System.Security.Permissions.FileIOPermissionAttribute Class

```
[ILASM]
.class public sealed serializable FileIOPermissionAttribute
extends
System.Security.Permissions.CodeAccessSecurityAttribute

[C#]
public sealed class FileIOPermissionAttribute:
CodeAccessSecurityAttribute
```

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)

Type Attributes:

- AttributeUsageAttribute(AttributeTargets.Assembly | AttributeTargets.Class | AttributeTargets.Struct | AttributeTargets.Constructor | AttributeTargets.Method, AllowMultiple=true, Inherited=false)

Summary

Used to declaratively specify security actions to control access to files and directories.

Inherits From: System.Security.Permissions.CodeAccessSecurityAttribute

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:* The level of access to a file or directory is specified using the members of the current instance. For example, to specify read permissions for a file, set the **System.Security.Permissions.FileIOPermissionAttribute.Read** property equal to the name of the file.

The security information declared by a security attribute is stored in the metadata of the attribute target, and is accessed by the system at run-time. Security attributes are used for declarative security only. For imperative security, use the corresponding permission class, **System.Security.Permissions.FileIOPermission**.

The allowable **System.Security.Permissions.FileIOPermissionAttribute** targets are determined by the **System.Security.Permissions.SecurityAction** passed to the constructor.] Case-sensitivity of file and directory names is platform dependent. The set of characters that are valid for use in file and directory names is determined by the current file system.

Example

The following example shows a declarative request for full access to the specified file. The **System.Security.Permissions.SecurityAction.RequestMinimum** security action indicates that this is the minimum permission required for the target assembly to be able to execute.

```
[assembly:FileIOPermissionAttribute(SecurityAction.RequestM
inimum, All="\\example\\sample.txt")]
```

The following example shows how to demand that the calling code has unrestricted access to files and directories. Demands are typically made to protect methods or classes from malicious code.

```
[FileIOPermissionAttribute(SecurityAction.Demand,
Unrestricted=true)]
```

FileIOPermissionAttribute(System.Security.Permissions.SecurityAction) Constructor

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

[C#]
public FileIOPermissionAttribute(SecurityAction action)
```

Summary

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

Parameters

Parameter	Description
action	A **System.Security.Permissions.SecurityAction** value.

Exceptions

Exception	Condition
System.ArgumentException	*action* is not a valid **System.Security.Permissions.SecurityAction** value.

FileIOPermissionAttribute.CreatePermission() Method

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

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

Summary

Returns a new **System.Security.Permissions.FileIOPermission** that contains the security information of the current instance.

Return Value

A new **System.Security.Permissions.FileIOPermission** object with the security information of the current instance.

Description

[*Note:* Applications typically do not call this method; it is intended for use by the system.

The security information declared by a security attribute is stored in the metadata of the attribute target, and is accessed by the system at run-time. The system uses the object returned by this method to convert the security information of the current instance into the form stored in metadata.

This method overrides **System.Security.Permissions.SecurityAttribute.CreatePermission**.]

FileIOPermissionAttribute.All Property

[ILASM]

```
.property string All { public hidebysig specialname
instance void set_All(string value) }
```

[C#]

```
public string All { set; }
```

Summary

Sets the name of a file or directory for which full access is secured.

Property Value

A **System.String** containing the absolute path of the file or directory for which full access is secured.

Description

This property is write-only.

[*Note:* This property sets full access for a single file or directory; use additional **System.Security.Permissions.FileIOPermissionAttribute** attributes to specify additional files and directories.]

Exceptions

Exception	Condition
System.ArgumentException	The path information specified for a set operation contains invalid characters or wild card specifiers.

FileIOPermissionAttribute.Append Property

```
[ILASM]
.property string Append { public hidebysig specialname
instance string get_Append() public hidebysig specialname
instance void set_Append(string value) }

[C#]
public string Append { get; set; }
```

Summary

Gets or sets the name of a file or directory for which append access is secured.

Property Value

A **System.String** containing the absolute path of the file or directory for which append access is secured.

Description

[*Note:* This property sets append access for a single file or directory; use additional **System.Security.Permissions.FileIOPermissionAttribute** attributes to specify additional files and directories.]

Exceptions

Exception	Condition
System.ArgumentException	The path information specified for a set operation contains invalid characters or wild card specifiers.

FileIOPermissionAttribute.PathDiscovery Property

[ILASM]

```
.property string PathDiscovery { public hidebysig
specialname instance string get_PathDiscovery() public
hidebysig specialname instance void
set_PathDiscovery(string value) }
```

[C#]

```
public string PathDiscovery { get; set; }
```

Summary

Gets or sets the name of a file or directory for which path discovery access is secured.

Property Value

A **System.String** containing the absolute path of the file or directory for which access to the contents of the path is secured.

Description

[*Note:* This property sets path discovery access for a single file or directory; use additional **System.Security.Permissions.FileIOPermissionAttribute** attributes to specify additional files and directories.

Path discovery controls access to the information in the path itself. This protects sensitive information in the path, such as user names, as well as information about the directory structure revealed in the path. This value does not secure access to files or folders represented by the path.]

Exceptions

Exception	Condition
System.ArgumentException	The path information specified for a set operation contains invalid characters or wild card specifiers.

FileIOPermissionAttribute.Read Property

```
[ILASM]
.property string Read { public hidebysig specialname
instance string get_Read() public hidebysig specialname
instance void set_Read(string value) }

[C#]
public string Read { get; set; }
```

Summary

Gets or sets the name of a file or directory for which read access is secured.

Property Value

A **System.String** containing the absolute path of the file or directory for which read access is secured.

Description

[*Note:* This property sets read access for a single file or directory; use additional **System.Security.Permissions.FileIOPermissionAttribute** attributes to specify additional files and directories.]

Exceptions

Exception	Condition
System.ArgumentException	The path information specified for a set operation contains invalid characters or wild card specifiers.

FileIOPermissionAttribute.Write Property

```
[ILASM]
.property string Write { public hidebysig specialname
instance string get_Write() public hidebysig specialname
instance void set_Write(string value) }

[C#]
public string Write { get; set; }
```

Summary

Gets or sets the name of a file or directory for which write access is secured.

Property Value

A **System.String** containing the absolute path of the file or directory for which write access is secured.

Description

[*Note:* This property sets write access for a single file or directory; use additional **System.Security.Permissions.FileIOPermissionAttribute** attributes to specify additional files and directories.]

Exceptions

Exception	Condition
System.ArgumentException	The path information specified for a set operation contains invalid characters or wild card specifiers.