

# 1 System.Reflection.Module Class

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

## 6 Assembly Info:

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

## 12 Summary

13 Provides access to module metadata.

## 14 Inherits From: System.Object

15

16 **Library:** Reflection

17

18 **Thread Safety:** All public static members of this type are safe for multithreaded operations.  
19 No instance members are guaranteed to be thread safe.

20

## 21 Description

22 A module is a single portable executable (PE) file.

23

24 [*Note:* One or more modules deployed as a unit composes an assembly.

25

26 For more information on modules, see Partition II of the CLI Specification.

27

28 ]

29

# 1 Module.GetField(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Reflection.FieldInfo  
4 GetField(string name)  
  
5 [C#]  
6 public FieldInfo GetField(string name)
```

## 7 Summary

8 Returns a `System.Reflection.FieldInfo` instance that reflects the global public field  
9 that has the specified name, and is a member of the module reflected by the current  
10 instance.

## 11 Parameters

Parameter	Description
<i>name</i>	A <code>System.String</code> that specifies the name of the field to be returned.

## 12 13 Return Value

14 Returns a `System.Reflection.FieldInfo` instance that reflects the global public field  
15 that has the name *name*, and is a member of the module reflected by the current  
16 instance, if found; otherwise, returns `null`.

## 17 Description

18 [Note: A global public field is a static field with a module-level scope.]  
19  
20

## 21 Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>name</i> is <code>null</code> .

22

23

# 1 Module.GetField(System.String, 2 System.Reflection.BindingFlags) Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Reflection.FieldInfo  
5 GetField(string name, valuetype System.Reflection.BindingFlags  
6 bindingAttr)  
  
7 [C#]  
8 public FieldInfo GetField(string name, BindingFlags bindingAttr)
```

## 9 Summary

10 Returns a `System.Reflection.FieldInfo` instance that reflects the global field that has  
11 the specified name and has the specified binding attributes, and is a member of the  
12 module reflected by the current instance.

## 13 Parameters

Parameter	Description
<i>name</i>	A <code>System.String</code> that specifies the name of the field to be returned.
<i>bindingAttr</i>	A bitwise combination of <code>System.Reflection.BindingFlags</code> value that control the binding process. [Note: Specify <code>System.Reflection.BindingFlags.Public</code> or <code>System.Reflection.BindingFlags.NonPublic</code> , or <code>System.Reflection.BindingFlags.Static</code> ; otherwise, this method will return <code>null</code> .]

14

## 15 Return Value

16 Returns a `System.Reflection.FieldInfo` instance that reflects the global field that has  
17 the name *name* and characteristics specified by *bindingAttr*, and is a member of the  
18 module reflected by the current instance, if found; otherwise, returns `null`.

## 19 Description

20 [Note: A global field is a field with a module-level scope.]  
21  
22

## 23 Exceptions

Exception	Condition
System.ArgumentNullException	<i>name</i> is null.

1

2

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23

## Module.GetFields(System.Reflection.BindingFlags) Method

```
[ILAsm]  
.method public hidebysig instance class System.Reflection.FieldInfo[]  
GetFields(valuetype System.Reflection.BindingFlags bindingAttr)  
  
[C#]  
public FieldInfo[] GetFields(BindingFlags bindingAttr)
```

### Summary

Returns an array whose elements reflect the global fields that have the specified binding attributes, and are members of the module reflected by the current instance.

### Parameters

Parameter	Description
<i>bindingAttr</i>	A bitwise combination of System.Reflection.BindingFlags values that control the binding process. [Note: Specify System.Reflection.BindingFlags.Public or System.Reflection.BindingFlags.NonPublic, and System.Reflection.BindingFlags.Static; otherwise, this method will return null.]

### Return Value

An array System.Reflection.FieldInfo objects that reflect the global fields that have the specified binding attributes, and are members of the module reflected by the current instance. If no global fields are contained in the module reflected by the current instance, returns an array with zero elements.

### Description

[Note: A global public field is a static field with a module-level scope.]

# 1 Module.GetFields() Method

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Reflection.FieldInfo[]  
4 GetFields()  
  
5 [C#]  
6 public FieldInfo[] GetFields()
```

## 7 Summary

8 Returns an array whose elements reflect the global public fields that are members of the  
9 module reflected by the current instance.

## 10 Return Value

11 An array of `System.Reflection.FieldInfo` objects that reflect the global public fields  
12 that are members of the module reflected by the current instance. If no global public  
13 fields are contained in the module reflected by the current instance, returns an array  
14 with zero elements.

## 15 Description

16 [*Note:* A global public field is a static field with a module-level scope.]  
17  
18  
19

# 1 Module.GetMethod(System.String, 2 System.Type[]) Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Reflection.MethodInfo  
5 GetMethod(string name, class System.Type[] types)  
  
6 [C#]  
7 public MethodInfo GetMethod(string name, Type[] types)
```

## 8 Summary

9 Returns a `System.Reflection.MethodInfo` instance that reflects the global public  
10 method that has the specified name and parameter types, and is a member of the  
11 module reflected by the current instance.

## 12 Parameters

Parameter	Description
<i>name</i>	A <code>System.String</code> that specifies the name of the method to be returned.
<i>types</i>	An array of <code>System.Type</code> objects that contain the parameter types to search for.

## 13 14 Return Value

15 A `System.Reflection.MethodInfo` instance that reflects the global public method that  
16 has the name *name* and parameter types *types*, and is a member of the module  
17 reflected by the current instance, if found; otherwise, returns `null`.

