

FileContentV2

功能以及更新介绍

- v1
 1. 支持对某一个目录下的文件路径进行解析，并将各个文件和文件夹的路径以树的结构组织起来。
 2. 可以将文件树打印在控制台，从而可以很方便地查阅一个文件的底层结构。
 3. 可以对大文件夹进行约束，若一个文件夹内文件超过约束则在打印时将相应文件夹折叠。
- v2
 1. 支持使用正则表达式对文件名进行筛选，可根据文件名后缀或文件名设置筛选条件。
 2. 更加灵活的策略运用，支持两种筛选策略以及两种筛选约束表。
 3. 文件树打印时可支持精简打印，即只打印和目标文件相关的文件树路径。
 4. 配置导入与导出，支持将文件树的策略2规则配置导出或加载，可以更方便地生成配置模板文件。
 5. 更好的抽象设计使底层解析和顶层用户操作得以分离。

使用说明与演示

- 使用流程

1. 创建FileContent对象。

```
cont=FileContent()
```

2. 设置相关规则

```
#设置起始目录路径，此处利用os模块读取到了.py文件所在目录的路径,默认为空,必须设置。
cont.set_startpath(os.getcwd())
#设置筛选策略,默认为"MODE_RAR",可以不设置。
cont.set_mode("MODE_RAR")
#设置大文件夹折叠约束，默认为20，可以不设置
cont.set_filefold(20)
#向指定筛选约束表写入筛选规则（正则表达式）
cont.set_rules("(.*).(docx)","ACCEPT_RULE")
```

3. 调用接口生成文件树

```
#生成文件树
cont.filetree_update()
```

4. 将结果打印在控制台

```
#参数指明了打印方式
cont.show('SAMPLE_TREE_11')
```

- 策略模式介绍

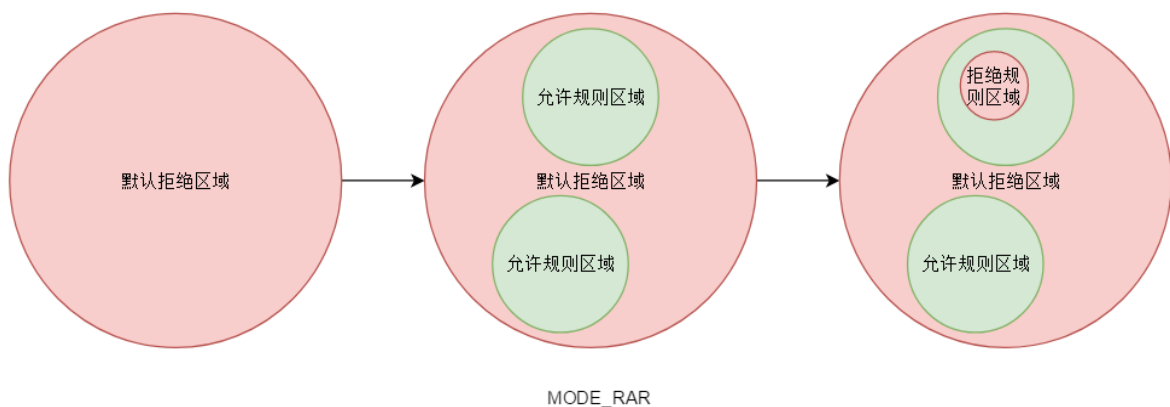
存在两个筛选约束表，一个为允许列表（ACCEPT），一个为拒绝列表（REJECT）。

1. "MODE_RAR"

首先将文件名与允许列表内规则一一匹配，如果没有匹配到，则拒绝该文件，如果匹配到了，进入下一步。然后将文件名和拒绝列表一一匹配，如果没有匹配到，则允许该文件，如果匹配到了，那么拒绝该文件。

#设置筛选策略,默认为"MODE_RAR",可以不设置。

```
cont.set_mode("MODE_RAR")
```

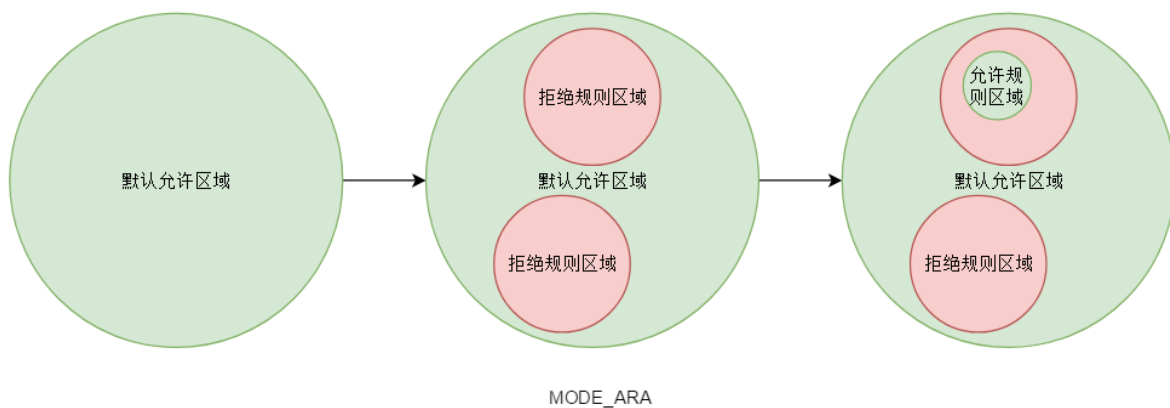


2. "MODE_ARA"

