

# 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 04 00 00 00 00 00 00 00]
- *Version:* 2.0.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.

]

# Module.GetField(System.String) Method

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

## Summary

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

## Parameters

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

## Return Value

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

## Description

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

## Exceptions

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



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

```
[ILAsm]
.method public hidebysig instance class System.Reflection.FieldInfo
GetField(string name, valuetype System.Reflection.BindingFlags
bindingAttr)

[C#]
public FieldInfo GetField(string name, BindingFlags bindingAttr)
```

## Summary

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

## 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. [ <i>Note:</i> 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> .]

## Return Value

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

## Description

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

1   **Exceptions**

2

3

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

4

5

6

## 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 <code>System.Reflection.BindingFlags</code> values that control the binding process. [Note: Specify <code>System.Reflection.BindingFlags.Public</code> or <code>System.Reflection.BindingFlags.NonPublic</code> , and <code>System.Reflection.BindingFlags.Static</code> ; 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.]

# Module.GetFields() Method

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

## Summary

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

## Return Value

An array of `System.Reflection.FieldInfo` objects that reflect the global public fields that are members of the module reflected by the current instance. If no global public 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.]

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

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

## Summary

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

## 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.

## Return Value

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

## Description

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

## Exceptions

Exception	Condition
-----------	-----------



**System.ArgumentNullException**

*name* is null.

-or-

*types* is null.

-or-

At least one element of *types* is null.

1

2

3

# Module.GetMethod(System.String) Method

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

## Summary

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

## Parameters

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

## Return Value

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

## Description

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

## Exceptions

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



## 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 <code>System.Reflection.BindingFlags</code> values that control the binding process. [Note: Specify <code>System.Reflection.BindingFlags.Public</code> or <code>System.Reflection.BindingFlags.NonPublic</code> , and <code>System.Reflection.BindingFlags.Static</code> ; 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.]

# Module.GetMethods() Method

```
[ILAsm]
.method public hidebysig instance class System.Reflection.MethodInfo[]
GetMethods()

[C#]
public MethodInfo[] GetMethods()
```

## Summary

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

## Return Value

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

## Description

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

# Module.ToString() Method

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

## Summary

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

## Return Value

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

## Description

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

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

## Module.Assembly Property

```
[ILAsm]
.property class System.Reflection.Assembly Assembly { public hidebysig
specialname instance class System.Reflection.Assembly get_Assembly() }

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

### Summary

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

### Property Value

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

### Description

This property is read-only.

# Module.FullyQualifiedName Property

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

## Summary

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

## Property Value

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

## Description

This property is read-only.

## Behaviors

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

## How and When to Override

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

## Usage

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

## 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  
3

# Module.Name Property

```
[ILAsm]
.property string Name { public hidebysig specialname instance string
get_Name() }

[C#]
public string Name { get; }
```

## Summary

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

## Property Value

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

## Description

This property is read-only.

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

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

]