

# System.Reflection.TypeAttributes Enum

```
[ILAsm]  
.class public sealed serializable TypeAttributes extends System.Enum  
  
[C#]  
public enum TypeAttributes
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Type Attributes:

- FlagsAttribute

## Summary

Specifies attributes of a type.

**Inherits From:** System.Enum

**Library:** Reflection

This enumeration is used by the `System.Type` class.

## TypeAttributes.Abstract Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes Abstract = 0x80
```

```
[C#]  
Abstract = 0x80
```

### Summary

Specifies that the type is not implemented in the declaring type.

## TypeAttributes.AnsiClass Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes AnsiClass = 0x0  
  
[C#]  
AnsiClass = 0x0
```

### Summary

Specifies that LPSTR is interpreted as ANSI.

## TypeAttributes.AutoClass Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes AutoClass = 0x20000  
  
[C#]  
AutoClass = 0x20000
```

### Summary

Specifies that LPSTR is interpreted automatically.

## TypeAttributes.AutoLayout Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes AutoLayout = 0x0  
  
[C#]  
AutoLayout = 0x0
```

### Summary

Specifies that fields of the type are automatically laid out by the system.

## TypeAttributes.BeforeFieldInit Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes BeforeFieldInit = 0x100000  
  
[C#]  
BeforeFieldInit = 0x100000
```

### Summary

Specifies that calling static methods of the type does not force the system to initialize the type.

## TypeAttributes.Class Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes Class = 0x0  
  
[C#]  
Class = 0x0
```

### Summary

Specifies that the type is a class.

## TypeAttributes.ClassSemanticsMask Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes ClassSemanticsMask = Interface  
  
[C#]  
ClassSemanticsMask = Interface
```

### Summary

Specifies a bitmask used to determine whether a type is a class or interface.



## TypeAttributes.CustomFormatClass Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes CustomFormatClass = 0x30000  
  
[C#]  
CustomFormatClass = 0x30000
```

### Summary

Specifies that LPSTR is interpreted by some implementation-specific means, which includes the possibility of throwing a `System.NotSupportedException`.

# TypeAttributes.CustomStringFormatMask Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes CustomStringFormatMask = 0xC00000  
  
[C#]  
CustomStringFormatMask = 0xC00000
```

## Summary

.This mask is used to retrieve non-standard encoding information for System.Reflection.TypeAttributes.CustomFormatClass. The meaning of the values of these two bits is unspecified.

## TypeAttributes.ExplicitLayout Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes ExplicitLayout = 0x10  
  
[C#]  
ExplicitLayout = 0x10
```

### Summary

Specifies that the layout of fields in the type is provided explicitly.

## TypeAttributes.Interface Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes Interface = 0x20  
  
[C#]  
Interface = 0x20
```

### Summary

Specifies that the type is an interface.

## TypeAttributes.LayoutMask Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes LayoutMask = SequentialLayout |  
ExplicitLayout
```

```
[C#]  
LayoutMask = SequentialLayout | ExplicitLayout
```

### Summary

Specifies a bitmask used to obtain layout information.

## TypeAttributes.NestedAssembly Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes NestedAssembly = Public |  
NestedFamily
```

```
[C#]  
NestedAssembly = Public | NestedFamily
```

### Summary

Specifies that the type is nested and is accessible only to members within its assembly.

# TypeAttributes.NestedFamANDAssem Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes NestedFamANDAssem = NestedPublic |  
NestedFamily  
  
[C#]  
NestedFamANDAssem = NestedPublic | NestedFamily
```

## Summary

Specifies that the type is nested and is accessible only to members of its family in its assembly.

## TypeAttributes.NestedFamily Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes NestedFamily = 0x4  
  
[C#]  
NestedFamily = 0x4
```

### Summary

Specifies that the type is nested and is accessible only to members of its family.



## TypeAttributes.NestedFamORAssem Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes NestedFamORAssem = Public |  
NestedPublic | NestedFamily
```

```
[C#]  
NestedFamORAssem = Public | NestedPublic | NestedFamily
```

### Summary

Specifies that the type is nested and is accessible only to members of its family and throughout its assembly.

## TypeAttributes.NestedPrivate Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes NestedPrivate = Public |  
NestedPublic
```

```
[C#]  
NestedPrivate = Public | NestedPublic
```

### Summary

Specifies that the type is nested with private visibility.

## TypeAttributes.NestedPublic Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes NestedPublic = 0x2  
  
[C#]  
NestedPublic = 0x2
```

### Summary

Specifies that the type is nested with public visibility.

## TypeAttributes.NotPublic Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes NotPublic = 0x0
```

```
[C#]  
NotPublic = 0x0
```

### Summary

Specifies that the type is not public.

## TypeAttributes.Public Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes Public = 0x1  
  
[C#]  
Public = 0x1
```

### Summary

Specifies that the type has public visibility.

## TypeAttributes.Sealed Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes Sealed = 0x100  
  
[C#]  
Sealed = 0x100
```

### Summary

Specifies that the type cannot be used to derive new types.

## TypeAttributes.SequentialLayout Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes SequentialLayout = 0x8  
  
[C#]  
SequentialLayout = 0x8
```

### Summary

Specifies that fields in the type are laid out sequentially.

## TypeAttributes.SpecialName Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes SpecialName = 0x400  
  
[C#]  
SpecialName = 0x400
```

### Summary

Specifies that the type is treated in a special way by some tools.

[*Note:* For more information on special names, see Partition I of the CLI Specification.

For more information on `SpecialName` in metadata, see Partition II of the CLI Specification.

]



## TypeAttributes.StringFormatMask Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes StringFormatMask = UnicodeClass |  
AutoClass
```

```
[C#]  
StringFormatMask = UnicodeClass | AutoClass
```

### Summary

Specifies a bitmask used to obtain string format information.

## TypeAttributes.UnicodeClass Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes UnicodeClass = 0x10000  
  
[C#]  
UnicodeClass = 0x10000
```

### Summary

Specifies that LPSTR is interpreted as Unicode.

## TypeAttributes.VisibilityMask Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.TypeAttributes VisibilityMask = Public |  
NestedPublic | NestedFamily
```

```
[C#]  
VisibilityMask = Public | NestedPublic | NestedFamily
```

### Summary

Specifies a bitmask used to obtain visibility information.