

# System.FlagsAttribute Class

```
[ILAsm]
.class public serializable FlagsAttribute extends System.Attribute

[C#]
public class FlagsAttribute: 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.Enum, AllowMultiple=false, Inherited=false)

## Summary

Indicates that the `System.Enum` targeted by the current attribute is declared as a bit-field.

## 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

The `System.FlagsAttribute` class provides the consumer of a `System.Enum` the information that the enumeration is to be used as a bit-field. Additionally, when formatting a `System.Enum`, using the `System.FlagsAttribute` causes a value that is a bitwise OR combination of multiple fields to print correctly.

[*Note:* Bit-fields are generally used for lists of elements that might occur in combination; whereas enumeration constants are generally used for lists of mutually exclusive elements. Therefore, bit-fields are designed to be combined with the bitwise OR operator to generate unnamed values, whereas enumerated constants are not. Languages vary in their usage of bit-fields compared to enumeration constants.

This attribute can only be applied to enumerations.

]

## 1 Example

2 The following example demonstrates the use of `System.FlagsAttribute` on the  
3 formatting of a `System.Enum`. With this attribute, the *Position* enumeration is used as a  
4 bit-field, and the value 3 (Top | Left) is considered a valid value for the enumeration  
5 when formatting. Without this attribute, the enumeration *Color* is not used as a bit-field,  
6 and the value 3 (Red | Blue) is not considered a valid value for the enumeration when  
7 formatting.

8  
9 [C#]

```
10
11 using System;
12 [FlagsAttribute()]
13 public enum Position {
14
15     Top = 0x1,
16     Left = 0x2,
17     Bottom = 0x4,
18     Right = 0x8
19 }
20
21 //enum Color declared without FlagsAttribute
22 public enum Color {
23
24     Red = 0x1,
25     Blue = 0x2,
26     Yellow = 0x4
27 }
28
29 public class enumFormat {
30
31     public static void Main() {
32
33         Position p = Position.Top | Position.Left;
34         Console.WriteLine("Position: {0}", p);
35         Color c = Color.Red | Color.Blue;
36         Console.WriteLine("Color: {0}", c);
37     }
38 }
39
```

40 The output is

41  
42 Position: Top, Left

43  
44  
45 Color: 3

46

47

## 1 **FlagsAttribute()** Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor()  
  
4 [C#]  
5 public FlagsAttribute()
```

## 6 **Summary**

7 Constructs a new instance of the `System.FlagsAttribute` class.

8