# 文件复制移动删除需要用到的类

以下示例说明如何使用 [System.IO](https://msdn.microsoft.com/zh-cn/library/system.io.aspx) 命名空间中的 [System.IO.File](https://msdn.microsoft.com/zh-cn/library/system.io.file.aspx)、[System.IO.Directory](https://msdn.microsoft.com/zh-cn/library/system.io.directory.aspx)、[System.IO.FileInfo](https://msdn.microsoft.com/zh-cn/library/system.io.fileinfo.aspx) 和 [System.IO.DirectoryInfo](https://msdn.microsoft.com/zh-cn/library/system.io.directoryinfo.aspx) 类以同步方式复制、移动和删除文件和文件夹。 这些示例没有提供进度栏或其他任何用户界面。 如果您想提供一个标准进度对话框，请参见[如何：提供文件操作进度对话框（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/cc165446.aspx)。

在操作多个文件时，请使用 [System.IO.FileSystemWatcher](https://msdn.microsoft.com/zh-cn/library/system.io.filesystemwatcher.aspx) 提供一些事件，以便可以利用这些事件计算进度。 另一种方法是使用平台调用来调用 Windows Shell 中相应的文件相关方法。 有关如何异步执行这些文件操作的信息，请参见[异步文件 I/O](https://msdn.microsoft.com/zh-cn/library/kztecsys.aspx)。

# 1.递归复制整个文件夹

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DirCopy  {  internal class Program  {  static void DirCopy(string dstPath,string srcPath)  {  string fileName = null;  if (!Directory.Exists(dstPath)) //如果目标文件夹不存在,就创建  {  Directory.CreateDirectory(dstPath);  }  if (Directory.Exists(srcPath))  {  string[] files = Directory.GetFiles(srcPath);  foreach (string file in files)  {  fileName = Path.GetFileName(file);  fileName = Path.Combine(dstPath, fileName);  File.Copy(file, fileName,true);  }  string[] dirs = Directory.GetDirectories(srcPath);  foreach (var item in dirs)  {  string dstSubdir = Path.GetFileName(item);  dstSubdir = Path.Combine(dstPath, dstSubdir);  if(!Directory.Exists(dstSubdir))  {  Directory.CreateDirectory(dstSubdir);  }  DirCopy(dstSubdir, item);  }  }  else  {  Console.WriteLine("Source path does not exist!");  return;  }  Console.WriteLine("Copy Finished...");  }  static void Main(string[] args)  {  //string src = @"e:\123";  //string dst =  DirCopy("e:\\123cpy", "e:\\123");  }  }  } |

## 这个是我在网上的实例的基础上完善的功能,可以递归复制

# 2.删除文件夹

## c#默认只能够删除可写而且是空的文件夹,我们需要递归删除所有文件,然后才能删除文件夹.如果需要删除非空文件夹,只需要把第二个参数设置为true即可如: Directory.Delete(@"e:\123cpy2",true);

# 3.移动整个文件夹

## 其实移动可以怎么做:先把源文件夹的内容全部复制到目标文件夹,然后删除源文件夹,我们已经实现看复制文件夹功能,实现起来非常方便

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DirCopy  {  internal class Program  {  static void DirCopy(string dstPath,string srcPath)  {  string fileName = null;  if (!Directory.Exists(dstPath)) //如果目标文件夹不存在,就创建  {  Directory.CreateDirectory(dstPath);  }  if (Directory.Exists(srcPath))  {  string[] files = Directory.GetFiles(srcPath);  foreach (string file in files)  {  fileName = Path.GetFileName(file);  fileName = Path.Combine(dstPath, fileName);  File.Copy(file, fileName,true);  }  string[] dirs = Directory.GetDirectories(srcPath);  foreach (var item in dirs)  {  string dstSubdir = Path.GetFileName(item);  dstSubdir = Path.Combine(dstPath, dstSubdir);  if(!Directory.Exists(dstSubdir))  {  Directory.CreateDirectory(dstSubdir);  }  DirCopy(dstSubdir, item);  }  }  else  {  Console.WriteLine("Source path does not exist!");  return;  }    }  static void DirMove(string dstPath, string srcPath)  {  //1.先复制内容到目标文件夹  DirCopy(dstPath, srcPath);  //2.然后递归删除源文件夹  Directory.Delete(srcPath, true);  }  static void Main(string[] args)  {  //DirCopy("e:\\images", "d:\\images");  // Directory.Delete(@"e:\123cpy2",true);  DirMove("e:\\pictures", "e:\\images");  }  }  } |

# 任务

# [如何：提供文件操作进度对话框（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/cc165446.aspx)

# 参考

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

# 概念

# [C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx) [通用 I/O 任务](https://msdn.microsoft.com/zh-cn/library/ms404278.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) [文件和流 I/O](https://msdn.microsoft.com/zh-cn/library/k3352a4t.aspx)