# FileSystemInfo

一、构造函数

|  |  |
| --- | --- |
| [*FileSystemInfo()*](http://www.csref.cn/vs100/class/FileSystemInfo().html) | 初始化 FileSystemInfo 类的新实例。 |
| [*FileSystemInfo(SerializationInfo, StreamingContext)*](http://www.csref.cn/vs100/class/FileSystemInfo(SerializationInfo,%20StreamingContext).html) | 用序列化数据初始化 FileSystemInfo 类的新实例。 |

二、成员/方法

|  |  |
| --- | --- |
| **方法名称** | **方法描述** |
|  |  |
| [*CreateObjRef*](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-CreateObjRef.html) | 创建一个对象，该对象包含生成用于与远程对象进行通信的代理所需的全部相关信息。 （继承自 MarshalByRefObject。） |
| [*Delete*](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-Delete.html) | 删除文件或目录。 |
| [*Equals(Object)*](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-Equals.html) | 确定指定的 Object 是否等于当前的 Object。 （继承自 Object。） |
| [*Finalize*](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-Finalize.html) | 允许对象在“垃圾回收”回收之前尝试释放资源并执行其他清理操作。 （继承自 Object。） |
| [*LastWriteTime*](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-GetHashCode.html) | 最近写入时间 |
| LastAccessTimeUtc | 最近访问时间，格式：标准时间 |
| *LastAccessTime* | 最近访问时间 |
| [*GetType*](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-GetType.html) | 获取当前实例的 Type。 （继承自 Object。） |
| [*InitializeLifetimeService*](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-InitializeLifetimeService.html) | 获取控制此实例的生存期策略的生存期服务对象。 （继承自 MarshalByRefObject。） |
| [*F*ullName](http://www.csref.cn/vs100/method/System-IO-FileSystemInfo-MemberwiseClone.html) | 全名 |
| *Extension* | 后缀 |
| *Exists* | 是否存在该文件。 |
| *CreationTime* | 创建时间 |
| *LastWriteTimeUtc* | 最近写入时间（标准时间） |
| *Attributes* | 属性 |
| *CreationTimeUtc* | 创建时间（标准时间） |
| *Name* | 名字 |

三、说明

FileSystemInfo 类包含文件和目录操作所共有的方法。 FileSystemInfo 对象可以表示文件或目录，从而可以作为 FileInfo 或 DirectoryInfo 对象的基础。 当分析许多文件和目录时，请使用该基类。

一个派生类只有从 FileIOPermissionAccess 枚举获得 AllAccess 权限时才能从 FileSystemInfo 继承。

在接受路径的成员中，路径可以是指文件或仅是目录。 指定路径也可以是相对路径或者服务器和共享名称的统一命名约定 (UNC) 路径。 例如，以下都是可接受的路径：

C# 中的“c:\\MyDir\\MyFile.txt”或 Visual Basic 中的“c:\MyDir\MyFile.txt”。

C# 中的“c:\\MyDir”或 Visual Basic 中的“c:\MyDir”。

C# 中的“MyDir\\MySubdir”或 Visual Basic 中的“MyDir\MySubDir”。

C# 中的“\\\\MyServer\\MyShare”或 Visual Basic 中的“\\MyServer\MyShare”。

有关通用 I/O 任务的列表，请参见通用 I/O 任务。

四、示例

手机屏幕截图

描述已自动生成结果：

电脑萤幕截图

描述已自动生成