

1 System.Security.IPermission Interface

```
2 [ILAsm]  
3 .class interface public abstract IPermission  
  
4 [C#]  
5 public interface IPermission
```

6 Assembly Info:

- 7 • *Name:* mscorlib
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

13 Defines methods implemented by permission types.

14 **Library:** BCL

15

16 Description

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

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

27]

28

1 IPermission.Copy() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract class  
4 System.Security.IPermission Copy()  
  
5 [C#]  
6 IPermission Copy()
```

7 Summary

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

10 Return Value

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

13 Behaviors

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

16

17 How and When to Override

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

20

21 Usage

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

24

25

1 IPermission.Demand() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void Demand()  
4 [C#]  
5 void Demand()
```

6 Summary

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

9 Description

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

14 Behaviors

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

21

22 Usage

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

26

27 Exceptions

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

28

29

1
2 **IPermission.Intersect(System.Security.IPerm**
3 **ission) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract class  
6 System.Security.IPermission Intersect(class System.Security.IPermission  
7 target)  
8 [C#]  
9 IPermission Intersect(IPermission target)
```

10 **Summary**

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

13 **Parameters**

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

14
15 **Return Value**

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

19 **Description**

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

24 **Behaviors**

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

27
28 The following statements are required to be true for all implementations of the
29 System.Security.IPermission.Intersect method. X and Y represent non-null
30 System.Security.IPermission object references.

- 31
- X.Intersect(X) returns a value equal to X.

- 1 • `X.Intersect(Y)` returns the same value as `Y.Intersect(X)`.
- 2 • `X.Intersect(null)` returns `null`.

3 Usage

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

6

7 Exceptions

Exception	Condition
System.ArgumentException	<i>target</i> is not <code>null</code> and is not of the same type as the current instance.

8

9

1 2 **IPermission.IsSubsetOf(System.Security.IPer** 3 **mission) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract bool IsSubsetOf(class  
6 System.Security.IPermission target)  
7 [C#]  
8 bool IsSubsetOf(IPermission target)
```

9 **Summary**

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

11 **Parameters**

Parameter	Description
<i>target</i>	A System.Security.IPermission object of the same type as the current instance that is to be tested for the subset relationship.

12 13 **Return Value**

14 true if the current instance is a subset of *target*; otherwise, false. If the current
15 instance is unrestricted, and *target* is not, returns false. If *target* is unrestricted,
16 returns true. If *target* is null and the current instance does not secure any resources
17 or operations, returns true. If *target* is null and the current instance secures one or
18 more resources or operations, returns false.

19 **Description**

20 The current instance is a subset of *target* if the current instance specifies a set of
21 accesses to resources or operations that is wholly contained by *target*. For example, a
22 permission that represents read access to a file is a subset of a permission that
23 represents read and write access to the file.

24 **Behaviors**

- 25 If *target* is not null and is not of the same type as the current instance, a
26 System.ArgumentException exception is required to be thrown.
27
- 28 The following statements are required to be true for all implementations of the
29 System.Security.IPermission.IsSubsetOf method. X, Y, and Z represent non-null
30 System.Security.IPermission objects.
- 31 • X.IsSubsetOf(X) returns true.

- 1
- `X.IsSubsetOf(Y)` returns the same value as `Y.IsSubsetOf(X)` if and only if `X` and `Y`
- 2 represent the same set of permissions.
- if `X.IsSubsetOf(Y)` and `Y.IsSubsetOf(Z)` both return `true`, `X.IsSubsetOf(Z)` returns
- 3 `true`.
- 4

5 **Usage**

6 Use this method to determine if the permissions described by the current instance are

7 also described by the specified object.

8

9 **Exceptions**

Exception	Condition
System.ArgumentException	<i>target</i> is not <code>null</code> and is not of the same type as the current instance.

10

11

1
2 **IPermission.Union(System.Security.IPermissi**
3 **on) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract class  
6 System.Security.IPermission Union(class System.Security.IPermission  
7 target )  
8 [C#]  
9 IPermission Union(IPermission target)
```

10 **Summary**

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

12 **Parameters**

Parameter	Description
<i>target</i>	A System.Security.IPermission object of the same type as the current instance to combine with the current instance.

13
14 **Return Value**

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

17 **Description**

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

22 **Behaviors**

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

25
26 The following statements are required to be true for all implementations of the
27 System.Security.IPermission.Union method. X and Y represent non-null
28 System.Security.IPermission objects.

- 29
- X.Union(X) returns an object that is value-equal to X.

1 • `X.Union(Y)` returns an object that is value-equal to the object returned by
2 `Y.Union(X)`.

3 • `X.Union(null)` returns an object that is value-equal to `X`.

4 Usage

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

8

9 Exceptions

Exception	Condition
System.ArgumentException	<i>target</i> is not <code>null</code> and is not of the same type as the current instance.

10

11