System.Security.Permissions.ReflectionPermission Class

```
[ILAsm]
.class public sealed serializable ReflectionPermission extends
System.Security.CodeAccessPermission

[C#]
public sealed class ReflectionPermission: CodeAccessPermission
```

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)

Implements:

- **System.Security.IPermission**

Summary

Secures access to the metadata of non-public types and members through reflection.

Inherits From: System.Security.CodeAccessPermission

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

Code with the appropriate `System.Security.Permissions.ReflectionPermission` has access to non-public members of a type. Without `System.Security.Permissions.ReflectionPermission`, code can access only the public members of assemblies.

[*Note:* Without `System.Security.Permissions.ReflectionPermission`, untrusted code can perform the following operations on members of loaded assemblies:

- Obtain type information from metadata for public types and members.

- Invoke public members.
- Invoke members defined with family access in the calling code's base classes.
- Invoke members defined with assembly access in the calling code's assembly.
- Invoke members defined with `FamilyAndAssembly` or `FamilyOrAssembly` access in the calling code's base classes and/or assembly.
- Enumerate assemblies.
- Enumerate public types.
- Enumerate types in the calling code's assembly.

]

`System.Security.Permissions.ReflectionPermission` instances can allow untrusted code to obtain type and member information, invoke members, and enumerate types that would otherwise be inaccessible. [*Note:* Because `System.Security.Permissions.ReflectionPermission` can provide access to members and information that were not intended for public access, it is recommended that `System.Security.Permissions.ReflectionPermission` be granted only to trusted code.]

The XML encoding of a `System.Security.Permissions.ReflectionPermission` instance is defined below in EBNF format. The following conventions are used:

- All non-literals in the grammar below are shown in normal type.
- All literals are in bold font.

The following meta-language symbols are used:

- '*' represents a meta-language symbol suffixing an expression that can appear zero or more times.
- '?' represents a meta-language symbol suffixing an expression that can appear zero or one time.
- '+' represents a meta-language symbol suffixing an expression that can appear one or more times.
- '(',')' is used to group literals, non-literals or a mixture of literals and non-literals.
- '|' denotes an exclusive disjunction between two expressions.

- '::= ' denotes a production rule where a left hand non-literal is replaced by a right hand expression containing literals, non-literals or both.

BuildVersion refers to the build version of the shipping CLI. This is specified as a dotted build number such as '2412.0'.

ECMAPubKeyToken::= b77a5c561934e089

ReflectionPermissionFlag = MemberAccess | TypeInformation

Each ReflectionPermissionFlag can appear in the XML no more than once. For example, Flags=MemberAccess,MemberAccess is illegal.

The XML encoding of a System.Security.Permissions.ReflectionPermission instance is as follows:

ReflectionPermissionXML::=

```
<IPermission

class="

System.Security.Permissions.ReflectionPermission, mscorlib,

Version=1.0.BuildVersion,

Culture=neutral,

PublicKeyToken=ECMAPubKeyToken"

version="1"

(

Unrestricted="true"

)
```

```
|


(

Flags="NoFlags | (ReflectionPermissionFlag (,ReflectionPermissionFlag)*"


)


/>

```

ReflectionPermission(System.Security.Permissions.ReflectionPermissionFlag) Constructor

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

[C#]
public ReflectionPermission(ReflectionPermissionFlag flag)
```

Summary

Constructs and initializes a new instance of the `System.Security.Permissions.ReflectionPermission` class with the specified access.

Parameters

Parameter	Description
flag	One or more `System.Security.Permissions. ReflectionPermissionFlag` values.

Exceptions

Exception	Condition
System.ArgumentException	The *flag* parameter contains a value that is not a combination of `System.Security.Permissions. ReflectionPermissionFlag` values.

ReflectionPermission.Copy() Method

```
[ILAsm]
.method public hidebysig virtual class System.Security.IPermission Copy()

[C#]
public override IPermission Copy()
```

Summary

Returns a new `System.Security.Permissions.ReflectionPermission` object containing the same values as the current instance.

Return Value

A new `System.Security.Permissions.ReflectionPermission` instance that contains the same values as the current instance.

Description

[*Note:* The object returned by this method represents the same access to resources as the current instance.

This method overrides `System.Security.CodeAccessPermission.Copy` and is implemented to support the `System.Security.IPermission` interface.

]

ReflectionPermission.FromXml(System.Security.SecurityElement) Method

```
[ILAsm]
.method public hidebysig virtual void FromXml(class
System.Security.SecurityElement esd)

