

System.Security.Permissions.ReflectionPermissionAttribute Class

```
[ILAsm]
.class public sealed serializable ReflectionPermissionAttribute extends
System.Security.Permissions.CodeAccessSecurityAttribute

[C#]
public sealed class ReflectionPermissionAttribute :
CodeAccessSecurityAttribute
```

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)

Type Attributes:

- AttributeUsageAttribute(AttributeTargets.Assembly | AttributeTargets.Class | AttributeTargets.Struct | AttributeTargets.Constructor | AttributeTargets.Method, AllowMultiple=true, Inherited=false)

Summary

Used to declaratively specify security actions to control access to non-public types using reflection.

Inherits From: System.Security.Permissions.CodeAccessSecurityAttribute

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

[Note: The level of access to non-public types and members is specified using the `System.Security.Permissions.ReflectionPermissionAttribute.Flags` property and the `System.Security.Permissions.ReflectionPermissionFlag` enumeration.

The security information declared by a security attribute is stored in the metadata of the attribute target, and is accessed by the system at run-time. Security attributes are used for declarative security only. For imperative security, use the corresponding permission

```
1    class, System.Security.Permissions.ReflectionPermission.  
2  
3    The allowable System.Security.Permissions.ReflectionPermissionAttribute  
4    targets are determined by the System.Security.Permissions.SecurityAction passed  
5    to the constructor.  
6  
7    ]
```

8 **Example**

9

10 The following example shows a declarative request for access to non-public members of
11 loaded assemblies. The
12 System.Security.Permissions.SecurityAction.RequestMinimum security action
13 indicates that this is the minimum permission required for the target assembly to be
14 able to execute.

```
15  
16 [assembly:ReflectionPermissionAttribute(SecurityAction.RequestMinimum,  
17 MemberAccess=true)]
```

18

19 The following example shows how to demand that the calling code has unrestricted
20 access to non-public types. Demands are typically made to protect methods or classes
21 from malicious code.

22

```
23 [ReflectionPermissionAttribute(SecurityAction.Demand, Unrestricted=true)]
```

24

ReflectionPermissionAttribute(System.Security.Permissions.SecurityAction) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(valuetype
System.Security.Permissions.SecurityAction action)

[C#]
public ReflectionPermissionAttribute(SecurityAction action)
```

Summary

Constructs and initializes a new instance of the `System.Security.Permissions.ReflectionPermissionAttribute` class with the specified `System.Security.Permissions.SecurityAction` value.

Parameters

Parameter	Description
<i>action</i>	A <code>System.Security.Permissions.SecurityAction</code> value.

Exceptions

Exception	Condition
System.ArgumentException	<i>action</i> is not a valid <code>System.Security.Permissions.SecurityAction</code> value.

ReflectionPermissionAttribute.CreatePermission() Method

```
[ILAsm]  
.method public hidebysig virtual class System.Security.IPermission  
CreatePermission()  
  
[C#]  
public override IPermission CreatePermission()
```

Summary

Returns a new `System.Security.Permissions.ReflectionPermission` that contains the security information of the current instance.

Return Value

A new `System.Security.Permissions.ReflectionPermission` object with the security information of the current instance.

Description

[*Note:* Applications typically do not call this method; it is intended for use by the system.

The security information described by a security attribute is stored in the metadata of the attribute target, and is accessed by the system at run-time. The system uses the object returned by this method to convert the security information of the current instance into the form stored in metadata.

This method overrides `System.Security.Permissions.SecurityAttribute.CreatePermission`.

]

ReflectionPermissionAttribute.Flags Property

```
[ILAsm]  
.property valuetype System.Security.Permissions.ReflectionPermissionFlag  
Flags { public hidebysig specialname instance valuetype  
System.Security.Permissions.ReflectionPermissionFlag get_Flags() public  
hidebysig specialname instance void set_Flags(valuetype  
System.Security.Permissions.ReflectionPermissionFlag value) }  
  
[C#]  
public ReflectionPermissionFlag Flags { get; set; }
```

Summary

Gets or sets levels of access to non-public types using reflection.

Property Value

One or more of the `System.Security.Permissions.ReflectionPermissionFlag` values.

Description

[*Note:* To specify multiple `System.Security.Permissions.ReflectionPermissionFlag` values for a set operation, use the bitwise OR operator.]