

# System.Reflection.BindingFlags Enum

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

## 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 flags that control the binding and the invocation processes conducted by reflection.

## Inherits From: System.Enum

**Library:** Reflection

## Description

This enumeration is used by reflection classes such as `System.Reflection.Binder`, `System.Reflection.Module`, and `System.Reflection.ConstructorInfo`. `System.Reflection.BindingFlags` values are used to control binding in methods in classes that find and invoke, create, get, and set members and types.

To specify multiple `System.Reflection.BindingFlags` values, use the bitwise 'OR' operator.

## BindingFlags.CreateInstance Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags CreateInstance = 0x200  
  
[C#]  
CreateInstance = 0x200
```

### Summary

Specifies that an instance of a type will be created.

## BindingFlags.DeclaredOnly Field

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

### Summary

Specifies that only the members declared on a type will be searched when binding. When this field is specified, inherited members will not be searched.

## BindingFlags.ExactBinding Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags ExactBinding = 0x10000
```

```
[C#]  
ExactBinding = 0x10000
```

### Summary

Specifies that, when binding to a method, the types of the arguments passed to the binder are required to be the same as the types of the corresponding parameters in the method signature.

[*Note:* Custom binders are allowed to ignore this flag. This flag is provided as a hint for custom binders.]

## BindingFlags.GetField Field

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

### Summary

Specifies that the value of a field will be returned.

## BindingFlags.GetProperty Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags GetProperty = 0x1000
```

```
[C#]  
GetProperty = 0x1000
```

### Summary

Specifies that the value of a property will be returned by invoking the `get` accessor of a `System.Reflection.PropertyInfo` instance that reflects that property.

## BindingFlags.IgnoreCase Field

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

### Summary

Specifies that a member name will be handled in a case-insensitive manner when binding.

## BindingFlags.Instance Field

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

### Summary

Specifies that instance members will be included in the binding search.



## BindingFlags.InvokeMethod Field

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

### Summary

Specifies that a method will be invoked.

## BindingFlags.NonPublic Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags NonPublic = 0x20
```

```
[C#]  
NonPublic = 0x20
```

### Summary

Specifies that non-public (i.e. family and private) members will be included in the binding search if the search has the permission required to bind non-public members.

## BindingFlags.OptionalParamBinding Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags OptionalParamBinding = 0x40000  
  
[C#]  
OptionalParamBinding = 0x40000
```

### Summary

Specifies that the set of members whose parameter count matches the number of arguments supplied to the binding method are to be returned by the binding method. [*Note:* This flag is used in conjunction with methods that have parameters with default values and with methods that have variable arguments.]

## BindingFlags.Public Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags Public = 0x10
```

```
[C#]  
Public = 0x10
```

### Summary

Specifies that public members will be included in the binding search.

## BindingFlags.SetField Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags SetField = 0x800
```

```
[C#]  
SetField = 0x800
```

### Summary

Specifies that the value of a field will be set.

## BindingFlags.SetProperty Field

```
[ILAsm]  
.field public static literal valuetype  
System.Reflection.BindingFlags SetProperty = 0x2000
```

```
[C#]  
SetProperty = 0x2000
```

### Summary

Specifies that the value of a property will be set by invoking the `set` accessor of a `System.Reflection.PropertyInfo` instance that reflects that property.

## BindingFlags.Static Field

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

### Summary

Specifies that static members will be included in the binding search.

## BindingFlags.SuppressChangeType Field

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

### Summary

Specifies that the system will not change the types of parameters when binding with the default binder.

[*Note:* Custom binders are allowed to ignore this flag. This flag is provided as a hint for custom binders.]