# Directory

## 创建目录

// 摘要:按 path 的指定创建所有目录和子目录。

public static DirectoryInfo CreateDirectory(string path);

// 摘要:创建指定路径中的所有目录，并应用指定的 Windows 安全性。

public static DirectoryInfo CreateDirectory(string path, DirectorySecurity directorySecurity);

　　这里要说明一下，我们使用CreateDirectory方法时，不是只能创建一级目录，而是可以创建一个层级目录，只要你指定的目录路径无此目录，就可以创建。

## 删除目录

// 摘要:从指定路径删除空目录。

public static void Delete(string path);

// 摘要:删除指定的目录并（如果指示）删除该目录中的任何子目录。

public static void Delete(string path, bool recursive);

　　我们看方法的注释可以知道，第一个方法只能删除一个空的目录，若删除非空目录会提示错误，如图：



第二个方法则是可以通过指定第二个参数来控制是否递归删除子目录，因此，建议在使用时先使用第一个方法，当捕获到异常时，再提示使用第二个方法递归删除（此处只是个人建议，并非固定模式）。

## 获取子目录、文件

//获取目录集合<br>// 摘要:返回指定路径中的目录名称的可枚举集合。

public static IEnumerable<string>EnumerateDirectories(string path) // 摘要:返回指定路径中与搜索模式匹配的目录名称的可枚举集合。

public static IEnumerable<string>EnumerateDirectories(string path, string searchPattern);

// 摘要:返回指定路径中与搜索模式匹配的目录名称的可枚举集合，还可以搜索子目录。

public static IEnumerable<string> EnumerateDirectories(string path, string searchPattern, SearchOption searchOption);

// 摘要:返回指定路径中的文件名的可枚举集合。

public static IEnumerable<string> EnumerateFiles(string path);

// 摘要:返回指定路径中与搜索模式匹配的文件名称的可枚举集合。

public static IEnumerable<string> EnumerateFiles(string path, string searchPattern);

// 摘要:返回指定路径中与搜索模式匹配的文件名称的可枚举集合，还可以搜索子目录。

public static IEnumerable<string> EnumerateFiles(string path, string searchPattern, SearchOption searchOption);

// 摘要:返回指定路径中的文件系统项的可枚举集合。

public static IEnumerable<string>EnumerateFileSystemEntries(string path);

// 摘要:返回指定路径中与搜索模式匹配的文件系统项的可枚举集合。

public static IEnumerable<string>EnumerateFileSystemEntries(string path, string searchPattern);

// 摘要:返回指定路径中与搜索模式匹配的文件名称和目录名称的可枚举集合，还可以搜索子目录。

public static IEnumerable<string>EnumerateFileSystemEntries(string path, string searchPattern, SearchOption searchOption);

//获取目录数组

// 摘要:获取指定目录中子目录的名称。

public static string[] GetDirectories(string path);

// 摘要:从当前目录获取与指定搜索模式匹配的目录的数组。

public static string[] GetDirectories(string path, string searchPattern);

// 摘要:获取当前目录中与指定搜索模式匹配并使用某个值确定是否在子目录中搜索的目录的数组。

public static string[] GetDirectories(string path, string searchPattern, SearchOption searchOption);

<br>//获取文件

// 摘要:返回指定目录中的文件的名称。

public static string[] GetFiles(string path);

// 摘要: 返回指定目录中与指定搜索模式匹配的文件的名称。

public static string[] GetFiles(string path, string searchPattern);

// 摘要: 返回指定目录中文件的名称，该目录与指定搜索模式匹配并使用某个值确定是否在子目录中搜索。

public static string[] GetFiles(string path, string searchPattern, SearchOption searchOption);

// 摘要:返回指定目录中所有文件和子目录的名称。

public static string[] GetFileSystemEntries(string path);

// 摘要: 返回与指定搜索条件匹配的文件系统项的数组。

public static string[] GetFileSystemEntries(string path, string searchPattern);

