System.Security.Permissions.EnvironmentPermission Class

```
[ILASM]
.class public sealed serializable EnvironmentPermission
extends System.Security.CodeAccessPermission

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

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)

Implements:

- **System.Security.IPermission**

Summary

Controls access to environment variables.

Inherits From: System.Security.CodeAccessPermission

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:* **System.Security.Permissions.EnvironmentPermission** objects describe protected operations on environment variables. This permission distinguishes between the following types of access provided by **System.Security.Permissions.EnvironmentPermissionAccess**:

- **Read**: Read values from environment variables.

- **Write**: Write values to environment variables. Also allows for creating and deleting values.

- **NoAccess**: No access to environment variables.
- **AllAccess**: Full access to environment variables. Identical to specifying **Read** and **Write** access.

These access levels are independent, meaning that rights to one do not imply rights to another. For example, **Write** permission does not imply permission to **Read**. **System.Security.Permissions.EnvironmentPermissionAccess** values can be combined using a bitwise OR operator.

The **System.Environment** class is used to access environment variables, subject to the permissions defined by **System.Security.Permissions.EnvironmentPermission**. Environment variables are case-insensitive.]

The XML encoding of a **System.Security.Permissions.EnvironmentPermission** 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**

EnvironmentVariable refers to the name of a single environment variable, such as 'PROMPT'.

The XML encoding of an **EnvironmentPermission** instance is as follows:

`EnvironmentPermissionXML::=`

<IPermission

class="

System.Security.Permissions.EnvironmentPermission,

mscorlib,

Version=1.0.`BuildVersion`**,**

Culture=neutral,

PublicKeyToken=`ECMAPubKeyToken`**"**

version="1"

`(`

Unrestricted="true"

`)`

|

`(`

`(`**Read="** `EnvironmentVariable (`**;** `EnvironmentVariable)*`**"**`)?`

`(`**Write="**`EnvironmentVariable (`**;** `EnvironmentVariable)*` **"**`)?`

`)`

/ >

EnvironmentPermission(System.Security.Permissions.PermissionState) Constructor

```
[ILASM]
public rtspecialname specialname instance void
.ctor(valuetype System.Security.Permissions.PermissionState
state)

[C#]
public EnvironmentPermission(PermissionState state)
```

Summary

Constructs and initializes a new instance of the **System.Security.Permissions.EnvironmentPermission** class with the specified **System.Security.Permissions.PermissionState** value.

Parameters

Parameter	Description
state	A **System.Security.Permissions.PermissionState** value.

Description

[*Note:* The instance returned by this constructor has either fully restricted (**System.Security.Permissions.PermissionState.None**) or unrestricted (**System.Security.Permissions.PermissionState.Unrestricted**) access to all environment variables.]

Exceptions

Exception	Condition
System.ArgumentException	*state* is not a valid **System.Security.Permissions.PermissionState** value.

EnvironmentPermission(System.Security.Permissions.EnvironmentPermissionAccess, System.String) Constructor

```
[ILASM]
public rtspecialname specialname instance void
.ctor(valuetype
System.Security.Permissions.EnvironmentPermissionAccess
flag, string pathList)

[C#]
public EnvironmentPermission(EnvironmentPermissionAccess
flag, string pathList)
```

Summary

Constructs a new instance of the **System.Security.Permissions.EnvironmentPermission** class with the specified access to the specified environment variables.

Parameters

Parameter	Description
flag	One of values defined by **System.Security.Permissions. EnvironmentPermissionAccess**.
pathList	A **System.String** containing one or more case-insensitive environment variable names separated by **System.IO.Path.PathSeparator**.

Description

The specified access is applied to all environment variables in *pathList*.

Exceptions

Exception	Condition
System.ArgumentNullException	*pathList* is **null**.
System.ArgumentException	*flag* specifies a value not defined in **System.Security.Permissions. EnvironmentPermissionAccess**.

EnvironmentPermission.Copy() Method

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

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

Summary

Returns a new **System.Security.Permissions.EnvironmentPermission** object containing the same values as the current instance.

Return Value

A new **System.Security.Permissions.EnvironmentPermission** containing the same values as the current instance.

Description

[*Note:* The object returned by this method represents the same level of access to the same environment variables as the current instance.]

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

EnvironmentPermission.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.EnvironmentPermission** 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.EnvironmentPermission** 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.EnvironmentPermission** class page.

This method overrides **System.Security.CodeAccessPermission.FromXml**.]

Exceptions

Exception	Condition
System.ArgumentNullException	*esd* is **null**.
System.ArgumentException	*esd* does not contain the encoding for a **System.Security.Permissions.EnvironmentPermissi** instance. The version number of *esd* is not valid.

EnvironmentPermission.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.EnvironmentPermission** object that is the intersection of the current instance and the specified object.

Parameters

Parameter	Description
target	A **System.Security.Permissions.EnvironmentPermission** instance to intersect with the current instance.

Return Value

A new **System.Security.Permissions.EnvironmentPermission** instance that 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

[*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	*target* is not **null** and is not of type **System.Security.Permissions.EnvironmentPermission**.

EnvironmentPermission.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.EnvironmentPermission** 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 no environment variables are secured by the current instance, returns **true**. If target is **null**, and the current instance secures one or more environment variables, 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 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 describes a level of access to a set of environment variables that is also described by *target*.

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

Exceptions

Exception	Condition
System.ArgumentException	*target* is not **null** and is not of type **System.Security.Permissions.EnvironmentPermission**.

EnvironmentPermission.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.EnvironmentPermission** class page.

This method overrides **System.Security.CodeAccessPermission.ToXml**.]

EnvironmentPermission.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.EnvironmentPermission** that is the union of the current instance and the specified object.

Parameters

Parameter	Description
other	A **System.Security.Permissions.EnvironmentPermission** instance to combine with the current instance.

Return Value

A new **System.Security.Permissions.EnvironmentPermission** instance that represents the union of the current instance and *other*. If the current instance or *other* is unrestricted, returns a **System.Security.Permissions.EnvironmentPermission** instance that is unrestricted. If *other* is **null**, returns a copy of the current instance via the **System.Security.IPermission.Copy** method. If the current instance and *other* do not specify any environment variables, returns **null**.

Description

[*Note:* The result of a call to **System.Security.Permissions.EnvironmentPermission.Union** is a permission that represents the access to environment variables represented by the current instance as well as the access to environment variables represented by *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	*other* is not **null** and is not of type **System.Security.Permissions.EnvironmentPermission**.