

FileList 项目技术文档

1. 项目概述

项目名称: FileList

目标平台: .NET Framework 4.0, C# 4.0, 控制台应用程序

命名空间: FileList

功能: 扫描指定文件夹, 生成文件列表并输出到 FileList.txt, 支持树形展示或文件内容导出, 可通过命令行参数灵活过滤。

2. 命令行接口

2.1 参数定义

程序接受最多 4 个位置参数, 按顺序定义如下:

位置	参数名	类型	默认值	说明
1	folderPath	string	应用程序所在目录	要扫描的文件夹路径
2	includeExts	string	空	包含的扩展名列表, 用 分隔
3	excludeExts	string	空	排除的扩展名列表, 格式同上
4	showContent	bool	false	是否输出文件内容, true 或 false

2.2 忽略规则

- 若未提供参数, 对应位置使用默认值
- 用户可仅提供前 N 个参数, 后面的参数自动采用默认值
- 参数可通过 “” 占位, 表示使用默认值

2.3 特殊参数

- 如果第一个参数是 /?、-h 或 -help (不区分大小写), 输出使用帮助并退出 (不执行扫描)

2.4 参数解析细节

folderPath: - folderPath 为空或省略时, 取 AppDomain.CurrentDomain.BaseDirectory

扩展名分割字符: - 在 CMD 中 | 是管道符, 用户需转义或用双引号包裹整个参数 (如 “.cs|.php” 或 .cs^|.php), 程序本身不处理转义

扩展名规范化 (NormalizeExtensions): - 对每个分割后的字符串去除首尾空白 - 若不以 . 开头, 则自动添加 . (例如 asp → .asp) - 过滤规则匹配时不区分大小写

showContent 参数: - 使用 bool.TryParse 解析, 仅接受 true 或 false (不区分大小写) - 解析失败打印错误并退出

3. 输出模式

3.1 树形模式 (showContent = false, 默认)

输出格式示例:

```
E:\Web
|—AIAPP.md
|—index.html
|—src
|   |—app.tsx
```

```

|   └─components
|     └─Layout.tsx
└─vite.config.ts

```

规则:

- 首行为根目录的完整路径（不带树形连接符）
- 每个子项（文件或目录）前均使用 `|——`（非最后一项）或 `└——`（最后一项）
- 缩进使用 4 个空格或 `|` 结构：
 - 对于根的直接子项：无任何前导缩进，即连接符紧接行首
 - 对于更深层级：
 - 若父节点是最后一项（使用 `└——` 输出的），子项缩进为父级缩进 + （4 个空格）
 - 若父节点不是最后一项（使用 `|——` 输出的），子项缩进为父级缩进 + `|`（竖线+3 个空格）
- 目录仅当其子树（递归）中包含至少一个符合过滤条件的文件时，才被显示。空目录（无文件或无匹配文件）不输出
- 同级条目（文件与目录混合）按名称字符串升序排列，排序比较采用 `StringComparer.OrdinalIgnoreCase`

3.2 内容模式 (showContent = true)

输出格式示例:

文件完整路径:
文件文本内容 (或提示信息)

规则:

- 每个文件条目之间用 60 个短横线 - 分隔
- 文件列表扁平，仅包含符合过滤条件的文件，无目录项
- 文件顺序：按完整路径字符串升序排列 (`OrdinalIgnoreCase` 比较)

内容读取注意事项:

- 使用 UTF-8 编码读取文件全部文本 (`File.ReadAllText(path, Encoding.UTF8)`)，该方式可正确处理带 BOM 的 UTF-8 文件
- 若文件大小超过 1,048,576 字节 (1 MB)，不读取内容，输出 [File size exceeds 1MB, content not displayed]
- 若文件扩展名不在预定义的文本文件扩展名集合中，视为非文本文件，不读取内容，输出 [Non-text file, content not displayed]
- 若文件读取过程发生权限或 I/O 错误，不忽略错误，程序立即终止并报错（内容模式下的权限或读取错误被认为是严重问题）

文本文件扩展名集合 (TextExtensions):

```

.txt, .md, .markdown, .html, .htm, .css, .js, .ts, .tsx, .jsx,
.json, .xml, .csv, .config, .cs, .vb, .cpp, .c, .h, .hpp,
.py, .php, .asp, .aspx, .ini, .log, .bat, .cmd, .sh, .yml,
.yaml, .sql, .svg, .rb, .java, .pl, .pm, .go, .rs, .toml,
.lock, .gitignore, .gitattributes, .editorconfig, .env, .proto

```

4. 过滤与目录显示逻辑

4.1 文件包含/排除规则 (IsFileIncluded 函数)

参数优先级:

1. 如果 includeExts 非空, 则文件扩展名必须存在于该数组中 (忽略 excludeExts)
2. 否则, 如果 excludeExts 非空, 则文件扩展名不能存在于该数组中
3. 若两者都为空, 所有文件均被包含

比较方式: - 扩展名均转化为小写或使用 OrdinalIgnoreCase 比较, 确保不区分大小写

4.2 目录节点的保留逻辑 (BuildTree 函数)

处理逻辑:

- 文件: 只保留符合过滤条件的
- 子目录: 仅当递归判断其子树包含至少一个符合条件的文件时, 才保留该子目录
- 最终, 目录节点的 Children 列表由所有保留的子项组成
- 如果 Children 为空 (即没有任何符合条件的后代文件), 该目录节点不被保留 (即不被添加到其父节点的子项列表中)
- 根目录不受此限制 (始终输出根路径)

