

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]
- *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.

31

32

1 File.AppendText(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static class  
4 System.IO.StreamWriter AppendText(string path)  
5  
6 [C#]  
7 public static StreamWriter AppendText(string path)
```

7 Summary

8 Appends UTF-8 encoded text to an existing file.

9 Parameters

10
11

Parameter	Description
<i>path</i>	A System.String containing the name of the file to append to.

12
13
14

13 Return Value

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

17 Description

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

22
23
24
25
26

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**.]

27 Exceptions

28
29

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

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

1
2
3
4

Permissions

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

5
6
7

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

2 Method

```
3 [ILASM]  
4 .method public hidebysig static void Copy(string  
5 sourceFileName, string destFileName)  
  
6 [C#]  
7 public static void Copy(string sourceFileName, string  
8 destFileName)
```

9 Summary

10 Copies the specified file to a new location.

11 Parameters

12
13

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

14
15

Description

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

18
19
20
21
22
23

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**.]

24 Exceptions

25
26

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.Copy(System.String, System.String, 2 System.Boolean) Method

```
3 [ILASM]  
4 .method public hidebysig static void Copy(string  
5 sourceFileName, string destFileName, bool overwrite)  
  
6 [C#]  
7 public static void Copy(string sourceFileName, string  
8 destFileName, bool overwrite)
```

9 Summary

10 Copies the specified file to a new location.

11 Parameters

12
13

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

14
15

Description

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

21 Exceptions

22
23

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.Create(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static class System.IO.FileStream  
4 Create(string path)  
  
5 [C#]  
6 public static FileStream Create(string path)
```

7 Summary

8 Creates or overwrites the specified file.

9 Parameters

10
11

Parameter	Description
<i>path</i>	A System.String containing the name of the file.

12
13
14

13 Return Value

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

17 Description

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

20
21
22
23

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

24
25
26
27

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**.]

28 Exceptions

29
30

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

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

1
2
3
4

Permissions

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

5
6
7

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

2 Method

```
3 [ILASM]  
4 .method public hidebysig static class System.IO.FileStream  
5 Create(string path, int32 bufferSize)  
  
6 [C#]  
7 public static FileStream Create(string path, int  
8 bufferSize)
```

9 Summary

10 Creates or overwrites the specified file.

11 Parameters

12
13

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

14
15
16

15 Return Value

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

19 Description

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

23
24 If the specified file does not exist, it is created; if it does exist and it is
25 not read-only, the contents are overwritten.

26
27 The *path* argument is permitted to specify relative or absolute path
28 information. Relative path information is interpreted as relative to the
29 current working directory. [Note: To obtain the current working
30 directory, see **System.IO.Directory.GetCurrentDirectory**.]

31 Exceptions

32
33

Exception	Condition
-----------	-----------

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

1
2
3
4

Permissions

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

5
6
7

1 File.CreateText(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static class  
4 System.IO.StreamWriter CreateText(string path)  
  
5 [C#]  
6 public static StreamWriter CreateText(string path)
```

7 Summary

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

9 Parameters

10
11

Parameter	Description
<i>path</i>	The file to be opened for writing.

12
13
14

Return Value

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

17 Description

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

22
23
24
25
26

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**.

27 Exceptions

28
29

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.Delete(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static void Delete(string path)  
4 [C#]  
5 public static void Delete(string path)
```

6 Summary

7 Deletes the specified file.

8 Parameters

9
10

Parameter	Description
<i>path</i>	A System.String containing the name of the file to be deleted.

11
12

Description

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

17 Exceptions

18
19

Exception	Condition
System.ArgumentException	<i>path</i> is a zero-length string, contains only white space, or contains one or more implementation-defined invalid characters.
System.ArgumentNullException	<i>path</i> is null .
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> 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	<i>path</i> identifies a directory.
System.IO.PathTooLongException	The length of <i>path</i> or the absolute path information for <i>path</i> exceeds the system-defined maximum length.

20
21

Permissions

1
2

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write to the specified file. See System.Security.Permissions.FileIOPermissionAccessWrite .

3
4
5

1 File.Exists(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static bool Exists(string path)  
4 [C#]  
5 public static bool Exists(string path)
```

6 Summary

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

9 Parameters

10
11

Parameter	Description
<i>path</i>	A System.String containing the name of the file to check.

12
13
14

13 Return Value

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

18 Description

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

22

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

27 Permissions

28
29

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

30
31
32

1 File.GetCreationTime(System.String)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static valuetype System.DateTime  
5 GetCreationTime(string path)  
  
6 [C#]  
7 public static DateTime GetCreationTime(string path)
```

8 Summary

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

10 Parameters

11
12

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

13
14
15

Return Value

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

18
19 Platforms that do not support this feature return
20 **System.DateTime.MinValue**.

