn	The set of application elements to which the attribute will be applied. When indicating multiple application elements, *validOn* 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 can 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;
}

[Author( "John Doe" )]
```

```
[Author( "John Q Public" )]
class JohnsClass {

  public static void Main() {}
}
```

Example #2:

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

[C#]

```
using System;

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

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

[Author( "John Doe" )]
[Author( "John Q Public" )]
class JohnsClass {

  public static void Main() {}
}
```

This should throw an error similar to:

```
error CS0579: Duplicate 'Author' attribute
```

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.

AttributeUsageAttribute.ValidOn Property

[ILAsm]
```
.property valuetype System.AttributeTargets ValidOn { public hidebysig
specialname instance valuetype System.AttributeTargets get_ValidOn() }
```

[C#]
```
public AttributeTargets ValidOn { get; }
```

Summary

Gets the set of values sent to the `System.AttributeUsageAttribute` constructor that indicate to which targets the custom attribute can be applied.

Property Value

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

Description

This property is read-only.