首先将文件名与拒绝列表内规则一一匹配，如果没有匹配到，则允许该文件，如果匹配到了，进入下一步。然后将文件名和允许列表一一匹配，如果没有匹配到，则拒绝该文件，如果匹配到了，那么允许该文件。

#设置筛选策略,默认为"MODE_RAR",可以不设置。

```
cont.set_mode("MODE_ARA")
```



3. 示例1

如果要检索出一个文件内的所有py文件和docx文件，同时不希望检索到pyd文件。那么可以先设置规则策略为MODE_RAR，这样默认情况下不检索任何文件，然后我们往允许规则表里输入规则使得程序可以检索py和docx，但这样检索过后可能会检索到pyd文件，那我们就在拒绝规则表里输入规则使得程序不检索pyd文件即可。

4. 示例2

如果不想检索出一个文件内的txt文件，除了文件名为sata*的txt文件（比如sata1，sata2）。那么可以先设置规则策略为MODE_ARA，这样默认情况下检索所有文件。然后往拒绝规则表里输入规则使得程序不检索txt文件，然后往允许规则表里输入规则使得程序不要丢弃文件名为sata*的txt文件。

- 打印方式介绍

1. 完整树打印

将所有结构呈现出来，好处是可以看到文件全貌，坏处是文件结构比较庞大时打印输出的内容会很多。

```
cont.show('TREE')
```

2. 简单打印

按列表打印而不是按照树结构打印，只打印目标文件的路径。

```
cont.show('SAMPLE')
```

3. 简单树打印（推荐）

对完整树进行简化从而去除和目标文件无关的树枝。格式为“SAMPLE_TREE_[优化等级]”

```
cont.show('SAMPLE_TREE_5')
```

- 示例程序

程序内容：

```
if __name__ == '__main__':
    #创建对象
    cont=FileContent()

    #写入规则
    cont.set_startpath(os.getcwd())
    cont.set_mode("MODE_RAR")
    cont.set_filefold(20)

    #往允许筛选表里写入规则，检索后缀为docx和pdf的文件
    cont.set_rules("(.*).(docx|pdf)","ACCEPT_RULE")

    #调用并显示

    cont.filetree_update()
```

```
cont.show('SAMPLE_TREE_11')
```

程序输出：

```
| 教研室选择
|   | 个人简介.docx
|   | 咨询问题.docx
|   | 团队丢弃区
|       | 复杂多维信息处理研究团队.pdf
|       | 感知与智能控制系统团队.pdf
|       | 成像探测与智能感知团队.pdf
|       | 空天院微波微系统部2020年招生宣传.pdf
|   | 团队预选区
|       | 光纤传感与器件研究团队
|           | 光纤传感与器件研究团队.pdf
|       | 光通信研究团队
|           | 光通信研究团队.pdf
|       | 电磁认知应用与先进射频集成电路团队
|           | 电磁认知应用与先进射频集成电路团队.pdf
```

改变筛选规则：打算检索不同种类的文件

```
#往允许筛选表里写入规则，检索后缀为docx和txt的文件
cont.set_rules("(.*).(docx|txt)","ACCEPT_RULE")
```

程序输出：

```
| 教研室选择
|   | 个人简介.docx
|   | 咨询问题.docx
|   | 团队预选区
|       | Novamind团队
|           | 相关信息.txt
|       | 光纤传感与器件研究团队
|           | 相关信息.txt
|       | 光通信研究团队
|           | 相关信息.txt
|       | 物联网智能芯片与系统团队
|           | 相关信息.txt
|       | 电磁认知应用与先进射频集成电路团队
|           | 相关信息.txt
|       | 计算机视觉与机器智能团队
|           | 相关信息.txt
```

改变筛选规则：检索出来的文件里不想看到包含“简介”两字的的文件。

```
#往允许筛选表里写入规则，检索后缀为docx和pdf的文件
cont.set_rules("(.*).(docx|txt)","ACCEPT_RULE")
#往拒绝筛选表里写入规则
cont.set_rules("(.*)简介(.*)","REJECT_RULE")
```

程序输出：

```
| 教研室选择
|   | 咨询问题.docx
|   | 团队预选区
|   |   | Novamind团队
|   |   |   | 相关信息.txt
|   |   | 光纤传感与器件研究团队
|   |   |   | 相关信息.txt
|   |   | 光通信研究团队
|   |   |   | 相关信息.txt
|   |   | 物联网智能芯片与系统团队
|   |   |   | 相关信息.txt
|   |   | 电磁认知应用与先进射频集成电路团队
|   |   |   | 相关信息.txt
|   |   | 计算机视觉与机器智能团队
|   |   |   | 相关信息.txt
```