System.Security.IPermission Interface

```
[ILASM]
.class interface public abstract IPermission

[C#]
public interface IPermission
```

Assembly Info:

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

Summary

Defines methods implemented by permission types.

Library: BCL

Description

[*Note:* Permission types describe a level of access to secured operations or resources, such as files or environment variables. Permission instances do not confer the right to access a resource or perform an operation; the security system determines whether or not requested permissions are granted. Permissions are used by both application code and the security system in the following ways:

- Code requests the permissions it needs in order to run.
- Permissions are granted by the security system.
- Code demands that calling code has a permission.
- Code alters the default behavior of the security system by asserting or denying permissions.

]

IPermission.Copy() Method

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

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

Summary

Returns a **System.Security.IPermission** object of the same type and containing the same values as the current instance.

Return Value

A new **System.Security.IPermission** object of the same type and containing the same values as the current instance.

Behaviors

The object returned by this method is required to be a deep copy of the current instance; any objects referenced by the current instance are duplicated in the copy.

How and When to Override

Implement this method to provide the system with a means of duplicating permission objects.

Usage

Use this method to obtain a copy of the current instance that is identical to the current instance.

IPermission.Demand() Method

```
[ILASM]
.method public hidebysig virtual abstract void Demand()

[C#]
void Demand()
```

Summary

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

Description

The permission check for **System.Security.IPermission.Demand** begins with the immediate caller of the code that calls this method and continues until all callers have been checked or a caller has been found that is not granted the demanded permission, in which case a **System.Security.SecurityException** exception is thrown.

Behaviors

Notifies the system that a security check is required for all callers of the method that invokes **System.Security.IPermission.Demand**. All callers are required to have the permissions described by the current instance. If one of the callers asserts the permissions and that caller has the permissions, the system is required to allow all callers that have not been checked to bypass the security check. If the security check fails, a **System.Security.SecurityException** is thrown by the system.

Usage

Use this method to ensure that callers that call the method containing a **System.Security.IPermission.Demand** call have the permissions described by the current instance.

Exceptions

Exception	Condition
System.Security.SecurityException	A caller does not have the permission specified by the current instance.

IPermission.Intersect(System.Security.IPermission) Method

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

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

Summary

Returns an object that is the intersection of the current instance and the specified **System.Security.IPermission** object.

Parameters

Parameter	Description
target	An object of the same type as the current instance to intersect with the current instance.

Return Value

A new **System.Security.IPermission** object that is the same type as the current instance and represents the intersection of the current instance and *target*. If the intersection is empty, or *target* is **null**, returns **null**. If the current instance is unrestricted, returns a copy of *target*. If *target* is unrestricted, returns a copy of the current instance.

Description

The object returns by **System.Security.IPermission.Intersect** is a permission that secures the resources and operations secured by two **System.Security.IPermission** objects: a demand passes the intersection of two objects only if it passes both of the objects.

Behaviors

If *target* is not **null** and is not of the same type as the current instance, a **System.ArgumentException** exception is required to be thrown.

The following statements are required to be true for all implementations of the **System.Security.IPermission.Intersect** method. *X* and *Y* represent non-null **System.Security.IPermission** object references.

- *X*.Intersect(*X*) returns a value equal to *X*.
- *X*.Intersect(*Y*) returns the same value as *Y*.Intersect(*X*).
- *X*.Intersect(**null**) returns **null**.

Usage

Use this method to obtain the set of permissions that are described both by the current instance and the specified object.

Exceptions

Exception	Condition
System.ArgumentException	*target* is not **null** and is not of the same type as the current instance.

IPermission.IsSubsetOf(System.Security.IPermission) Method

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

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

Summary

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

Parameters

Parameter	Description
target	A **System.Security.IPermission** object of the same type as the current 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 current instance does not secure any resources or operations, returns **true**. If target is **null** and the current instance secures one or more resources or operations, returns **false**.

Description

The current instance is a subset of *target* if the current instance specifies a set of accesses to resources or operations 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.

Behaviors

If *target* is not **null** and is not of the same type as the current instance, a **System.ArgumentException** exception is required to be thrown.

The following statements are required to be true for all implementations of the **System.Security.IPermission.IsSubsetOf** method. *X*, *Y*, and *Z* represent non-null **System.Security.IPermission** objects.

- *X*.IsSubsetOf(*X*) returns **true**.
- *X*.IsSubsetOf(*Y*) returns the same value as *Y*.IsSubsetOf(*X*) if and only if *X* and *Y* represent the same set of permissions.
- if *X*.IsSubsetOf(*Y*) and *Y*.IsSubsetOf(*Z*) both return **true**, *X*.IsSubsetOf(*Z*) returns **true**.

Usage

Use this method to determine if the permissions described by the current instance are also described by the specified object.

Exceptions

Exception	Condition
System.ArgumentException	*target* is not **null** and is not of the same type as the current instance.

IPermission.Union(System.Security.IPermission) Method

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

[C#]
```
IPermission Union(IPermission target)
```

Summary

Returns an object that is the union of the current instance and the specified object.

Parameters

Parameter	Description
target	A **System.Security.IPermission** object of the same type as the current instance to combine with the current instance.

Return Value

A new **System.Security.IPermission** object of the same type as the current instance that represents the union of the current instance and *target*.

Description

The object returned by **System.Security.IPermission.Union** is a permission that represents the permissions described by the current instance and those described by *target*. Any demand that passes either the current instance or *target* is required to pass the union of the two.

Behaviors

If *target* is not **null** and is not of the same type as the current instance, a **System.ArgumentException** exception is required to be thrown.

The following statements are required to be true for all implementations of the **System.Security.IPermission.Union** method. *X* and *Y* represent non-null **System.Security.IPermission** objects.

- *X*.Union(*X*) returns an object that is value-equal to *X*.
- *X*.Union(*Y*) returns an object that is value-equal to the object returned by *Y*.Union(*X*).
- *X*.Union(**null**) returns an object that is value-equal to *X*.

Usage

Use this method to obtain a **System.Security.IPermission** object of the same type as the current instance and *target* that describes the permissions described by the current instance and those described by *target*.

Exceptions

Exception	Condition
System.ArgumentException	*target* is not **null** and is not of the same type as the current instance.