

C# Write to Text File

C# Write to File

To write text to file in C#, you can use `System.IO.File.WriteAllLines()` function or `System.IO.File.WriteAllText()` function.

Or you can use `WriteLine` in recursive loop to write lines to a file. It gives more control on each line to be written.

Example 1 – Write to Text File – `System.IO.File.WriteAllLines()`

`System.IO.File.WriteAllLines()` writes strings in an array, to a text file.

Program.cs

```
using System;

namespace CSharpExamples {

    class Program {
        static void Main(string[] args) {
            string[] lines = { "Hello World.", "Welcome to C# Tutorial.", "TutorialKart." };
            try{
                System.IO.File.WriteAllLines(@"D:\lines.txt", lines);
                Console.WriteLine("Lines written to file successfully.");
            } catch(Exception err) {
                Console.WriteLine(err.Message);
            }
        }
    }
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
Lines written to file successfully.
```

D:\lines.txt

```
Hello World.
Welcome to C# Tutorial.
```

Example 2 – Write to Text File – System.IO.File.WriteAllText()

System.IO.File.WriteAllText() writes string to a text file specified by the argument passed to the function.

Program.cs

```
using System;

namespace CSharpExamples {

    class Program {
        static void Main(string[] args) {
            string str = "Hello World. Welcome to C# Tutorial. TutorialKart.";
            try{
                System.IO.File.WriteAllText(@"D:\sample.txt", str);
                Console.WriteLine("String written to file successfully.");
            } catch(Exception err) {
                Console.WriteLine(err.Message);
            }
        }
    }
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
String written to file successfully.
```

D:\sample.txt

```
Hello World. Welcome to C# Tutorial. TutorialKart.
```

Example 3 – Write to Text File – WriteLine in Loop

In the following example, we get StreamWriter for a file and then write each line in the string array to the file.

Program.cs

```
using System;

namespace CSharpExamples {

    class Program {
        static void Main(string[] args) {
            string[] lines = { "Hello World.", "Welcome to C# Tutorial.", "TutorialKart." };
            try{
                using (System.IO.StreamWriter file =
                    new System.IO.StreamWriter(@"D:\lines.txt")) {
```

```
        new System.IO.StreamWriter(@"D:\lines.txt") {
        foreach (string line in lines) {
            file.WriteLine(line);
        }
    }
    Console.WriteLine("Lines written to file successfully.");
} catch (Exception err) {
    Console.WriteLine(err.Message);
}
}
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
Lines written to file successfully.
```

D:\lines.txt

```
Hello World.
Welcome to C# Tutorial.
TutorialKart.
```

Conclusion

In this [C# Tutorial](#), we learned to write string to a text file, or lines stored in a string array to a text file.

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# Queue

◆ C# LinkedList

◆ C# Dictionary

◆ C# SortedDictionary

C# Errors [Solved]

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