# 1.创建文件或者文件夹

## 1>创建文件夹,

### 在创建之前可以使用System.IO.Path.Combine(pathString, "SubSubFolder");进行路径拼接

### 然后创建文件夹

|  |
| --- |
| System.IO.Directory.CreateDirectory(pathString); |

## 2.创建文件

### 1.>创建随机文件:

|  |
| --- |
| string fileName = System.IO.Path.GetRandomFileName(); |

### 2>创建自定义文件

|  |
| --- |
| string fileName = "MyNewFile.txt";  // Use Combine to add the file name to the path.  pathString = System.IO.Path.Combine(pathString, fileName);  // Verify the path that you have constructed.可以测试一下对不对  Console.WriteLine("Path to my file: {0}\n", pathString);  if (!System.IO.File.Exists(pathString))  {  using (System.IO.FileStream fs = System.IO.File.Create(pathString))  {  for (byte i = 0; i < 100; i++)  {  fs.WriteByte(i);  }  }  }  else  {  Console.WriteLine("File \"{0}\" already exists.", fileName);  return;  }  // Read and display the data from your file.  try  {  byte[] readBuffer = System.IO.File.ReadAllBytes(pathString);  foreach (byte b in readBuffer)  {  Console.Write(b + " ");  }  Console.WriteLine();  }  catch (System.IO.IOException e)  {  Console.WriteLine(e.Message);  }  // Keep the console window open in debug mode.  System.Console.WriteLine("Press any key to exit.");  System.Console.ReadKey();  } |

如果该文件夹已存在，则 [CreateDirectory](https://msdn.microsoft.com/zh-cn/library/system.io.directory.createdirectory.aspx) 不执行任何操作，且不会引发异常。但是，[File.Create](https://msdn.microsoft.com/zh-cn/library/system.io.file.create.aspx) 用新的文件替换现有文件。该示例使用一个 **if**-**else** 语句阻止现有文件被替换。

通过在示例中做出以下更改，您可以根据具有某个名称的程序是否存在来指定不同的结果。如果该文件不存在，代码将创建一个文件。如果该文件存在，代码将把数据添加到该文件中。

* 指定一个非随机文件名。

**// Comment out the following line.**

**//string fileName = System.IO.Path.GetRandomFileName();**

**// Replace that line with the following assignment.**

运行该示例若干次以验证数据是否每次都添加到文件中。

有关更多可以尝试的 **FileMode** 值的信息，请参阅 [FileMode](https://msdn.microsoft.com/zh-cn/library/system.io.filemode.aspx)。

以下情况可能会导致异常：

* 文件夹名称格式不正确。例如，它包含非法字符或仅仅是空白（[ArgumentException](https://msdn.microsoft.com/zh-cn/library/system.argumentexception.aspx) 类）。使用 [Path](https://msdn.microsoft.com/zh-cn/library/system.io.path.aspx) 类创建有效路径名。
* 要创建的文件夹的父文件夹是只读的（[IOException](https://msdn.microsoft.com/zh-cn/library/system.io.ioexception.aspx) 类）。
* 文件夹名称是 **null**（[ArgumentNullException](https://msdn.microsoft.com/zh-cn/library/system.argumentnullexception.aspx) 类）。
* 文件夹名称太长（[PathTooLongException](https://msdn.microsoft.com/zh-cn/library/system.io.pathtoolongexception.aspx) 类）。
* 文件夹名称只是冒号“:”（[PathTooLongException](https://msdn.microsoft.com/zh-cn/library/system.io.pathtoolongexception.aspx) 类）。

## .NET Framework 安全性

在部分信任的情况下可能会引发 [SecurityException](https://msdn.microsoft.com/zh-cn/library/system.security.securityexception.aspx) 类的实例。

如果没有创建文件夹的权限，则该示例引发 [UnauthorizedAccessException](https://msdn.microsoft.com/zh-cn/library/system.unauthorizedaccessexception.aspx) 类的实例。

## 请参阅

[System.IO](https://msdn.microsoft.com/zh-cn/library/system.io.aspx)

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)

[文件系统和注册表（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/2kzb96fk.aspx)

