Path.GetFullPath Method

Reference

Definition

Namespace: System.IO

Assembly: System.Runtime.dll

Overloads

Expand table

GetFullPath(String)	Returns the absolute path for the specified path string.
GetFullPath(String, String)	Returns an absolute path from a relative path and a fully qualified base path.

GetFullPath(String)

Source: Path.Unix.cs

Returns the absolute path for the specified path string.

C#

```
public static string GetFullPath (string path);
```

Parameters

`path` String

The file or directory for which to obtain absolute path information.

Returns

String

The fully qualified location of `path`, such as "C:\MyFile.txt".

Exceptions

ArgumentException

`path` is a zero-length string, contains only white space on Windows systems, or contains one or more of the invalid characters defined in GetInvalidPathChars().

-or-

The system could not retrieve the absolute path.

SecurityException

The caller does not have the required permissions.

ArgumentNullException

`path` is `null`.

NotSupportedException

.NET Framework only: `path` contains a colon (":") that is not part of a volume identifier (for example, "c:\").

PathTooLongException

The specified path, file name, or both exceed the system-defined maximum length.

Examples

The following example demonstrates the `GetFullPath` method on a Windows-based desktop platform.

C#

```
string fileName = "myfile.ext";
string path1 = @"mydir";
string path2 = @"\mydir";
string fullPath;

fullPath = Path.GetFullPath(path1);
Console.WriteLine("GetFullPath('{0}') returns '{1}'",
    path1, fullPath);

fullPath = Path.GetFullPath(fileName);
Console.WriteLine("GetFullPath('{0}') returns '{1}'",
    fileName, fullPath);

fullPath = Path.GetFullPath(path2);
Console.WriteLine("GetFullPath('{0}') returns '{1}'",
    path2, fullPath);

// Output is based on your current directory, except
```

```
// in the last case, where it is based on the root drive
// GetFullPath('mydir') returns 'C:\temp\Demo\mydir'
// GetFullPath('myfile.ext') returns 'C:\temp\Demo\myfile.ext'
// GetFullPath('\mydir') returns 'C:\mydir'
```

Remarks

The absolute path includes all information required to locate a file or directory on a system.

The file or directory specified by `path` is not required to exist. For example, if *c:\temp\newdir* is the current directory, calling `GetFullPath` on a file name such as *test.txt* returns *c:\temp\newdir\test.txt*. The file need not exist.

> **Important**
>
> If `path` is a relative path, this overload returns a fully qualified path that can be based on the current drive and current directory. The current drive and current directory can change at any time as an application executes. As a result, the path returned by this overload cannot be determined in advance. To return a deterministic path, call the GetFullPath(String, String) overload. You can also call the IsPathFullyQualified method to determine whether a path is fully qualified or relative and therefore whether a call to `GetFullPath` is necessary.

However, if `path` does exist, the caller must have permission to obtain path information for `path`. Note that unlike most members of the Path class, this method accesses the file system.

This method uses the current directory and current volume information to fully qualify `path`. If you specify a file name only in `path`, `GetFullPath` returns the fully qualified path of the current directory.

If you pass in a short file name, it is expanded to a long file name.

If a path contains no significant characters, it is invalid unless it contains one or more "." characters followed by any number of spaces; then it will be parsed as either "." or "..".

.NET Core 1.1 and later versions and .NET Framework 4.6.2 and later versions also support paths that include device names, such as "\?\C:\".

For more information on file path formats on Windows, see File path formats on Windows systems. For a list of common I/O tasks, see Common I/O Tasks.

See also

- File path formats on Windows systems
- File and Stream I/O
- How to: Read Text from a File
- How to: Write Text to a File

Applies to

▼ .NET 9 and other versions

Product	Versions
.NET	Core 1.0, Core 1.1, Core 2.0, Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7, 8, 9
.NET Framework	1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8, 4.8.1
.NET Standard	1.3, 1.4, 1.5, 1.6, 2.0, 2.1
UWP	10.0

GetFullPath(String, String)

Source: Path.Unix.cs

Returns an absolute path from a relative path and a fully qualified base path.

C#

```
public static string GetFullPath (string path, string basePath);
```

Parameters

`path` String

A relative path to concatenate to `basePath`.

`basePath` String

The beginning of a fully qualified path.

Returns

String

The absolute path.

Exceptions

ArgumentNullException

`path` or `basePath` is `null`.

ArgumentException

`basePath` is not a fully qualified path.

-or-

`path` or `basePath` contains one or more of the invalid characters defined in GetInvalidPathChars().

Examples

The following example defines a variable, `basePath`, to represent an application's current directory. It then passes it to the `GetFullPath` method to get a fully qualified path to the application's data directory.

C#

```
using System;
using System.IO;

class Program
{
    static void Main()
    {
        string basePath = Environment.CurrentDirectory;
        string relativePath = "./data/output.xml";

        // Unexpectedly change the current directory.
        Environment.CurrentDirectory =
"C:/Users/Public/Documents/";

        string fullPath = Path.GetFullPath(relativePath,
basePath);
        Console.WriteLine($"Current directory:\n   {Environmen-
t.CurrentDirectory}");
```

```
        Console.WriteLine($"Fully qualified path:\n
{fullPath}");
    }
}
// The example displays the following output:
//   Current directory:
//      C:\Users\Public\Documents
//   Fully qualified path:
//      C:\Utilities\data\output.xml
```

Remarks

If `path` is an empty path, the method returns `basePath`. If `path` is a fully qualified path, the method passes `path` to the GetFullPath(String) method and returns the result.

Use this method to return a deterministic path based on a specified volume and rooted directory when you're using relative paths. Using a predefined `basePath` rather than one based on the current drive directory guards against unwanted file paths caused by unexpected changes in the current drive and directory.

Applies to

▼ .NET 9 and other versions

Product	Versions
.NET	Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7, 8, 9
.NET Standard	2.1