## 18 Description

19 [Note: A global public method is a static method with a module-level scope.]  
20  
21

## 22 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>name</i> is null. -or- <i>types</i> is null.

	-or- At least one element of <i>types</i> is null.
--	---

1

2

# 1 Module.GetMethod(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Reflection.MethodInfo  
4 GetMethod(string name)  
5 [C#]  
6 public MethodInfo GetMethod(string name)
```

## 7 Summary

8 Returns a `System.Reflection.MethodInfo` instance that reflects the global public  
9 method that has the specified name, and is a member of the module reflected by the  
10 current instance.

## 11 Parameters

Parameter	Description
<i>name</i>	A <code>System.String</code> that specifies the name of the method to be returned.

## 12 13 Return Value

14 A `System.Reflection.MethodInfo` instance that reflects the global public method that  
15 has the name *name*, and is a member of the module reflected by the current instance, if  
16 found; otherwise, returns `null`.

## 17 Description

18 [Note: A global public method is a static method with a module-level scope.]  
19  
20

## 21 Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>name</i> is null.

22

23

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

# Module.GetMethods(System.Reflection.BindingFlags) Method

```
[ILAsm]  
.method public hidebysig instance class System.Reflection.MethodInfo[]  
GetMethods(valuetype System.Reflection.BindingFlags bindingAttr)  
  
[C#]  
public MethodInfo[] GetMethods(BindingFlags bindingAttr)
```

## Summary

Returns an array whose elements reflect the global methods that have the specified binding attributes, and are members of the module reflected by the current instance.

## Parameters

Parameter	Description
<i>bindingAttr</i>	A bitwise combination of System.Reflection.BindingFlags values that control the binding process. [Note: Specify System.Reflection.BindingFlags.Public or System.Reflection.BindingFlags.NonPublic, and System.Reflection.BindingFlags.Static; otherwise, this method will return null.]

## Return Value

An array of System.Reflection.MethodInfo objects that reflect the global methods that have characteristics specified by bindingAttr, and are members of the module reflected by the current instance, if found; otherwise, returns null.

## Description

[Note: A global method is a method with a module-level scope.]

# 1 Module.GetMethods() Method

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Reflection.MethodInfo[]  
4 GetMethods()  
  
5 [C#]  
6 public MethodInfo[] GetMethods()
```

## 7 Summary

8 Returns an array whose elements reflect the global public methods that are members of  
9 the module reflected by the current instance.

## 10 Return Value

11 An array of `System.Reflection.MethodInfo` objects that reflect the global public  
12 methods that are members of the module reflected by the current instance.

## 13 Description

14 [*Note:* A global public method is a static method with a module-level scope.]  
15  
16  
17

# 1 Module.ToString() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual string ToString()  
4 [C#]  
5 public override string ToString()
```

## 6 Summary

7 Returns a string representation of the name of the module reflected by the current  
8 instance.

## 9 Return Value

10 A `System.String` representation of the name of the module reflected by the current  
11 instance.

## 12 Description

13 [*Note:* This method overrides `System.Object.ToString.`]  
14  
15

16

1 **The following member must be implemented if the RuntimeInfrastructure library is**  
2 **present in the implementation.**

### 3 **Module.Assembly Property**

```
4 [ILAsm]  
5 .property class System.Reflection.Assembly Assembly { public hidebysig  
6 specialname instance class System.Reflection.Assembly get_Assembly() }  
7 [C#]  
8 public Assembly Assembly { get; }
```

#### 9 **Summary**

10 Gets the appropriate assembly for the module reflected by the current instance.

#### 11 **Property Value**

12 A System.Reflection.Assembly instance that reflects the assembly that contains the  
13 module reflected by the current instance.

#### 14 **Description**

15 This property is read-only.

16

# 1 Module.FullyQualifiedName Property

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

## 7 Summary

8 Gets a string that represents the full path of the module reflected by the current  
9 instance.

## 10 Property Value

11 A `System.String` that represents the full path of the module reflected by the current  
12 instance. If the assembly that contains the module reflected by the current instance was  
13 loaded from a `System.Byte` array, the value of this string is "<Unknown>".

## 14 Description

15 This property is read-only.

## 16 Behaviors

17 The case-sensitivity of the module name is implementation-specific.

18

## 19 How and When to Override

20 Override this property to customize the content of the `System.String` returned by this  
21 property in types derived from `System.Reflection.Module`.

22

## 23 Usage

24 To obtain the name of the module without path information, use the  
25 `System.Reflection.Module.Name` property.

26

## 27 Permissions

Permission	Description
------------	-------------

<b>System.Security.Permissions. FileIOPermission</b>	Requires permission to access path information. See <code>System.Security.Permissions.FileIOPermission</code> and <code>System.Security.Permissions.FileIOPermissionAccess.PathDiscovery</code> .
--	---

- 1
- 2

# 1 Module.Name Property

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

## 7 Summary

8 Gets a string containing the name of the module reflected by the current instance, with  
9 the path component removed.

## 10 Property Value

11 A `System.String` containing the name of the module reflected by the current instance,  
12 with the path component removed. If the assembly that contains the module reflected  
13 by the current instance was loaded from a `System.Byte` array, the value of this string is  
14 "<Unknown>".

## 15 Description

16 This property is read-only.

17  
18 [*Note:* The value of this property is equivalent to the value of the string returned by the  
19 `System.Reflection.Module.ToString` method.

20  
21 Use `System.Reflection.Module.FullyQualifiedName` to get the name and path of the  
22 module reflected by the current instance.

23  
24 ]

25