

# System.Reflection.MethodAttributes Enum

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

## 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)

## Summary

Specifies flags for method attributes.

## Inherits From: System.Enum

**Library:** Reflection

## Description

This enumeration is used by the `System.Reflection.MethodBase.Attributes` property.

# MethodAttributes.Abstract Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
Abstract = 1024  
  
[C#]  
Abstract = 1024
```

## Summary

Specifies that the type that declares the method does not provide an implementation.

# MethodAttributes.Assembly Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
Assembly = 3  
  
[C#]  
Assembly = 3
```

## Summary

Specifies that the method is accessible throughout the assembly.

# MethodAttributes.FamANDAssem Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
FamANDAssem = 2  
  
[C#]  
FamANDAssem = 2
```

## Summary

Specifies that the method is accessible only to members of the type that defines the method and its subclasses in the assembly that contains the defining type.

# MethodAttributes.Family Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
Family = 4  
  
[C#]  
Family = 4
```

## Summary

Specifies that the method is accessible only to the members of the type that defines it and its subclasses.

# MethodAttributes.FamORAssem Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
FamORAssem = 5  
  
[C#]  
FamORAssem = 5
```

## Summary

Specifies that the method is accessible to members of the type that defines the method and its subclasses, and throughout the assembly.

# 1 MethodAttributes.Final Field

```
2 [ILAsm]  
3 .field public static literal valuetype System.Reflection.MethodAttributes  
4 Final = 32  
  
5 [C#]  
6 Final = 32
```

## 7 Summary

8 Specifies that the method cannot be overridden.

9

## MethodAttributes.HideBySig Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
HideBySig = 128  
  
[C#]  
HideBySig = 128
```

### Summary

Specifies that the method is hidden by name and signature.

[*Note:* A method that is not hidden by name and signature is hidden by name only.]



## MethodAttributes.MemberAccessMask Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
MemberAccessMask = 7  
  
[C#]  
MemberAccessMask = 7
```

### Summary

Specifies a bit-mask that, when combined with the attributes of a method using the logical AND operator, yields the bits that specify the accessibility of a method.

## MethodAttributes.NewSlot Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
NewSlot = 256  
  
[C#]  
NewSlot = 256
```

### Summary

Specifies that the method always gets a new slot in the v-table.

# MethodAttributes.PinvokeImpl Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
PinvokeImpl = 8192  
  
[C#]  
PinvokeImpl = 8192
```

## Summary

Specifies that the method implementation is forwarded through PInvoke (Platform Invocation Services).

# MethodAttributes.Private Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
Private = 1  
  
[C#]  
Private = 1
```

## Summary

Specifies that the method is accessible only to members in the defining type.

## MethodAttributes.PrivateScope Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
PrivateScope = 0  
  
[C#]  
PrivateScope = 0
```

### Summary

Specifies that the member cannot be referenced.

# 1 MethodAttributes.Public Field

```
2 [ILAsm]  
3 .field public static literal valuetype System.Reflection.MethodAttributes  
4 Public = 6  
  
5 [C#]  
6 Public = 6
```

## 7 Summary

8 Specifies that the method is accessible to members of any type.

9

## MethodAttributes.ReuseSlot Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
ReuseSlot = 0  
  
[C#]  
ReuseSlot = 0
```

### Summary

Specifies that the method will reuse an existing slot in the v-table.

[*Note:* This is the default behavior.]

# MethodAttributes.SpecialName Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
SpecialName = 2048  
  
[C#]  
SpecialName = 2048
```

## Summary

Specifies that the method 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.

]



# MethodAttributes.Static Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
Static = 16  
  
[C#]  
Static = 16
```

## Summary

Specifies that the method is invoked on the defining type.

[*Note:* For method invocation, the target of a static method is a `System.Type` object that represents the type where the method is defined.]

# MethodAttributes.Virtual Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
Virtual = 64  
  
[C#]  
Virtual = 64
```

## Summary

Specifies that the method is virtual.

## MethodAttributes.VtableLayoutMask Field

```
[ILAsm]  
.field public static literal valuetype System.Reflection.MethodAttributes  
VtableLayoutMask = 256  
  
[C#]  
VtableLayoutMask = 256
```

### Summary

Specifies a bit-mask that, when combined with the attributes of a method using the logical AND operator, yields the v-table attributes of a method.