System.Security.Permissions.FileIOPermission Class

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

[C#]
public sealed class FileIOPermission: 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

Secures access to files and directories.

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

System.Security.Permissions.FileIOPermission objects describe protected operations on files and directories. Files and directories are specified using absolute paths. Case-sensitivity of files and directories is platform and file system dependent.

[*Note:* This permission distinguishes between the following types of file I/O access provided by **System.Security.Permissions.FileIOPermissionAccess**:

- **Read**: Read access to the contents of the file or access to information about the file, such as its length or last modification time.

- **Write**: Write access to the contents of the file or access to change information about the file, such as its name. Also allows for deletion and overwriting.
- **Append**: Ability to write to the end of a file only.
- **PathDiscovery**: Ability to obtain path information about a file.
- **NoAccess**: No access to the file or directory.
- **AllAccess**: Full access to the file or directory.

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** or **Append**.
System.Security.Permissions.FileIOPermissionAccess values can be combined using a bitwise OR operator.

For information on security considerations when accessing files, see **System.IO.FileStream**.]

The XML encoding of a **FileIOPermission** 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**

FileName refers to the full path and file name of a file, or to a path name, such as "C:\Temp\test.exe" or "C:\ ".

The XML encoding of a **FileIOPermission** instance is as follows:

FileIOPermissionXML::=

<IPermission class="

System.Security.Permissions.FileIOPermission,

mscorlib,

Version=1.0. BuildVersion**,**

Culture=neutral,

PublicKeyToken=ECMAPubKeyToken**"**

version="1"

(

Unrestricted="true"

)

|

(

(**Read="**FileName (**;** FileName)***"**) ?

(**Write="**FileName (**;** FileName)***"**) ?

(**Append="**FileName (**;** FileName)***"**) ?

(**PathDiscovery="**FileName (**;** FileName)***"**) ?

)

/>

FileIOPermission(System.Security.Permissions.PermissionState) Constructor

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

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

Summary

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

Parameters

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

Description

[*Note:* This constructor creates either fully restricted (**System.Security.Permissions.PermissionState.None**) or **System.Security.Permissions.PermissionState.Unrestricted** access to files and directories.]

Exceptions

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

FileIOPermission(System.Security.Permissions.FileIOPermissionAccess, System.String) Constructor

```
[ILASM]
public rtspecialname specialname instance void
.ctor(valuetype
System.Security.Permissions.FileIOPermissionAccess access,
string path)

[C#]
public FileIOPermission(FileIOPermissionAccess access,
string path)
```

Summary

Constructs and initializes a new instance of the **System.Security.Permissions.FileIOPermission** class with the specified access to the specified file or directory.

Parameters

Parameter	Description
access	One or more values defined in **System.Security.Permissions.FileIOPermissionAccess**. Specify multiple values using the bitwise OR operator.
path	The absolute path of the file or directory.

Description

The set of characters that are invalid for use in file or directory names is platform specific.

Exceptions

Exception	Condition
System.ArgumentException	*access* specifies values not defined in **System.Security.Permissions.FileIOPermissionAccess**. *path* contains one or more characters that are invalid for use in files or directory names. *path* did not specify the absolute path to the file or

	directory.

1
2
3

FileIOPermission.Copy() Method

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

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

Summary

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

Return Value

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

Description

[*Note:* The object returned by this method represents the same level of access to files and directories as the current instance.

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

FileIOPermission.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.FileIOPermission** 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.FileIOPermission** 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.FileIOPermission** 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.FileIOPermission** instance. The version number of *esd* is not valid.

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

Parameters

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

Return Value

A new **System.Security.Permissions.FileIOPermission** 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.FileIOPermission**.

1
2
3

FileIOPermission.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.FileIOPermission** 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 files or directories are secured by the current instance, returns **true**. If target is **null**, and the current instance secures one or more files or directories, 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 files and directories 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.FileIOPermission**.

FileIOPermission.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.FileIOPermission** class page.

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

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

Parameters

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

Return Value

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

Description

[*Note:* The result of a call to **System.Security.Permissions.FileIOPermission.Union** is a permission that represents all of the access to files and directories represented by the current instance as well as the access to files and directories 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.FileIOPermission**.