// 摘要: 获取指定路径中与搜索模式匹配的所有文件名称和目录名称的数组，还可以搜索子目录。

public static string[]GetFileSystemEntries(string path,string searchPattern, SearchOption searchOption);

　　通过上述的方法我们可以完成目录的遍历功能，所以大家在练习本节的方法时，希望是以完成一个目录遍历的小例子为最终目的，我会在下篇中上传我做的一个目录遍历的程序给大家做参考，希望能给大家提供帮助。

## 获取和设置目录属性

// 摘要:获取一个 System.Security.AccessControl.DirectorySecurity 对象，该对象封装指定目录的访问控制列表(ACL) 项。

public static DirectorySecurity GetAccessControl(string path);

// 摘要:获取一个 System.Security.AccessControl.DirectorySecurity 对象，它封装指定目录的指定类型的访问控制列表(ACL) 项。

public static DirectorySecurity GetAccessControl(string path, AccessControlSections includeSections);

// 摘要:获取目录的创建日期和时间。

public static DateTime GetCreationTime(string path);

// 摘要:获取目录创建的日期和时间，其格式为协调世界时 (UTC)。

public static DateTime GetCreationTimeUtc(string path);

// 摘要:返回上次访问指定文件或目录的日期和时间。

public static DateTime GetLastAccessTime(string path);

// 摘要:返回上次访问指定文件或目录的日期和时间，其格式为协调世界时 (UTC)。

public static DateTime GetLastAccessTimeUtc(string path);

// 摘要: 返回上次写入指定文件或目录的日期和时间。

public static DateTime GetLastWriteTime(string path);

// 摘要: 返回上次写入指定文件或目录的日期和时间，其格式为协调世界时 (UTC)。

public static DateTime GetLastWriteTimeUtc(string path);

// 摘要:将 System.Security.AccessControl.DirectorySecurity 对象描述的访问控制列表 (ACL) 项应用于指定的目录。

public static void SetAccessControl(string path, DirectorySecurity directorySecurity);

// 摘要:为指定的文件或目录设置创建日期和时间。

public static void SetCreationTime(string path, DateTime creationTime);

// 摘要:设置指定文件或目录的创建日期和时间，其格式为协调世界时 (UTC)。

public static void SetCreationTimeUtc(string path, DateTime creationTimeUtc);

// 摘要:设置上次访问指定文件或目录的日期和时间。

public static void SetLastAccessTime(string path, DateTime lastAccessTime);

// 摘要:设置上次访问指定文件或目录的日期和时间，其格式为协调世界时 (UTC)。

public static void SetLastAccessTimeUtc(string path, DateTime lastAccessTimeUtc);

// 摘要:设置上次写入目录的日期和时间。

public static void SetLastWriteTime(string path, DateTime lastWriteTime);

// 摘要:设置上次写入某个目录的日期和时间，其格式为协调世界时 (UTC)。

public static void SetLastWriteTimeUtc(string path, DateTime lastWriteTimeUtc);

## 移动、判断是否存在

// 摘要:将文件或目录及其内容移到新位置。

public static void Move(string sourceDirName, string destDirName);

// 摘要: 确定给定路径是否引用磁盘上的现有目录。

public static bool Exists(string path);

## 6、获取卷信息、根信息、当前工作目录

// 摘要:获取应用程序的当前工作目录。

public static string GetCurrentDirectory();

// 摘要:返回指定路径的卷信息、根信息或两者同时返回。

public static string GetDirectoryRoot(string path);

// 摘要:检索此计算机上格式为“<盘符>:\”的逻辑驱动器的名称。

public static string[] GetLogicalDrives();

// 摘要:检索指定路径的父目录，包括绝对路径和相对路径。

public static DirectoryInfo GetParent(string path);

// 摘要: 将应用程序的当前工作目录设置为指定的目录。

public static void SetCurrentDirectory(string path);