21 Description

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

26 Exceptions

27
28

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

1
2
3
4

Permissions

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

5
6
7

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

1 File.GetLastAccessTime(System.String)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static valuetype System.DateTime  
5 GetLastAccessTime(string path)  
  
6 [C#]  
7 public static DateTime GetLastAccessTime(string path)
```

8 Summary

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

11 Parameters

12
13

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

14
15
16

15 Return Value

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

20
21 Platforms that do not support this feature return
22 **System.DateTime.MinValue**.

23 Description

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

28 Exceptions

29
30

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.GetLastWriteTime(System.String)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static valuetype System.DateTime  
5 GetLastWriteTime(string path)  
  
6 [C#]  
7 public static DateTime GetLastWriteTime(string path)
```

8 Summary

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

11 Parameters

12
13

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

14
15
16

Return Value

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

20
21 Platforms that do not support this feature return
22 **System.DateTime.MinValue**.

23 Description

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

28 Exceptions

29
30

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

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

1
2
3
4

Permissions

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

5
6
7

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

2 Method

```
3 [ILASM]  
4 .method public hidebysig static void Move(string  
5 sourceFileName, string destFileName)  
  
6 [C#]  
7 public static void Move(string sourceFileName, string  
8 destFileName)
```

9 Summary

10 Moves the specified file to a new location.

11 Parameters

12
13

Parameter	Description
<i>sourceFileName</i>	A System.String containing the name of the file to move.
<i>destFileName</i>	A System.String containing the name of the new location for the file.

14
15

Description

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

18
19
20
21
22
23

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.**]

24 Exceptions

25
26

Exception	Condition
System.IO.IOException	The destination file is read-only or <i>destFileName</i> is a directory.
System.ArgumentNullException	<i>sourceFileName</i> or <i>destFileName</i> is null .
System.ArgumentException	<i>sourceFileName</i> or <i>destFileName</i> 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 <i>sourceFileName</i> or <i>destFileName</i> was not found.
System.IO.FileNotFoundException	<i>sourceFileName</i> was not found or specifies a directory.
System.IO.PathTooLongException	The length or absolute path information for <i>sourceFileName</i> or <i>destFileName</i> exceeds the system-defined maximum length.

1
2
3
4

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to read from <i>sourceFileName</i> , and write to <i>sourceFileName</i> and <i>destFileName</i> . See System.Security.Permissions.FileIOPermissionAccess.Read and System.Security.Permissions.FileIOPermissionAccess.Write .

5
6
7

1 File.Open(System.String, 2 System.IO.FileMode) Method

```
3 [ILASM]  
4 .method public hidebysig static class System.IO.FileStream  
5 Open(string path, valuetype System.IO.FileMode mode)  
  
6 [C#]  
7 public static FileStream Open(string path, FileMode mode)
```

8 Summary

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

11 Parameters

12
13

Parameter	Description
<i>path</i>	A System.String containing the name of the file to open.
<i>mode</i>	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.

14
15
16

15 Return Value

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

19 Description

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

22
23
24
25
26

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**.]

27 Exceptions

28
29

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.Open(System.String, 2 System.IO.FileMode, 3 System.IO.FileAccess) Method

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

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

11 Summary

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

13 Parameters

14
15

Parameter	Description
<i>path</i>	A System.String containing the name of the file to open.
<i>mode</i>	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.
<i>access</i>	A System.IO.FileAccess value that specifies the operations that can be performed on the file.

16
17
18

17 Return Value

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

20 Description

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

23
24
25
26
27

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.**]

28 Exceptions

29
30

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

1
2
3
4

Permissions

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

5
6
7

1 File.Open(System.String, 2 System.IO.FileMode, 3 System.IO.FileAccess, 4 System.IO.FileShare) Method

```
5 [ILASM]  
6 .method public hidebysig static class System.IO.FileStream  
7 Open(string path, valuetype System.IO.FileMode mode,  
8 valuetype System.IO.FileAccess access, valuetype  
9 System.IO.FileShare share)  
  
10 [C#]  
11 public static FileStream Open(string path, FileMode mode,  
12 FileAccess access, FileShare share)
```

13 Summary

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

15 Parameters

Parameter	Description
<i>path</i>	A System.String containing the name of the file to open.
<i>mode</i>	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.
<i>access</i>	A System.IO.FileAccess value that specifies the operations that can be performed on the file.
<i>share</i>	A System.IO.FileShare value specifying the type of access other threads have to the file.

