

# System.Reflection.Module Class

```
[ILASM]
.class public serializable Module extends System.Object

[C#]
public class Module
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Summary

Provides access to module metadata.

## Inherits From: System.Object

**Library:** Reflection

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

## Description

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

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

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

# 1 Module.GetField(System.String) Method

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

## 7 Summary

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

## 11 Parameters

12  
13

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

14  
15  
16

## 15 Return Value

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

## 21 Description

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

## 23 Exceptions

24  
25

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

26  
27  
28

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

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

## 10 Summary

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

## 15 Parameters

16  
17

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

18

## 19 Return Value

20

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

## 25 Description

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

## 27 Exceptions

28

29

1  
2  
3

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

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

```
3 [ILASM]
4 .method public hidebysig instance class
5 System.Reflection.FieldInfo[] GetFields(valuetype
6 System.Reflection.BindingFlags bindingAttr)
7
8 [C#]
9 public FieldInfo[] GetFields(BindingFlags bindingAttr)
```

## 9 Summary

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

## 13 Parameters

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

## 16 Return Value

17 An array **System.Reflection.FieldInfo** objects that reflect the global  
18 fields that have the specified binding attributes, and are members of  
19 the module reflected by the current instance. If no global fields are  
20 contained in the module reflected by the current instance, returns an  
21 array with zero elements.  
22  
23

## 24 Description

25 [Note: A global public field is a static field with a module-level scope.]  
26

# 1 Module.GetFields() Method

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

## 7 Summary

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

## 10 Return Value

11

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

## 16 Description

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

18

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

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

## 9 Summary

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

## 14 Parameters

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

## 17 Return Value

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

## 24 Description

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

## 27 Exceptions

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

1  
2  
3

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

# 1 Module.GetMethod(System.String)

## 2 Method

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

### 8 Summary

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

### 12 Parameters

13  
14

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

15  
16  
17

### 16 Return Value

18 A **System.Reflection.MethodInfo** instance that reflects the global  
19 public method that has the name *name*, and is a member of the  
20 module reflected by the current instance, if found; otherwise, returns  
21 **null**.

### 22 Description

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

### 25 Exceptions

26  
27

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

28  
29  
30

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

```
3 [ILASM]
4 .method public hidebysig instance class
5 System.Reflection.MethodInfo[] GetMethods(valuetype
6 System.Reflection.BindingFlags bindingAttr)
7
8 [C#]
9 public MethodInfo[] GetMethods(BindingFlags bindingAttr)
```

## 9 Summary

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

## 13 Parameters

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

## 17 Return Value

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

## 23 Description

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

25

# 1 Module.GetMethods() Method

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

## 7 Summary

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

## 10 Return Value

11

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

## 15 Description

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

18

# 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  
8 the current instance.

## 9 Return Value 10

11 A **System.String** representation of the name of the module reflected  
12 by the current instance.

## 13 Description

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

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 {  
6 public hidebysig specialname instance class  
7 System.Reflection.Assembly get_Assembly() }
```

```
8 [C#]  
9 public Assembly Assembly { get; }
```

#### 10 **Summary**

11 Gets the appropriate assembly for the module reflected by the current  
12 instance.

#### 13 **Property Value**

14

15 A **System.Reflection.Assembly** instance that reflects the assembly  
16 that contains the module reflected by the current instance.

#### 17 **Description**

18 This property is read-only.

19

# 1 Module.FullyQualifiedName Property

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

## 7 Summary

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

## 10 Property Value

11

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

## 16 Description

17 This property is read-only.

## 18 Behaviors

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

## 20 How and When to Override

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

## 24 Usage

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

## 27 Permissions

28

29

Permission	Description
<b>System.Security.Permissions.FileIOPermission</b>	Requires permission to access path information. See <b>System.Security.Permissions.FileIOPermission</b> and <b>System.Security.Permissions.FileIOPermissionAccess</b> .

1  
2  
3

	<b>PathDiscovery.</b>
--	-----------------------

# 1 Module.Name Property

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

## 7 Summary

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

## 10 Property Value

11

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

## 16 Description

17 This property is read-only.

18

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

21

22 Use **System.Reflection.Module.FullyQualifiedName** to get the  
23 name and path of the module reflected by the current instance.]

24