5. 文件与目录枚举

- 使用 DirectoryInfo.GetFileSystemInfos() 获取所有文件系统项, 该方法默认返回包含隐藏文件、系统文件的所有项
- 所有目录遍历使用递归
- 若在遍历过程中遇到 UnauthorizedAccessException, 程序应立即终止, 不跳过无权限的目录

6. 输出文件

- 目标文件路径: Path.Combine(folderPath, "FileList.txt")
- 文件存在处理:
 - 如果该路径已存在且是一个目录, 程序输出错误信息并退出 (错误码 1)
 - 如果已存在一个文件, 直接覆盖
- 写入编码: UTF-8 无 BOM (使用 File.WriteAllText 或 StreamWriter 指定 Encoding.UTF8)

7. 运行环境检测与程序退出

7.1 父进程检测

目的: 区分”从命令行启动 (cmd/powershell) “与” 双击 EXE 启动”

实现方法: 通过 WMI 查询父进程名称

具体步骤: - 使用 System.Management.ManagementObjectSearcher 执行查询: `sql SELECT ParentProcessId FROM Win32_Process WHERE ProcessId = <当前进程 ID>` - 获取父进程 ID 后, 通过 Process.GetProcessById 获取进程, 再获取 .ProcessName, 转为小写进行比较 - 若父进程名为 “cmd” 或 “powershell”, 视为命令环境; 否则视为独立运行 - 如果 WMI 查询异常 (例如无权限或系统不支持), 默认按独立运行处理 (安全策略)

7.2 程序结束行为

命令行环境 (isCmdEnv == true): - 执行完毕 (成功或失败) 后直接返回退出码 (0 或 1), 不暂停

独立运行环境 (isCmdEnv == false): - 无论成功或失败, 在控制台打印结果后, 输出 "Press any key to exit..." 并等待按键, 然后退出

退出状态: - 成功完成: 控制台输出 FileList.txt created successfully., 退出码 0 - 任何异常或校验失败: 在标准错误流输出以 Error: 开头的消息, 退出码 1

8. 错误处理详细

程序在 Main 函数中捕获以下异常 (不再区分, 统一处理):

- **UnauthorizedAccessException:** 输出 Error: access denied - {消息}
- **IOException:** 输出 Error: I/O error - {消息}
- **其他 Exception:** 输出 Error: {消息}

注意: - 所有权限错误或 I/O 错误均会导致程序立即退出, 不会部分输出结果

9. 核心算法细节

9.1 树形结构构建

步骤:

1. 创建根节点 TreeNode, Name 和 FullPath 均设置为根目录路径, IsDirectory = true
2. 调用 BuildTree(root, includeExts, excludeExts):
 - 若节点为文件, 直接返回 IsFileIncluded 结果
 - 若为目录, 枚举所有子项, 对每个子项递归调用 BuildTree。仅将返回 true 的子项加入临时列表
 - 对临时列表排序后赋给 node.Children, 返回 children.Count > 0
3. 完成构建后, 调用 RenderTree(root, "", true, sb) 渲染为字符串

9.2 树形渲染缩进计算

递归函数: RenderTree(node, indent, isLast, sb)

参数说明: - indent: 当前行的前置缩进字符串, 初始为空 - isLast: 该节点是否为父节点的最后一个子节点

处理逻辑:

- 对于非根节点: 输出 indent + (isLast ? "└──" : "├──") + node.Name
- 根节点特殊判断: 当 node.IsDirectory && node.Name == node.FullPath 时, 直接输出 node.Name, 不添加任何前缀
- 递归子节点时, 计算子节点的 indent:
 - 如果当前节点是根节点 (node.Name == node.FullPath), 子节点的 indent 为空字符串 ""
 - 否则, 子节点 indent = 当前 indent + (若当前节点 isLast 则 " ", 否则 " | ")
- 遍历 node.Children, 对每个子节点调用 RenderTree(child, childIndent, lastChild, sb)

9.3 内容模式文件收集

函数: GetFilteredFiles(rootPath, includeExts, excludeExts)

处理逻辑： - 递归枚举所有文件系统项，仅收集符合条件的文件全路径 - 最终列表排序后输出

10. 类定义

```
class TreeNode
{
    public string Name;           // 文件或目录名
    public string FullPath;      // 完整路径
    public bool IsDirectory;     // 是否为目录
    public List<TreeNode> Children; // 子节点 (仅目录有效)
}
```

11. 帮助信息内容

当用户调用 `FileList.exe /?` 等时，输出以下英文帮助：

`FileList` - List files in a directory tree.

Usage:

```
FileList.exe [folderPath] [includeExts] [excludeExts] [showContent]
```

Parameters:

```
folderPath   : Target directory. Default: current directory.
includeExts  : Use '|' separated extensions (e.g. .cs|.php).
               If provided, only these extensions are listed. (ignore excludeExts)
excludeExts  : Extensions to exclude (e.g. .tmp|.log).
showContent  : 'true' to output file paths and content, 'false' for tree view.
```

Notes:

- In CMD, escape '|' with ^ or use quotes: ".cs|.php"
- Extensions are case-insensitive and may be specified with or without leading dot.

12. 其他技术约定

- 项目必须引用 `System.Management.dll`
- 所有比较和排序除非特别说明，均使用 `StringComparer.OrdinalIgnoreCase`
- 文本输出统一使用 UTF-8 编码
- 内容模式下文件大小限制为 1,048,576 字节，常量硬编码
- 文本扩展名集合硬编码在 `TextExtensions` 静态字段中
- 不得使用 .NET 4.5+ 特性，必须兼容 C# 4.0