18 Return Value

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

22 Description

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

25
26 The *path* argument is permitted to specify relative or absolute path
27 information. Relative path information is interpreted as relative to the
28 current working directory. [Note: To obtain the current working
29 directory, see **System.IO.Directory.GetCurrentDirectory**.]

1 Exceptions

2
3

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

4
5
6
7

Permissions

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

8
9
10

1 File.OpenRead(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static class System.IO.FileStream  
4 OpenRead(string path)  
  
5 [C#]  
6 public static FileStream OpenRead(string path)
```

7 Summary

8 Opens an existing file for reading.

9 Parameters

10
11

Parameter	Description
<i>path</i>	A System.String containing the name of the file to be opened for reading.

12
13
14

Return Value

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

17 Description

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

21
22 The *path* argument is permitted to specify relative or absolute path
23 information. Relative path information is interpreted as relative to the
24 current working directory. [Note: To obtain the current working
25 directory, see **System.IO.Directory.GetCurrentDirectory**.]

26 Exceptions

27
28

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.OpenText(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static class  
4 System.IO.StreamReader OpenText(string path)  
5  
6 [C#]  
7 public static StreamReader OpenText(string path)
```

7 Summary

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

9 Parameters

10
11

Parameter	Description
<i>path</i>	A System.String containing the name of the file to be opened for reading.

12
13
14

13 Return Value

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

17 Description

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

19
20
21
22
23

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**.]

24 Exceptions

25
26

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.OpenWrite(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static class System.IO.FileStream  
4 OpenWrite(string path)  
  
5 [C#]  
6 public static FileStream OpenWrite(string path)
```

7 Summary

8 Opens an existing file for writing.

9 Parameters

10
11

Parameter	Description
<i>path</i>	A System.String containing the name of the file to be opened for writing.

12
13
14

Return Value

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

17 Description

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

21
22 The *path* argument is permitted to specify relative or absolute path
23 information. Relative path information is interpreted as relative to the
24 current working directory. [*Note*: To obtain the current working
25 directory, see **System.IO.Directory.GetCurrentDirectory**.]

26 Exceptions

27
28

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.SetCreationTime(System.String, 2 System.DateTime) Method

```
3 [ILASM]  
4 .method public hidebysig static void SetCreationTime(string  
5 path, valuetype System.DateTime creationTime)  
  
6 [C#]  
7 public static void SetCreationTime(string path, DateTime  
8 creationTime)
```

9 Summary

10 Sets the creation date and time for the specified file.

11 Parameters

12
13

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

14

15 Description

16 The *path* argument is permitted to specify relative or absolute path
17 information. Relative path information is interpreted as relative to the
18 current working directory. [Note: To obtain the current working
19 directory, see **System.IO.Directory.GetCurrentDirectory**.]
20

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

24 Exceptions

25
26

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

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

1
2
3
4

Permissions

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

5
6
7

1 File.SetLastAccessTime(System.String, 2 System.DateTime) Method

```
3 [ILASM]  
4 .method public hidebysig static void  
5 SetLastAccessTime(string path, valuetype System.DateTime  
6 lastAccessTime)  
  
7 [C#]  
8 public static void SetLastAccessTime(string path, DateTime  
9 lastAccessTime)
```

10 Summary

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

12 Parameters

13
14

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

15
16

Description

17 The *path* argument is permitted to specify relative or absolute path
18 information. Relative path information is interpreted as relative to the
19 current working directory. [Note: To obtain the current working
20 directory, see **System.IO.Directory.GetCurrentDirectory**.]
21

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

25 Exceptions

26
27

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

1
2
3
4

Permissions

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

5
6
7

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

1 File.SetLastWriteTime(System.String, 2 System.DateTime) Method

```
3 [ILASM]  
4 .method public hidebysig static void  
5 SetLastWriteTime(string path, valuetype System.DateTime  
6 lastWriteTime)  
  
7 [C#]  
8 public static void SetLastWriteTime(string path, DateTime  
9 lastWriteTime)
```

10 Summary

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

12 Parameters

13
14

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

15
16

Description

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

21
22
23
24

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.

25 Exceptions

26
27

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

1
2
3
4

Permissions

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

5
6

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