System.Security.CodeAccessPermission Class

[ILAsm]
```
.class public abstract serializable CodeAccessPermission extends
System.Object implements System.Security.IPermission
```

[C#]
```
public abstract class CodeAccessPermission: IPermission
```

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

Serves as the base class for all code access permissions.

Inherits From: System.Object

Library: BCL

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:* Classes derived from `System.Security.CodeAccessPermission` are required to override the following methods of the `System.Security.CodeAccessPermission` class:

- `System.Security.CodeAccessPermission.Copy` - Creates a `System.Security.IPermission` object of the same type and containing the same values as the current instance.

- `System.Security.CodeAccessPermission.FromXml` - Reconstructs the state of a `System.Security.CodeAccessPermission` object using an XML encoding.

- `System.Security.CodeAccessPermission.Intersect` - Returns a `System.Security.IPermission` object that is the intersection of the current instance and the specified object.

- `System.Security.CodeAccessPermission.IsSubsetOf` - Determines if the current instance is a subset of the specified object.
- `System.Security.CodeAccessPermission.ToXml` - Creates an XML encoding of the current instance.
- `System.Security.CodeAccessPermission.Union` - Returns a `System.Security.IPermission` object that is the union of the current instance and the specified object.

In addition, classes derived from `System.Security.CodeAccessPermission` are required to implement a constructor that takes a `System.Security.Permissions.PermissionState` as its only parameter.

]

The XML encoding of a `System.Security.CodeAccessPermission` 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.

ClassName is the name of the class implementing the permission, such as `System.Security.Permissions.EnvironmentPermission`.

AssemblyName is the name of the assembly that contains the class implementing the permission, such as mscorlib.

Version is the three part version number indicating the version of the assembly implementing the permission, such as 1.0.1.

StrongNamePublicKeyToken is the strong name public key token constituting the strong name of the assembly that implements the permission.

PermissionAttributes is any attribute and attribute value on the `System.Security.IPermission` element used by the permission to represent a particular permission state, for example, unrestricted="true".

PermissionXML is any valid XML used by the permission to represent permission state.

The XML encoding of a `System.Security.CodeAccessPermission` instance is as follows:

```
CodeAccessPermissionXML::=

<IPermission class="

ClassName,

AssemblyName,

Version=Version,

Culture=neutral,

PublicKeyToken=StrongNamePublicKeyToken"

version="1"

(PermissionAttributes)*

>

(PermissionXML)?

</IPermission>
```

CodeAccessPermission() Constructor

```
[ILAsm]
family rtspecialname specialname instance void .ctor()

[C#]
protected CodeAccessPermission()
```

Summary

Constructs a new instance of the `System.Security.CodeAccessPermission` class.

CodeAccessPermission.Assert() Method

```
[ILAsm]
.method public final hidebysig virtual void Assert()

[C#]
public void Assert()
```

Summary

Asserts that calling code can access the resource identified by the current instance through the code that calls this method, even if callers have not been granted permission to access the resource.

Description

Calling `System.Security.CodeAccessPermission.Assert` stops the permission check on callers that are after the code performing the assert. An assertion is effective only if the code that calls `System.Security.CodeAccessPermission.Assert` passes the security check for the permission that it is asserting.

[*Note:* Even if the callers that are after the code performing the assert do not have the requisite permissions, they can still access resources through the code that calls this method. Because the assertion only applies to the callers of the code performing the assert, a security check for the asserted permission can still fail if the code calling `System.Security.CodeAccessPermission.Assert` has not itself been granted that permission.

A call to `System.Security.CodeAccessPermission.Assert` is effective until the code containing the call returns to its caller.

Caution: Because calling `System.Security.CodeAccessPermission.Assert` removes the requirement that all code be granted permission to access the specified resource, it can open up security vulnerabilities if used incorrectly or inappropriately.

]

Exceptions

Exception	Condition
System.Security.SecurityException	The calling code does not have `System.Security.Permissions.SecurityPermissionFlag.Assertion`.

Permissions

Permission	Description

System.Security.Permissions. SecurityPermission	Requires permission to call `System.Security.CodeAccessPermission.Assert`. See `System.Security.Permissions.SecurityPermissionFlag. Assertion`.

CodeAccessPermission.Copy() Method

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

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

Summary

Returns a `System.Security.CodeAccessPermission` containing the same values as the current instance.

Return Value

A new `System.Security.CodeAccessPermission` instance that is value equal to the current instance.

Description

[*Note:* This method is implemented to support the `System.Security.IPermission` interface.]

Behaviors

The object returned by this method is required be the same type as the current instance and to represent the same access to resources as the current instance.

How and When to Override

Override this method to create a copy an instance in a type derived from `System.Security.CodeAccessPermission`.

Usage

Use this method to obtain a copy of the current instance that has values identical to those of the current instance.

CodeAccessPermission.Demand() Method

```
[ILAsm]
.method public final hidebysig virtual void Demand()

[C#]
public void Demand()
```

Summary

Forces a `System.Security.SecurityException` if all callers do not have the permission specified by the current instance.

Description

The permissions of the code that calls this method are not examined; the check begins from the immediate caller of that code and continues until all callers have been checked, one of the callers invokes `System.Security.CodeAccessPermission.Assert`, or a caller has been found that is not granted the demanded permission, in which case a `System.Security.SecurityException` is thrown.

[*Note:* `System.Security.CodeAccessPermission.Demand` is typically used by shared libraries to ensure that callers have permission to access a resource. For example, a method in a shared library calls `System.Security.CodeAccessPermission.Demand` for the necessary `System.Security.Permissions.FileIOPermission` before performing a file operation requested by the caller.

