

# 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
2    ]
```

### 3 Example

4

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

```
11
12 [C#]
```

```
13
14 using System;
15 [FlagsAttribute()]
16 public enum Position {
17
18     Top = 0x1,
19     Left = 0x2,
20     Bottom = 0x4,
21     Right = 0x8
22 }
23
24 //enum Color declared without FlagsAttribute
25 public enum Color {
26
27     Red = 0x1,
28     Blue = 0x2,
29     Yellow = 0x4
30 }
31
32 public class enumFormat {
33
34     public static void Main() {
35
36         Position p = Position.Top | Position.Left;
37         Console.WriteLine("Position: {0}", p);
38         Color c = Color.Red | Color.Blue;
39         Console.WriteLine("Color: {0}", c);
40     }
41 }
42
```

43 The output is

44

```
45 Position: Top, Left
```

46

47

```
48 Color: 3
```

49

50

## 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