System.IO.File Class

[ILASM]

```
.class public sealed File extends System.Object
```

[C#]

```
public sealed class File
```

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

Provides information and performs operations on files.

Inherits From: System.Object

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

Implementations of this class are required to preserve the case of path strings. Implementations are required to be case sensitive if and only if the platform is case-sensitive.

The following table describes the enumerations that are used to customize the behavior of various **System.IO.File** methods.

Enumeration	Description
System.IO.FileAccess	Specifies read and write access to a file.
System.IO.FileShare	Specifies the level of access permitted for a file that is already in use.
System.IO.FileMode	Specifies whether the contents of an existing file are preserved or overwritten, and whether requests to create an existing file cause an exception.

File.AppendText(System.String) Method

```
[ILASM]
.method public hidebysig static class
System.IO.StreamWriter AppendText(string path)

[C#]
public static StreamWriter AppendText(string path)
```

Summary

Appends UTF-8 encoded text to an existing file.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to append to.

Return Value

A **System.IO.StreamWriter** that appends UTF-8 encoded text to the specified file.

Description

This method is equivalent to **System.IO.StreamWriter** (*path*, **true**). If the file specified by *path* does not exist, it is created. If the file does exist, writes to the **System.IO.StreamWriter** append text to the file. Additional threads are permitted to read the file while it is open.

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more invalid characters.
System.IO.FileNotFoundException	*path* was not found.
System.ArgumentNullException	*path* is **null**.

System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write the specified file. See **System.Security.Permissions.FileIOPermissionAccess Write**.

File.Copy(System.String, System.String) Method

```
[ILASM]
.method public hidebysig static void Copy(string
sourceFileName, string destFileName)

[C#]
public static void Copy(string sourceFileName, string
destFileName)
```

Summary

Copies the specified file to a new location.

Parameters

Parameter	Description
sourceFileName	A **System.String** containing the name of the file to copy.
destFileName	A **System.String** containing the name of the destination file. Cannot specify a directory or an existing file.

Description

This method is equivalent to **System.IO.File.Copy** (*sourceFileName*, *destFileName*, **false**).

The *sourceFileName* and *destFileName* arguments are permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.IO.IOException	*destFileName* exists. -or- An I/O error occurred.
System.ArgumentNullException	*sourceFileName* or *destFileName* is **null**.

System.ArgumentException	*sourceFileName* or *destFileName* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters. -or- *sourceFileName* or *destFileName* specifies a directory.
System.Security.SecurityException	The caller does not have the required permission.
System.IO.DirectoryNotFoundException	Directory information in *sourceFileName* or *destFileName* was not found.
System.IO.FileNotFoundException	*sourceFileName* was not found.
System.IO.PathTooLongException	The length or the absolute path information for *sourceFileName* or *destFileName* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read the source file and write the destination file. See **System.Security.Permissions.FileIOPermissionAcces** **Read** and **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.Copy(System.String, System.String, System.Boolean) Method

```
[ILASM]
.method public hidebysig static void Copy(string
sourceFileName, string destFileName, bool overwrite)

[C#]
public static void Copy(string sourceFileName, string
destFileName, bool overwrite)
```

Summary

Copies the specified file to a new location.

Parameters

Parameter	Description
sourceFileName	A **System.String** containing the name of the file to copy.
destFileName	A **System.String** containing the name of the destination file. Cannot specify the name of a directory.
overwrite	A **System.Boolean** value. Specify **true** if the destination file can be overwritten; otherwise **false**.

Description

The *sourceFileName* and *destFileName* arguments are permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.IO.IOException	*destFileName* is read-only (write-protected), or *destFileName* exists and *overwrite* is **false**. -or- An I/O error occurred.
System.ArgumentNullException	*sourceFileName* or *destFileName* is

	null.
System.ArgumentException	*sourceFileName* or *destFileName* is a zero-length string, contains only white space, or contains one or more invalid characters. -or- *sourceFileName* or *destFileName* specifies a directory.
System.Security.SecurityException	The caller does not have the required permission.
System.IO.DirectoryNotFoundException	Directory information in *destFileName* or *sourceFileName* was not found.
System.IO.FileNotFoundException	*sourceFileName* was not found.
System.IO.PathTooLongException	The length or the absolute path information for *sourceFileName* or *destFileName* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read the source file and write the destination file. See **System.Security.Permissions.FileIOPermissionAcces Read** and **System.Security.Permissions.FileIOPermissionAcces Write**.