# 2. 写入文本文件

以下示例给出了将文本写入文件的各种方法。 前两个示例对 [System.IO.File](https://msdn.microsoft.com/zh-cn/library/system.io.file.aspx) 类使用静态便捷方法以将任何 IEnumerable<string> 的每个元素和一个字符串写入文本文件。 示例 3 展示了在写入文件时必须分别处理文本的每一行时，如何将文本添加到文件。 示例 1-3 覆盖文件中的所有现有内容，但示例 4 展示如何将文本追加到现有文件。

这些示例均会将字符串文本写入文件，但是你更有可能需要使用 [Format](https://msdn.microsoft.com/zh-cn/library/system.string.format.aspx) 方法，此方法具有很多用于写入不同类型值的控件，包括在字段中左右对齐、有无边距等。 还可以使用 C# [字符串内插](https://msdn.microsoft.com/zh-cn/library/dn961160.aspx)功能。

## 示例

class WriteTextFile

{

static void Main()

{

// These examples assume a "C:\Users\Public\TestFolder" folder on your machine.

// You can modify the path if necessary.

// Example #1: Write an array of strings to a file.

// Create a string array that consists of three lines.

string[] lines = { "First line", "Second line", "Third line" };

// WriteAllLines creates a file, writes a collection of strings to the file,

// and then closes the file. You do NOT need to call Flush() or Close().

## //方式1.File类直接操作

### System.IO.File.WriteAllLines(@"C:\Users\Public\TestFolder\WriteLines.txt", lines);

// Example #2: Write one string to a text file.

string text = "A class is the most powerful data type in C#. Like a structure, " +

"a class defines the data and behavior of the data type. ";

// WriteAllText creates a file, writes the specified string to the file,

// and then closes the file. You do NOT need to call Flush() or Close().

### System.IO.File.WriteAllText(@"C:\Users\Public\TestFolder\WriteText.txt", text);

// Example #3: Write only some strings in an array to a file.

// The using statement automatically flushes AND CLOSES the stream and calls

// IDisposable.Dispose on the stream object.

// NOTE: do not use FileStream for text files because it writes bytes, but StreamWriter

// encodes the output as text.

## //方式2.使用StreamWriter

using (System.IO.StreamWriter file =

new System.IO.StreamWriter(@"C:\Users\Public\TestFolder\WriteLines2.txt"))

{

foreach (string line in lines)

{

// If the line doesn't contain the word 'Second', write the line to the file.

if (!line.Contains("Second"))

{

file.WriteLine(line);

}

}

}

// Example #4: Append new text to an existing file.

// The using statement automatically flushes AND CLOSES the stream and calls

// IDisposable.Dispose on the stream object.

using (System.IO.StreamWriter file =

new System.IO.StreamWriter(@"C:\Users\Public\TestFolder\WriteLines2.txt", true))

{

file.WriteLine("Fourth line");

}

}

}

//Output (to WriteLines.txt):

// First line

// Second line

// Third line

//Output (to WriteText.txt):

// A class is the most powerful data type in C#. Like a structure, a class defines the data and behavior of the data type.

//Output to WriteLines2.txt after Example #3:

// First line

// Third line

//Output to WriteLines2.txt after Example #4:

// First line

// Third line

// Fourth line

这些示例均会将字符串文本写入文件，但是你更有可能需要使用 [Format](https://msdn.microsoft.com/zh-cn/library/system.string.format.aspx) 方法，此方法具有很多用于写入不同类型值的控件，包括在字段中左右对齐、有无边距等。 还可以使用 C# [字符串内插](https://msdn.microsoft.com/zh-cn/library/dn961160.aspx)功能。

## //fs3.使用FileStream,按字节写入

**string fileName = "MyNewFile.txt";**

**用以下代码中的 using 语句替换 if-else 语句。**

**using (System.IO.FileStream fs = new System.IO.FileStream(pathString, FileMode.Append))**

**{**

**for (byte i = 0; i <100; i++)**

**{**

**fs.WriteByte(i);**

**}**

**}**

## 可靠编程

以下情况可能会导致异常：

* 文件已存在并且为只读。
* 路径名可能太长。
* 磁盘可能已满。

## 请参见

#### 概念

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)

#### 其他资源

[文件系统和注册表（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/2kzb96fk.aspx) [示例：将集合保存到应用程序存储](http://code.msdn.microsoft.com/CSWinStoreAppSaveCollection-bed5d6e6)