This method is implemented to support the `System.Security.IPermission` interface.

]

Exceptions

Exception	Condition
System.Security.SecurityException	A caller does not have the permission specified by the current instance. -or- A caller has called `System.Security.CodeAccessPermission.Deny` for the resource protected by the current instance.

CodeAccessPermission.Deny() Method

```
[ILAsm]
.method public final hidebysig virtual void Deny()

[C#]
public void Deny()
```

Summary

Denies access to the resources specified by the current instance through the code that calls this method.

Description

This method prevents callers from accessing the protected resource through the code that calls this method, even if those callers have been granted permission to access it.

The call to `System.Security.CodeAccessPermission.Deny` is effective until the calling code returns.

[*Note:* `System.Security.CodeAccessPermission.Deny` is ignored for a permission not granted because a demand for that permission will not succeed.

`System.Security.CodeAccessPermission.Deny` can limit the liability of the programmer or prevent accidental security vulnerabilities because it prevents the method that calls `System.Security.CodeAccessPermission.Deny` from being used to access the resource protected by the denied permission.

]

CodeAccessPermission.FromXml(System.Security.SecurityElement) Method

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

[C#]
public abstract void FromXml(SecurityElement elem)
```

Summary

Reconstructs the state of a `System.Security.CodeAccessPermission` object using the specified XML encoding.

Parameters

Parameter	Description
elem	A `System.Security.SecurityElement` instance containing the XML encoding to use to reconstruct the state of a `System.Security.CodeAccessPermission` object.

Description

Behaviors

The values of the current instance are set to the values of the permission object encoded in *elem*.

How and When to Override

Override this method to reconstruct subclasses of `System.Security.CodeAccessPermission`.

Usage

This method is called by the system.

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

Exceptions

Exception	**Condition**
System.ArgumentException	*elem* does not contain the XML encoding for a instance of the same type as the current instance. -or- The version number of *elem* is not valid.

CodeAccessPermission.Intersect(System.Security.IPermission) Method

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

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

Summary

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

Parameters

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

Return Value

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

Description

[*Note:* This method is implemented to support the `System.Security.IPermission` interface.]

Behaviors

As described above.

How and When to Override

Override this method to provide a mechanism for creating an intersection of two `System.Security.IPermission` objects that are of the same type and are derived from `System.Security.CodeAccessPermission`.

Usage

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.

Exceptions

Exception	Condition
System.ArgumentException	*target* is not `null` and is not a `System.Security.CodeAccessPermission` object.

CodeAccessPermission.IsSubsetOf(System.Security.IPermission) Method

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

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

Summary

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

Parameters

Parameter	Description
target	A `System.Security.CodeAccessPermission` 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`.

Description

[*Note:* This method is implemented to support the `System.Security.IPermission` interface.]

Behaviors

As described above.

How and When to Override

Override this method to implement the test for the subset relationship in types derived from `System.Security.CodeAccessPermission`.

Usage

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 read access to a file is a subset of a permission that represents read and write access to the file.

If this method returns `true`, the current instance does not describe a level of access to a set of resources that is not already described by *target*.

Exceptions

Exception	**Condition**
System.ArgumentException	*target* is not `null` and is not of type `System.Security.CodeAccessPermission`.

CodeAccessPermission.ToString() Method

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

[C#]
public override string ToString()
```

Summary

Returns the XML representation of the state of the current instance.

Return Value

A `System.String` containing the XML representation of the state of the current instance.

Description

[*Note:* The XML representation of the current instance is obtained by first calling `System.Security.CodeAccessPermission.ToXml`, then calling `System.Object.ToString` on the object returned by that method.

This method overrides `System.Object.ToString`.

]

CodeAccessPermission.ToXml() Method

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

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

Summary

Returns the XML encoding of the current instance.

Return Value

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

Behaviors

The object returned by this method is required to use the XML encoding for the `System.Security.CodeAccessPermission` class as defined on the class page. The state of the current instance is required to be reproducible by invoking `System.Security.CodeAccessPermission.FromXml` on an instance of `System.Security.CodeAccessPermission` using the object returned by this method.

How and When to Override

Override this method to return an object containing the XML encoding for types derived from `System.Security.CodeAccessPermission`.

Usage

This method is called by the system.

CodeAccessPermission.Union(System.Security.IPermission) Method

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

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

Summary

Returns a `System.Security.CodeAccessPermission` object that is the union of the current instance and the specified object.

Parameters

Parameter	Description
other	A `System.Security.IPermission` object of the same type as the current instance to be combined with the current instance.

Return Value

If *other* is `null`, returns a copy of the current instance using the `System.Security.IPermission.Copy` method.

Description

[*Note:* This method is implemented to support the `System.Security.IPermission` interface.]

Behaviors

This method returns a new `System.Security.CodeAccessPermission` instance that represents the union of the current instance and *other*. If the current instance or *other* is unrestricted, returns a `System.Security.CodeAccessPermission` instance that is unrestricted. If *other* is `null`, returns a copy of the current instance using the `System.Security.IPermission.Copy` method.

Default

If *other* is not `null`, this method throws a `System.NotSupportedException` exception; otherwise, returns a copy of the current instance.

How and When to Override

Override this method to provide a mechanism for creating the union of two `System.Security.IPermission` objects that are of the same type and are derived from `System.Security.CodeAccessPermission`.

Usage

The result of a call to `System.Security.CodeAccessPermission.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 permission passes their union.

Exceptions

Exception	**Condition**
System.ArgumentException	*other* is not of type `System.Security.CodeAccessPermission`.
System.NotSupportedException	*other* is not `null`.