File.Create(System.String) Method

```
[ILASM]
.method public hidebysig static class System.IO.FileStream
Create(string path)

[C#]
public static FileStream Create(string path)
```

Summary

Creates or overwrites the specified file.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file.

Return Value

A **System.IO.FileStream** that provides read/write access to the specified file.

Description

This method is equivalent to **System.IO.File.Create** (*path*, **System.IO.FileStream.DefaultBufferSize**).

If the specified file does not exist, it is created; if it does exist and it is not read-only, the contents are overwritten.

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid

	characters.
System.ArgumentNullException	*path* is **null**.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.IOException	An I/O error occurred while creating the file.
System.UnauthorizedAccessException	*path* specified a file that is read-only (write-protected).
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.Create(System.String, System.Int32) Method

```
[ILASM]
.method public hidebysig static class System.IO.FileStream
Create(string path, int32 bufferSize)

[C#]
public static FileStream Create(string path, int
bufferSize)
```

Summary

Creates or overwrites the specified file.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file.
bufferSize	A **System.Int32** containing the number of bytes buffered for reads and writes to the file.

Return Value

A **System.IO.FileStream** that provides read/write access to the specified file.

Description

This method is equivalent to **System.IO.FileStream** (*path*, **System.IO.FileMode.Create**, **System.IO.FileAccess.ReadWrite**, **System.IO.FileShare.None**, *bufferSize*).

If the specified file does not exist, it is created; if it does exist and it is not read-only, the contents are overwritten.

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition

System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	*path* is **null**.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.IOException	An I/O error occurred while creating the file.
System.UnauthorizedAccessException	*path* specified a file that is read-only (write-protected).
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.CreateText(System.String) Method

```
[ILASM]
.method public hidebysig static class
System.IO.StreamWriter CreateText(string path)

[C#]
public static StreamWriter CreateText(string path)
```

Summary

Creates or opens a new file for writing UTF-8 encoded text.

Parameters

Parameter	Description
path	The file to be opened for writing.

Return Value

A **System.IO.StreamWriter** that writes to the specified file using UTF-8 encoding.

Description

This method is equivalent to **System.IO.StreamWriter** (*path*, **false**). If the file specified by *path* does not exist, it is created. If the file does exist, its contents are overwritten. Additional threads are permitted to read the file while it is open.

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more invalid characters.
System.ArgumentNullException	*path* is **null**.

System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	**Description**
System.Security.Permissions.FileIOPermission	Requires permission to write the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.Delete(System.String) Method

```
[ILASM]
.method public hidebysig static void Delete(string path)

[C#]
public static void Delete(string path)
```

Summary

Deletes the specified file.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to be deleted.

Description

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	*path* is **null**.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.IOException	The specified file is in use.
System.Security.SecurityException	The caller does not have the required permission.
System.UnauthorizedAccessException	*path* identifies a directory.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write to the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.Exists(System.String) Method

```
[ILASM]
.method public hidebysig static bool Exists(string path)

[C#]
public static bool Exists(string path)
```

Summary

Returns a **System.Boolean** indicating whether the specified file exists.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to check.

Return Value

true if the caller has the required permissions and *path* contains the name of an existing file; otherwise, **false**. If *path* is **null** or a zero-length string, returns **false**.

Description

If the caller does not have sufficient permissions to read the specified file, no exception is thrown and the method returns **false** regardless of the existence of *path*.

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Read**.

File.GetCreationTime(System.String) Method

[ILASM]
```
.method public hidebysig static valuetype System.DateTime
GetCreationTime(string path)
```

[C#]
```
public static DateTime GetCreationTime(string path)
```

Summary

Returns the creation date and time of the specified file or directory.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file or directory for which to obtain creation date and time information.

Return Value

A **System.DateTime** structure set to the creation date and time for the specified file or directory. This value is expressed in local time.

Platforms that do not support this feature return **System.DateTime.MinValue**.

Description

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	*path* is **null**.
System.IO.IOException	*path* was not found.

System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.
System.Security.SecurityException	The caller does not have the required permission.