[C#]
public override void FromXml(SecurityElement esd)
```

Summary

Reconstructs the state of a System.Security.Permissions.ReflectionPermission object using the specified XML encoding.

Parameters

Parameter	Description
esd	A System.Security.SecurityElement instance containing the XML encoding to use to reconstruct the state of a System.Security.Permissions.ReflectionPermission object.

Description

The state of the current instance is changed to the state encoded in *esd*.

[*Note:* For the XML encoding for this class, see the System.Security.Permissions.ReflectionPermission class page.

This method overrides System.Security.CodeAccessPermission.FromXml.

]

Exceptions

Exception	Condition
System.ArgumentNullException	The *esd* parameter is null.

System.ArgumentException	The *esd* parameter is not a `System.Security.Permissions.ReflectionPermission` element. -or- The *esd* parameter's version number is not valid.

ReflectionPermission.Intersect(System.Security.IPermission) Method

```
[ILAsm]
.method public hidebysig virtual class System.Security.IPermission
Intersect(class System.Security.IPermission target)

[C#]
public override IPermission Intersect(IPermission target)
```

Summary

Returns a new `System.Security.Permissions.ReflectionPermission` object that is the intersection of the current instance and the specified object.

Parameters

Parameter	Description
target	A `System.Security.Permissions.ReflectionPermission` instance to intersect with the current instance.

Return Value

A new `System.Security.Permissions.ReflectionPermission` instance that represents the intersection of the current instance and *target*. If the intersection is empty, returns `null`. If *target* is `null`, returns `null`.

Description

[*Note:* The intersection of two permissions is a permission that secures the resources and operations secured by both permissions. Specifically, it represents the minimum permission such that any demand that passes both permissions will also pass their intersection.

This method overrides `System.Security.CodeAccessPermission.Intersect` and is implemented to support the `System.Security.IPermission` interface.

]

Exceptions

Exception	Condition
System.ArgumentException	The *target* parameter is not `null` and is not an instance of `System.Security.Permissions.ReflectionPermission`.

ReflectionPermission.IsSubsetOf(System.Security.IPermission) Method

```
[ILAsm]
.method public hidebysig virtual bool IsSubsetOf(class
System.Security.IPermission target)

[C#]
public override bool IsSubsetOf(IPermission target)
```

Summary

Determines whether the current instance is a subset of the specified object.

Parameters

Parameter	Description
target	A `System.Security.Permissions.ReflectionPermission` instance that is to be tested for the subset relationship.

Return Value

`true` if the current instance is a subset of *target*; otherwise, `false`. If the current instance is unrestricted, and *target* is not, returns `false`. If *target* is unrestricted, returns `true`. If target is `null` and the access level of the current instance is `System.Security.Permissions.ReflectionPermissionFlag.NoFlags`, returns `true`. If target is `null` and the access level of the current instance is any value other than `System.Security.Permissions.ReflectionPermissionFlag.NoFlags`, returns `false`.

Description

[*Note:* The current instance is a subset of *target* if the current instance specifies a set of accesses to resources that is wholly contained by *target*. For example, a permission that represents access to type information is a subset of a permission that represents access to type information and members.

This method overrides `System.Security.CodeAccessPermission.IsSubsetOf` and is implemented to support the `System.Security.IPermission` interface.

]

Exceptions

Exception	Condition
System.ArgumentException	The *target* parameter is not `null` and is not an instance of `System.Security.Permissions.ReflectionPermission`.

ReflectionPermission.ToXml() Method

```
[ILAsm]
.method public hidebysig virtual class System.Security.SecurityElement
ToXml()

[C#]
public override SecurityElement ToXml()
```

Summary

Returns the XML encoding of the current instance.

Return Value

A `System.Security.SecurityElement` containing the XML encoding of the state of the current instance.

Description

[*Note:* For the XML encoding for this class, see the `System.Security.Permissions.ReflectionPermission` class page.

This method overrides `System.Security.CodeAccessPermission.ToXml`.

]

ReflectionPermission.Union(System.Security.IPermission) Method

```
[ILAsm]
.method public hidebysig virtual class System.Security.IPermission
Union(class System.Security.IPermission other)

[C#]
public override IPermission Union(IPermission other)
```

Summary

Returns a new System.Security.Permissions.ReflectionPermission object that is the union of the current instance and the specified object.

Parameters

Parameter	Description
other	A System.Security.Permissions.ReflectionPermission instance to be combined with the current instance.

Return Value

A new System.Security.Permissions.ReflectionPermission instance that represents the union of the current instance and *other*. If the current instance or *other* is unrestricted, returns a System.Security.Permissions.ReflectionPermission instance that is unrestricted. If *other* is null, returns a copy of the current instance.

Description

[*Note:* The result of a call to System.Security.Permissions.ReflectionPermission.Union is a permission that represents all of the access to resources represented by both the current instance and *other*. Any demand that passes either the current instance or *other* passes their union.

This method overrides System.Security.CodeAccessPermission.Union and is implemented to support the System.Security.IPermission interface.

]

Exceptions

Exception	Condition
System.ArgumentException	The *other* parameter is not `null` and is not an instance of `System.Security.Permissions.ReflectionPermission`.