## 写入文件的小例子

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.IO;  namespace csharpbasic\_file\_and\_reg  {  internal class Program  {  static void WriteFileDemo1()  {  string[] lines = { "恭喜发财,红包拿来", "但愿人长久,千里共婵娟","好好学习,天天向上" };  DirectoryInfo pathinfo = Directory.CreateDirectory(@"e:\filetest");  string filename = Path.Combine(pathinfo.FullName, "test.txt");  if (!File.Exists(filename))  {  File.WriteAllLines(filename, lines);//c#可以直接创建文件并且写入数据.  }  }  static void WriteFileDemo2()  {  string[] lines = { "你好我好大家好", "恭喜发财,万事如意", "耶耶耶" };  DirectoryInfo pathinfo = Directory.CreateDirectory(@"e:\filetest");  string filename = Path.Combine(pathinfo.FullName, "test.txt");  using (StreamWriter sw = new StreamWriter(filename,true))//true表示一追加的方式写入.  {  foreach (var line in lines)  {  sw.WriteLine(line);  }    }  }  static void WriteFileDemo3()//使用这种方法需要在每一行后加换行符  {  string[] lines = { "床前明月光\n", "疑是地上霜\n", "No No No\n" };  DirectoryInfo pathinfo = Directory.CreateDirectory(@"e:\filetest");  string filename = Path.Combine(pathinfo.FullName, "test.txt");  using (FileStream fs = new FileStream(filename,FileMode.Append))//.  {  foreach (var line in lines)  {  byte[] info = new UTF8Encoding(true).GetBytes(line);//c#字符串转化为字节的方法  fs.Write(info, 0, info.Length);  }  }  }  static void Main(string[] args)  {  //WriteFileDemo1();  //WriteFileDemo2();  WriteFileDemo3();  }  }  } |

# 3.读取文件内容

## 1.使用File类

|  |
| --- |
| class ReadFromFile  {  static void Main()  {  // The files used in this example are created in the topic  // How to: Write to a Text File. You can change the path and  // file name to substitute text files of your own.  // Example #1  // Read the file as one string.  string text = System.IO.File.ReadAllText(@"C:\Users\Public\TestFolder\WriteText.txt");  // Display the file contents to the console. Variable text is a string.  System.Console.WriteLine("Contents of WriteText.txt = {0}", text);  // Example #2  // Read each line of the file into a string array. Each element  // of the array is one line of the file.  string[] lines = System.IO.File.ReadAllLines(@"C:\Users\Public\TestFolder\WriteLines2.txt");  // Display the file contents by using a foreach loop.  System.Console.WriteLine("Contents of WriteLines2.txt = ");  foreach (string line in lines)  {  // Use a tab to indent each line of the file.  Console.WriteLine("\t" + line);  }  // Keep the console window open in debug mode.  Console.WriteLine("Press any key to exit.");  System.Console.ReadKey();  }  } |

## 2.使用StreamReader类

|  |
| --- |
| static void ReadFileDemoSR()  {  try  {  // Create an instance of StreamReader to read from a file.  // The using statement also closes the StreamReader.  using (StreamReader sr = new StreamReader(Path.Combine(@"e:\filetest", "test.txt")))  {  string line;  Console.WriteLine("File Content:\n=========================");  while ((line = sr.ReadLine()) != null)  {  Console.WriteLine(line);  }  Console.WriteLine("==================End=================");  }  }  catch (Exception e)  {  // Let the user know what went wrong.  Console.WriteLine("The file could not be read:");  Console.WriteLine(e.Message);  }    } |

### StreamReader还可以异步读取文件

|  |
| --- |
| using System;  using System.IO;  using System.Threading.Tasks;  class Example  {  static async Task Main()  {  await ReadAndDisplayFilesAsync();  }  static async Task ReadAndDisplayFilesAsync()  {  String filename = "TestFile1.txt";  Char[] buffer;  using (var sr = new StreamReader(filename)) {  buffer = new Char[(int)sr.BaseStream.Length];  await sr.ReadAsync(buffer, 0, (int)sr.BaseStream.Length);  }  Console.WriteLine(new String(buffer));  }  }  // The example displays the following output:  // This is the first line of text in a relatively short file.  // This is the second line.  // This is the third line.  // This is the fourth and final line. |