Permissions

Permission	**Description**
System.Security.Permissions.FileIOPermission	Requires permission to read the specified file or directory See **System.Security.Permissions.FileIOPermissionAcces** **Read**.

File.GetLastAccessTime(System.String) Method

```
[ILASM]
.method public hidebysig static valuetype System.DateTime
GetLastAccessTime(string path)

[C#]
public static DateTime GetLastAccessTime(string path)
```

Summary

Returns the date and time the specified file or directory was last accessed.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file or directory for which to obtain access date and time information.

Return Value

A **System.DateTime** structure set to the date and time the specified file or directory was last accessed. This value is expressed in local time.

Platforms that do not support this feature return **System.DateTime.MinValue**.

Description

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	*path* is **null**.

System.IO.IOException	*path* was not found.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.
System.Security.SecurityException	The caller does not have the required permission.

Permissions

Permission	**Description**
System.Security.Permissions.FileIOPermission	Requires permission to read the specified file or directory See **System.Security.Permissions.FileIOPermissionAcces** **Read**.

File.GetLastWriteTime(System.String) Method

```
[ILASM]
.method public hidebysig static valuetype System.DateTime
GetLastWriteTime(string path)

[C#]
public static DateTime GetLastWriteTime(string path)
```

Summary

Returns the date and time the specified file or directory was last written to.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file for which to obtain write date and time information.

Return Value

A **System.DateTime** structure set to the date and time the specified file or directory was last written to. This value is expressed in local time.

Platforms that do not support this feature return **System.DateTime.MinValue**.

Description

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	*path* is **null**.

System.IO.IOException	*path* was not found.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.
System.Security.SecurityException	The caller does not have the required permission.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read the specified file or directory See **System.Security.Permissions.FileIOPermissionAccess Read**.

File.Move(System.String, System.String) Method

[ILASM]

```
.method public hidebysig static void Move(string
sourceFileName, string destFileName)
```

[C#]

```
public static void Move(string sourceFileName, string
destFileName)
```

Summary

Moves the specified file to a new location.

Parameters

Parameter	Description
sourceFileName	A **System.String** containing the name of the file to move.
destFileName	A **System.String** containing the name of the new location for the file.

Description

This method does not throw an exception if the source and destination are the same.

The *sourceFileName* and *destFileName* arguments are permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.IO.IOException	The destination file is read-only or *destFileName* is a directory.
System.ArgumentNullException	*sourceFileName* or *destFileName* is **null**.
System.ArgumentException	*sourceFileName* or *destFileName* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid

	characters.
System.Security.SecurityException	The caller does not have the required permission.
System.IO.DirectoryNotFoundException	The directory information in *sourceFileName* or *destFileName* was not found.
System.IO.FileNotFoundException	*sourceFileName* was not found or specifies a directory.
System.IO.PathTooLongException	The length or absolute path information for *sourceFileName* or *destFileName* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read from *sourceFileName*, and write to *sourceFileName* and *destFileName*. See **System.Security.Permissions.FileIOPermissionAcces** **Read** and **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.Open(System.String, System.IO.FileMode) Method

```
[ILASM]
.method public hidebysig static class System.IO.FileStream
Open(string path, valuetype System.IO.FileMode mode)

[C#]
public static FileStream Open(string path, FileMode mode)
```

Summary

Opens a **System.IO.FileStream** on the specified file with read/write access.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to open.
mode	A **System.IO.FileMode** value that specifies whether a file is created if one does not exist, and determines whether the contents of existing files are retained or overwritten.

Return Value

A **System.IO.FileStream** that provides read/write access to the specified file.

Description

This method is equivalent to **System.IO.FileStream** (*path*, *mode*, **System.IO.FileAccess.ReadWrite**, **System.IO.FileShare.None**).

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.

System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.IO.FileNotFoundException	*path* was not found.
System.ArgumentNullException	*path* is **null**.
System.ArgumentOutOfRangeException	*mode* specified an invalid value.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.IOException	An I/O error occurred while opening the file.
System.UnauthorizedAccessException	*path* specified a read-only file (this method attempts to open the file with read/write access). -or- This operation is not supported on the current platform. -or- *path* specified a directory.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read and write the file. See **System.Security.Permissions.FileIOPermissionAcces Read** and **System.Security.Permissions.FileIOPermissionAcces Write**.

