

1 System.Reflection.MemberInfo Class

```
2 [ILAsm]  
3 .class public abstract serializable MemberInfo extends System.Object  
4 [C#]  
5 public abstract class MemberInfo
```

6 Assembly Info:

- 7 • *Name:* mscorlib
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

13 Provides access to member metadata.

14 Inherits From: System.Object

15

16 **Library:** Reflection

17

18 **Thread Safety:** This type is safe for multithreaded operations.

19

20 Description

21 [*Note:* System.Reflection.MemberInfo is used to represent all members of a type:
22 nested types, fields, events, properties, methods, and constructors. The Base Class
23 Library includes the following derived types:

- 24 • System.Reflection.FieldInfo
- 25 • System.Reflection.EventInfo
- 26 • System.Reflection.PropertyInfo
- 27 • System.Type

28]

29

1 MemberInfo() Constructor

```
2 [ILAsm]  
3 family rtspecialname specialname instance void .ctor()  
4 [C#]  
5 protected MemberInfo()
```

6 Summary

7 Constructs a new instance of the `System.Reflection.MemberInfo` class.

8

MemberInfo.DeclaringType Property

```
[ILAsm]
.property class System.Type DeclaringType { public hidebysig virtual
abstract specialname class System.Type get_DeclaringType() }

[C#]
public abstract Type DeclaringType { get; }
```

Summary

Gets the type that declares the member reflected by the current instance.

Property Value

The `System.Type` object of the class that declares the member reflected by the current instance; or, `null` if the member reflected by the current instance is a global member.

Description

[Note: A member of a class (or interface) is either declared on that type or inherited from a base class (or interface). The `System.Reflection.MemberInfo.DeclaringType` property value cannot be the same as the `System.Type` object used to obtain the current instance. These values will differ if either of the following conditions is true.

- If the `System.Type` object from which the current instance was obtained did not declare the member reflected by the current instance, the `System.Reflection.MemberInfo.DeclaringType` will represent the base type that is closest to that object in its hierarchy chain and declares the member reflected by the current instance.
- If the current instance reflects a global member, (that is, it was obtained from `System.Reflection.Module.GetMethods`, which returns global methods on a module), then the `System.Reflection.MemberInfo.DeclaringType` property value is `null`.

]

Behaviors

This property is read-only.

This property is required to return the `System.Type` object for the type that declares the member reflected by the current instance. This property value is required to be equal to the `System.Reflection.MemberInfo.ReflectedType` property value of the current instance if and only if the reflected type also contains a declaration for the member reflected by the current instance.

How and When to Override

1 Override this property to get the `System.Type` of the class that declared the member
2 that is reflected by the current instance.

3

4 **Usage**

5 Use this property to determine the `System.Type` of the class that declared the member
6 that is reflected by the current instance.

7

8 **Example**

9 The following example demonstrates the difference between the
10 `System.Reflection.MemberInfo.DeclaringType` and
11 `System.Reflection.MemberInfo.ReflectedType` of a member.

12 [C#]

```
14 using System;
15 using System.Reflection;
16
17 public class BaseClass {
18     public void ReflectedMethod() {}
19 }
20
21 public class DerivedClass: BaseClass {}
22
23 public class DeclaringTypeExample {
24     public static void Main() {
25         Type t = typeof(DerivedClass);
26         MemberInfo [] memInfo = t.GetMember("ReflectedMethod");
27         Console.WriteLine("Reflected type is {0}.", memInfo[0].ReflectedType);
28         Console.WriteLine("Declaring type is {0}.", memInfo[0].DeclaringType);
29     }
30 }
31
```

32 The output is

33

34 Reflected type is DerivedClass.

35

36

37 Declaring type is BaseClass.

38

39

1 MemberInfo.Name Property

```
2 [ILAsm]  
3 .property string Name { public hidebysig virtual abstract specialname  
4 string get_Name() }  
  
5 [C#]  
6 public abstract string Name { get; }
```

7 Summary

8 Gets the name of the member reflected by the current instance.

9 Property Value

10 A `System.String` containing the name of the member reflected by the current instance.

11 Behaviors

12 This property is read-only.

13

14 Only the simple name, not the fully qualified name, of the member reflected by the
15 current instance is returned.

16

17 [*Note:* For example, if the current instance reflects the member `Print` in
18 `System.MyClass`, the `System.Reflection.MemberInfo.Name` property would be "Print".]

19

20

21

1 MemberInfo.ReflectedType Property

```
2 [ILAsm]  
3 .property class System.Type ReflectedType { public hidebysig virtual  
4 abstract specialname class System.Type get_ReflectedType() }  
5 [C#]  
6 public abstract Type ReflectedType { get; }
```

7 Summary

8 Gets the type of the class through which the current instance was obtained.

9 Property Value

10 The `System.Type` object for the class through which the current instance was obtained.

11 Behaviors

12 This property is read-only.

13
14 `ReflectedType` is required to get the type of the object that was used to obtain the
15 current instance. This property value is required to be equal to the
16 `System.Reflection.MemberInfo.DeclaringType` property value of the current instance
17 if and only if the reflected type also contains a declaration for the member reflected by
18 the current instance.

19