## 3.使用FileStream,这个比较麻烦一点

|  |
| --- |
| static void ReadFileDemoFS()  {  using (FileStream fs =  new FileStream(Path.Combine(@"e:\filetest", "test.txt"), FileMode.Open, FileAccess.Read))  {  byte[] bytes = new byte[fs.Length];  int bytesToRead = (int)(fs.Length);  int bytesRead = 0;  Console.WriteLine("File Content:\n=========================");  while (bytesToRead>0)  {  int n = fs.Read(bytes, bytesRead, bytesToRead);  Console.WriteLine(new UTF8Encoding(true).GetString(bytes));//c#把byte数组转化为字符串  if (n == 0)//读取不到就跳出循环  break;  bytesRead += n;  bytesToRead -= n;  }  Console.WriteLine("==================End=================");  }  } |

## 比较完整的小例子

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.IO;  using static System.Net.Mime.MediaTypeNames;  namespace csharpbasic\_file\_and\_reg  {  internal class Program  {  static void WriteFileDemo1()  {  string[] lines = { "恭喜发财,红包拿来", "但愿人长久,千里共婵娟","好好学习,天天向上" };  DirectoryInfo pathinfo = Directory.CreateDirectory(@"e:\filetest");  string filename = Path.Combine(pathinfo.FullName, "test.txt");  if (!File.Exists(filename))  {  File.WriteAllLines(filename, lines);//c#可以直接创建文件并且写入数据.  }  }  static void WriteFileDemo2()  {  string[] lines = { "你好我好大家好", "恭喜发财,万事如意", "耶耶耶" };  DirectoryInfo pathinfo = Directory.CreateDirectory(@"e:\filetest");  string filename = Path.Combine(pathinfo.FullName, "test.txt");  using (StreamWriter sw = new StreamWriter(filename,true))//true表示一追加的方式写入.  {  foreach (var line in lines)  {  sw.WriteLine(line);  }    }  }  static void WriteFileDemo3()//使用这种方法需要在每一行后加换行符  {  string[] lines = { "床前明月光\n", "疑是地上霜\n", "No No No\n" };  DirectoryInfo pathinfo = Directory.CreateDirectory(@"e:\filetest");  string filename = Path.Combine(pathinfo.FullName, "test.txt");  using (FileStream fs = new FileStream(filename,FileMode.Append))//.  {  foreach (var line in lines)  {  byte[] info = new UTF8Encoding(true).GetBytes(line);//c#字符串转化为字节的方法  fs.Write(info, 0, info.Length);  }  }  }  static void ReadFileDemo1() //使用File类  {  string text = File.ReadAllText(Path.Combine(@"e:\filetest","test.txt"));  Console.WriteLine($"File Content:\n============================\n{text}");  }  static void ReadFileDemo2()  {  string[] lines = File.ReadAllLines(Path.Combine(@"e:\filetest", "test.txt"));  Console.WriteLine("File Content:\n============================");  foreach (var line in lines)  {  Console.WriteLine(line);  }  Console.WriteLine("===============End=============\n");  }  static void ReadFileDemoSR()  {  try  {  // Create an instance of StreamReader to read from a file.  // The using statement also closes the StreamReader.  using (StreamReader sr = new StreamReader(Path.Combine(@"e:\filetest", "test.txt")))  {  string line;  Console.WriteLine("File Content:\n=========================");  while ((line = sr.ReadLine()) != null)  {  Console.WriteLine(line);  }  Console.WriteLine("==================End=================");  }  }  catch (Exception e)  {  // Let the user know what went wrong.  Console.WriteLine("The file could not be read:");  Console.WriteLine(e.Message);  }    }  static void ReadFileDemoFS()  {  using (FileStream fs =  new FileStream(Path.Combine(@"e:\filetest", "test.txt"), FileMode.Open, FileAccess.Read))  {  byte[] bytes = new byte[fs.Length];  int bytesToRead = (int)(fs.Length);  int bytesRead = 0;  Console.WriteLine("File Content:\n=========================");  while (bytesToRead>0)  {  int n = fs.Read(bytes, bytesRead, bytesToRead);  Console.WriteLine(new UTF8Encoding(true).GetString(bytes));//c#把byte数组转化为字符串  if (n == 0)//读取不到就跳出循环  break;  bytesRead += n;  bytesToRead -= n;  }  Console.WriteLine("==================End=================");  }  }  static void Main(string[] args)  {  //WriteFileDemo1();  //WriteFileDemo2();  //WriteFileDemo3();  // ReadFileDemo1();  //ReadFileDemo2();  //ReadFileDemoSR();  ReadFileDemoFS();  }  }  } |