File.Open(System.String, System.IO.FileMode, System.IO.FileAccess) Method

[ILASM]
```
.method public hidebysig static class System.IO.FileStream
Open(string path, valuetype System.IO.FileMode mode,
valuetype System.IO.FileAccess access)
```

[C#]
```
public static FileStream Open(string path, FileMode mode,
FileAccess access)
```

Summary

Opens a **System.IO.FileStream** on the specified file.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to open.
mode	A **System.IO.FileMode** value that specifies whether a file is created if one does not exist, and determines whether the contents of existing files are retained or overwritten.
access	A **System.IO.FileAccess** value that specifies the operations that can be performed on the file.

Return Value

A **System.IO.FileStream** that provides access to the specified file.

Description

This method is equivalent to **System.IO.FileStream** (*path*, *mode*, *access*, **System.IO.FileShare.None**).

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters. -or- *access* specified **Read** and *mode* specified **Create**, **CreateNew**, **Truncate** or **Append**.
System.IO.FileNotFoundException	*path* was not found.
System.ArgumentNullException	*path* is **null**.
System.ArgumentOutOfRangeException	*mode* or *access* specified an invalid value.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.IOException	An I/O error occurred while opening the file.
System.UnauthorizedAccessException	*path* specified a read-only file and *access* is not **Read**, or *path* specified a directory.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read and may also require permission to write the file. See **System.Security.Permissions.FileIOPermissionAcces** **Read** and **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.Open(System.String, System.IO.FileMode, System.IO.FileAccess, System.IO.FileShare) Method

[ILASM]

```
.method public hidebysig static class System.IO.FileStream
Open(string path, valuetype System.IO.FileMode mode,
valuetype System.IO.FileAccess access, valuetype
System.IO.FileShare share)
```

[C#]

```
public static FileStream Open(string path, FileMode mode,
FileAccess access, FileShare share)
```

Summary

Opens a **System.IO.FileStream** on the specified file.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to open.
mode	A **System.IO.FileMode** value that specifies whether a file is created if one does not exist, and determines whether the contents of existing files are retained or overwritten.
access	A **System.IO.FileAccess** value that specifies the operations that can be performed on the file.
share	A **System.IO.FileShare** value specifying the type of access other threads have to the file.

Return Value

A **System.IO.FileStream** that provides access to the specified file.

Description

This method is equivalent to **System.IO.FileStream** (*path*, *mode*, *access*, *share*).

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters. -or- *access* specified **Read** and *mode* specified **Create**, **CreateNew**, **Truncate** or **Append**.
System.IO.FileNotFoundException	*path* was not found.
System.ArgumentNullException	*path* is **null**.
System.ArgumentOutOfRangeException	*mode*, *access*, or *share* specified an invalid value.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.IOException	An I/O error occurred while opening the file.
System.UnauthorizedAccessException	*path* specified a read-only file and *access* is not **Read**, or *path* specified a directory.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read and may also require permission to write the file. See **System.Security.Permissions.FileIOPermissionAcces** **Read** and **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.OpenRead(System.String) Method

```
[ILASM]
.method public hidebysig static class System.IO.FileStream
OpenRead(string path)

[C#]
public static FileStream OpenRead(string path)
```

Summary

Opens an existing file for reading.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to be opened for reading.

Return Value

A read-only **System.IO.FileStream** containing the contents of the specified file.

Description

This method is equivalent to **System.IO.FileStream** (*path*, **System.IO.FileMode.Open**, **System.IO.FileAccess.Read**, **System.IO.FileShare.Read**).

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.

System.IO.FileNotFoundException	*path* was not found.
System.ArgumentNullException	*path* is **null**.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.UnauthorizedAccessException	*path* specified a read-only (write-protected) file or a directory.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read the specified file. See **System.Security.Permissions.FileIOPermissionAcces Read**.

File.OpenText(System.String) Method

```
[ILASM]
.method public hidebysig static class
System.IO.StreamReader OpenText(string path)

[C#]
public static StreamReader OpenText(string path)
```

Summary

Opens an existing UTF-8 encoded text file for reading.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to be opened for reading.

Return Value

A **System.IO.StreamReader** containing the contents of the specified file.

Description

This method is equivalent to **System.IO.StreamReader** (*path*).

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.IO.FileNotFoundException	*path* was not found.
System.ArgumentNullException	*path* is **null**.

