

首页 » Powershell » Powershell 编写和运行脚本

Powershell 编写和运行脚本

28 4月, 2012 在 Powershell 标签 Powershell教程 / 脚本 来自 Mooser Lee

本文目录



1. 通过重定向创建脚本
2. 通过编辑器创建脚本
3. 运行Powershell脚本
4. 执行策略限制
5. 像命令一样执行脚本

一个Powershell仅仅是一个包含Powershell代码的文本文件。如果这个文本文件执行，Powershell解释器会逐行解释并执行它的的语句。Powershell脚本非常像以前CMD控制台上的批处理文件。您可以通过非常简单的文本编辑工具创建Powershell脚本。

通过重定向创建脚本

如果您的脚本不是很长，您甚至可以直接在控制台中要执行的语句重定向给一个脚本文件。

```
PS E:> '"Hello,Powershell Script"' > MyScript.ps1
PS E:> .\MyScript.ps1
Hello,Powershell Script
```

这样有个缺点，就是您的代码必须放在闭合的引号中。这样的书写方式一旦在脚本内部也有引号时，是一件很痛苦的事。甚至您还可能希望在脚本中换行。下面的Here-strings例子不错，也就是将脚本文件通过@‘ ’@闭合起来。

```
PS E:> @'
>> Get-Date
>> $Env:CommonProgramFiles
>> #Script End
>> "files count"
>> (ls).Count
>> #Script Really End
>>
>> '@ > myscript.ps1
>>
PS E:> .MyScript.ps1

2012年4月27日 8:15:10
C:\Program Files\Common Files
files count
20
```

Here-String以@‘开头，以’@结束.任何文本都可以存放在里面，哪怕是一些特殊字符，空号，白空格。但是如果您不小心将单引号写成了双引号，Powershell将会把里面的变量进行解析。

通过编辑器创建脚本

其实非常方便的还是最地道的文版编辑器Notepad，您可以直接在Powershell控制台中打开Notepad

```
PS E:> notepad.exe .\MyScript.ps1
PS E:> notepad.exe
```

编辑完记得保存即可。

运行Powershell脚本

当您的脚本编写成功后您可能第一次会像下面的方式运行它，也就是只输入脚本的文件名，会报错。

```
PS E:> MyScript.ps1
无法将“MyScript.ps1”项识别为 cmdlet、函数、脚本文件或可运行程序的名称。请检查名称的拼写，如果包括
路径，请确保路径正确，然后重试。
所在位置 行:1 字符: 13
+ MyScript.ps1 < <<<
+ CategoryInfo          : ObjectNotFound: (MyScript.ps1:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

Suggestion [3,General]: 未找到命令 MyScript.ps1，但它确实存在于当前位置。Windows PowerShell 默认情况
下不从当前位置加载命令。如果信任此命令，请改为键入 ".MyScript.ps1"。有关更多详细信息，请参阅 "get-h
elp about_Command_Precedence"。
```

除非您使用相对路径，或者绝对路径

```
PS E:> .\MyScript.ps1

2012年4月27日 8:33:03
C:\Program Files\Common Files
files count
20

PS E:> E:\MyScript.ps1

2012年4月27日 8:33:11
C:\Program Files\Common Files
files count
20
```

执行策略限制

Powershell一般初始化情况下都会禁止脚本执行。脚本能否执行取决于Powershell的执行策略。

```
PS E:> .\MyScript.ps1
无法加载文件 E:\MyScript.ps1，因为在此系统中禁止执行脚本。有关详细信息，请参阅 "get-help about_sign
ing"。
所在位置 行:1 字符: 15
+ .\MyScript.ps1 < <<<
+ CategoryInfo          : NotSpecified: (:) [], PSSecurityException
+ FullyQualifiedErrorId : RuntimeException
```

只有管理员才有权限更改这个策略。非管理员会报错。

查看脚本执行策略，可以通过：

```
PS E:> Get-ExecutionPolicy
```

更改脚本执行策略，可以通过

```
PS E:> Get-ExecutionPolicy
Restricted
PS E:> Set-ExecutionPolicy UnRestricted
```

执行策略更改

执行策略可以防止您执行不信任的脚本。更改执行策略可能会使您面临 `about_Execution_Policies` 帮助主题中所述的安全风险。是否要更改执行策略?
[Y] 是(Y) [N] 否(N) [S] 挂起(S) [?] 帮助 (默认值为“Y”): y

脚本执行策略类型为: **Microsoft.PowerShell.ExecutionPolicy**

查看所有支持的执行策略:

```
PS E:> [System.Enum]::GetNames([Microsoft.PowerShell.ExecutionPolicy])
Unrestricted
RemoteSigned
AllSigned
Restricted
Default
Bypass
Undefined
```

Unrestricted:权限最高, 可以不受限制执行任何脚本。

Default:为Powershell默认的策略: **Restricted**, 不允许任何脚本执行。

AllSigned: 所有脚本都必须经过签名才能在运行。

RemoteSigned: 本地脚本无限制, 但是对来自网络的脚本必须经过签名。

关于Powershell脚本的签名在后续会谈到。

像命令一样执行脚本

怎样像执行一个命令一样执行一个脚本, 不用输入脚本的相对路径或者绝对路径, 甚至*.ps1扩展名。

那便将脚本的执行语句保存为别名吧:

```
PS E:> Set-Alias Invok-MyScript .MyScript.ps1
PS E:> Invok-MyScript
```

```
2012年4月28日 0:24:22
C:\Program Files\Common Files
files count
20
```





新注册用户域名抢购1元起

.com首年28元 .cn首年19元

立即抢购



本文链接: <https://www.pstips.net/powershell-create-and-start-scripts.html>

请尊重原作者和编辑的辛勤劳动,欢迎转载,并注明出处!

相关文章推荐:

1. Powershell变量的作用域
2. Powershell 增强脚本的可读性

3. Powershell自动执行脚本之profile
 4. Powershell强类型数组
 5. Powershell属性：描述对象是什么
 6. Powershell Where-Object 条件过滤
 7. Powershell Switch 循环
 8. Powershell指定函数的返回值
 9. Powershell what-if
 10. PowerShell处理XML（三） 浏览扩展类型系统
-