

How to Delete File in C#? Examples

C# Delete File

To delete a file in C#, you can use `System.IO.File.Delete(path)` function. `File.Delete()` function deletes the file specified using the argument.

Example 1 – Delete File

In the following example, we shall delete a file with the path `D:\sample.txt`.

Program.cs

```
using System;
using System.IO;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            String path = @"D:\sample.txt";
            File.Delete(path);
            Console.WriteLine("Deleted file successfully.");
        }
    }
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
Deleted file successfully.
```

Note

If you try to delete a file that is not actually present using `File.Delete()`, no Exception is thrown or no error occurs.

Example 2 – Delete File with Path specified as null

If you specify path as `null`, `System.ArgumentNullException` is thrown by `File.Delete()` function.

Program.cs

```
using System;
using System.IO;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            String path = null;
            File.Delete(path);
            Console.WriteLine("Deleted file successfully.");
        }
    }
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run

Unhandled Exception: System.ArgumentNullException: Value cannot be null.
Parameter name: path
   at System.IO.File.Delete(String path)
   at CSharpExamples.Program.Main(String[] args) in D:\workspace\csharp\HelloWorld\Program.cs:line 12
```

Example 3 – Delete File – Path Containing Invalid Path Characters

The path should not contain invalid path characters. Invalid path characters' numeric range is [0, 31] and 124 . If path contains any of these invalid path characters, `System.IO.IOException` is thrown by `File.Delete()` function.

Program.cs

```
using System;
using System.IO;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            String path = "D:/sample1|.txt";
            File.Delete(path);
            Console.WriteLine("Deleted file successfully.");
        }
    }
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run

Unhandled Exception: System.IO.IOException: The filename, directory name, or volume label
```

```
Unhandled Exception: System.IO.IOException: The filename, directory name, or volume label
at System.IO.FileSystem.DeleteFile(String fullPath)
at System.IO.File.Delete(String path)
at CSharpExamples.Program.Main(String[] args) in D:\workspace\csharp\HelloWorld\Progr
```

Summary

In this [C# Tutorial](#), we have learned to delete a file with the path specified.

C# Tutorial

- ◆ [C# Tutorial](#)
- ◆ [C# Basic Example](#)
- ◆ [C# Comments](#)
- ◆ [C# Variables](#)
- ◆ [C# Constants](#)
- ◆ [C# if, if-else](#)
- ◆ [C# switch](#)
- ◆ [C# while loop](#)
- ◆ [C# for loop](#)
- ◆ [C# foreach](#)
- ◆ [C# break](#)
- ◆ [C# continue](#)
- ◆ [C# struct](#)
- ◆ [C# enum](#)
- ◆ [C# String](#)
- ◆ [C# Array](#)
- ◆ [C# Command Line Arguments](#)

C# Console Operations

- ◆ [C# Write to Console](#)
- ◆ [C# Read from Console](#)

C# Object Oriented Programming Concepts

◆ C# Class & Constructors

◆ C# Encapsulation

◆ C# Polymorphism

◆ C# Method Overloading

◆ C# Interfaces

C# String Operations

◆ C# String Length

◆ C# Substring

C# File Operations

◆ C# Read Text File

◆ C# Write to File

⇒ C# Delete File

◆ C# Copy File

C# Exception Handling

◆ C# try-catch

◆ C# finally

◆ C# throw

◆ C# Custom Exception

◆ C# SystemException

◆ C# DivideByZeroException

◆ C# NullReferenceException

◆ C# InvalidCastException

◆ C# IOException

◆ C# FieldAccessException

C# Collections

◆ C# List

◆ C# SortedList

◆ C# HashSet

◆ C# SortedSet

◆ C# Stack

C# Stack

- ◆ C# Queue
- ◆ C# LinkedList
- ◆ C# Dictionary
- ◆ C# SortedDictionary

C# Errors [Solved]

- ◆ C# Error: Class does not contain a constructor that takes arguments