System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write to the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.OpenWrite(System.String) Method

[ILASM]
```
.method public hidebysig static class System.IO.FileStream
OpenWrite(string path)
```

[C#]
```
public static FileStream OpenWrite(string path)
```

Summary

Opens an existing file for writing.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file to be opened for writing.

Return Value

A writable **System.IO.FileStream** that writes to the file specified by *path*.

Description

This method is equivalent to **System.IO.FileStream** (*path*, **System.IO.FileMode.OpenOrCreate**, **System.IO.FileAccess.Write**, **System.IO.FileShare.None**).

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

Exceptions

Exception	Condition
System.Security.SecurityException	The caller does not have the required permission.
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.

System.IO.FileNotFoundException	*path* was not found.
System.ArgumentNullException	*path* is **null**.
System.IO.DirectoryNotFoundException	The directory information specified in *path* was not found.
System.UnauthorizedAccessException	*path* specified a read-only file or a directory.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.SetCreationTime(System.String, System.DateTime) Method

```
[ILASM]
.method public hidebysig static void SetCreationTime(string
path, valuetype System.DateTime creationTime)

[C#]
public static void SetCreationTime(string path, DateTime
creationTime)
```

Summary

Sets the creation date and time for the specified file.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file for which to set the creation date and time information.
creationTime	A **System.DateTime** containing the value to set for the creation date and time of *path*. This value is expressed in local time.

Description

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

On platforms that do not support this feature, this method has no effect. If this feature is supported, the range of dates that is valid for this operation is implementation-specific.

Exceptions

Exception	Condition
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentOutOfRangeException	*creationTime* specifies a value outside the range of date/times permitted for this operation.

System.ArgumentNullException	*path* is **null**.
System.IO.FileNotFoundException	*path* was not found.
System.IO.IOException	An I/O error occurred while performing the operation.
System.IO.PathTooLongException	The length of *path* or the absolute path information for *path* exceeds the system-defined maximum length.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write to the specified file or directory. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.SetLastAccessTime(System.String, System.DateTime) Method

```
[ILASM]
.method public hidebysig static void
SetLastAccessTime(string path, valuetype System.DateTime
lastAccessTime)

[C#]
public static void SetLastAccessTime(string path, DateTime
lastAccessTime)
```

Summary

Sets the date and time the specified file was last accessed.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file for which to set the access date and time information.
lastAccessTime	A **System.DateTime** containing the value to set for the access date and time of *path*. This value is expressed in local time.

Description

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

On platforms that do not support this feature, this method has no effect. If this feature is supported, the range of dates that is valid for this operation is implementation-specific.

Exceptions

Exception	Condition
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	*path* is **null**.
System.IO.IOException	*path* was not found.
System.IO.PathTooLongException	The length of *path* or the absolute path

	information for *path* exceeds the system-defined maximum length.
System.Security.SecurityException	The caller does not have the required permission.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write to the specified file. See **System.Security.Permissions.FileIOPermissionAcces** **Write**.

File.SetLastWriteTime(System.String, System.DateTime) Method

[ILASM]

```
.method public hidebysig static void
SetLastWriteTime(string path, valuetype System.DateTime
lastWriteTime)
```

[C#]

```
public static void SetLastWriteTime(string path, DateTime
lastWriteTime)
```

Summary

Sets the date and time a file was last written to.

Parameters

Parameter	Description
path	A **System.String** containing the name of the file for which to set the date and time information.
lastWriteTime	A **System.DateTime** containing the value to set for the last write date and time of *path*. This value is expressed in local time.

Description

The *path* argument is permitted to specify relative or absolute path information. Relative path information is interpreted as relative to the current working directory. [*Note:* To obtain the current working directory, see **System.IO.Directory.GetCurrentDirectory**.]

On platforms that do not support this feature, this method has no effect. If this feature is supported, the range of dates that is valid for this operation is implementation-specific.

Exceptions

Exception	Condition
System.ArgumentException	*path* is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	*path* is **null**.
System.IO.IOException	*path* was not found.
System.IO.PathTooLongException	The length of *path* or the absolute path

	information for *path* exceeds the system-defined maximum length.
System.Security.SecurityException	The caller does not have the required permission.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write to the specified file. See **System.Security.Permissions.FileIOPermissionAccess** **Write**.