

3-2-4：管道使用

作者简介

曾垂鑫，毕业于中国人民大学商务管理专业。微软（2013-2017五届）MVP，具有丰富的项目实施和解决方案经验，51CTO传媒推荐博客、博客之星，微软MCP/MCTS/MCITP/MCSA/MCSE认证系统专家和解决方案顾问。
属于实战派的讲师，具备丰富的万人规模以上企业的运维经验。课程内容侧重于实战实用。

曾垂鑫的博客地址：<http://543925535.blog.51cto.com/>

曾垂鑫的课程地址：http://edu.51cto.com/lecturer/index/user_id-639838.html

课程后续会形成完善的Windows运维工程师路线图，涵盖企业常用的Windows平台运维技术，后期课程包括但不限于AD、Exchange、Windows、System Center、Powershell、Hyper-v、Office 365等。

管道

管道通过传输一个命令，将其输出作为另外一个命令的输入。
Powershell的管道类似Unix和Linux的shell中的管道功能。

管道输出到CSV

```
PS C:\> Get-Mailbox -Identity zengchuixin101 | Export-Csv c:\usermailbox.csv
```

打开CSV文件

```
PS C:\> Invoke-Item .\usermailbox.csv
```

查看打开的CSV文件

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	#TYPE	Microsoft.Exchange.Data.Directory.Management	Mailbox															
2	Database	MailboxPr	MessageC	MessageC	MailboxPr	UseData	RetainDele	DeliverTo	IsExcluded	IsHierarchy	IsHierarchy	HasSnacky	LitigationH	SingleItem	RetentionH	EndDateFc	StartDateF	Retentic
3	test		FALSE	FALSE	Microsoft.E	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE			
4																		
5																		

筛选后再输出

```
PS C:\> Get-Mailbox -Identity zengchuixin101 | select name, emailaddress, displayname, samaccountname | Export-Csv c:\usermailbox.csv
```

```
PS C:\> Invoke-Item .\usermailbox.csv
```

管道输出到HTML

```
PS C:\> Get-Mailbox -Identity zengchuixin101 | ConvertTo-Html | Out-File c:\usermailbox.html
```

```
PS C:\> Invoke-Item .\usermailbox.html
```

例：

```
Get-Process | Select-Object Name,ID,VM,PM | ConvertTo-Html -Title "this is the head" | Out-File C:\ps\process.html
```

（一）大部分情况下，使用相同名词的命令都可以使用ByValue的方式相互之间进行管道传输

例如Get-service -name s* | stop-service

```
PS C:\powershellscript> Get-Service -Name s* | Get-Member

TypeName: System.ServiceProcess.ServiceController

Name           MemberType      Definition
-----
Name           AliasProperty   Name = ServiceName
RequiredServices AliasProperty   RequiredServices = ServicesDependedOn
Disposed       Event           System.EventHandler Disposed(System.Object, System.Event
Close          Method          void Close()
Continue       Method          void Continue()
```

这两个命令有一个相同的参数-name，都支持通过byvalue的方式进行传递。

```
-Name <String[]>
    Specifies the service names of services to be retrieved. Wildcards are permitted. By default, this cmdlet gets all of the services on the computer.

必需?          false
位置?          0
别名?          None
是否接受管道输入? True (ByPropertyName, ByValue)
是否接受通配符? false
```

通过管道传递执行命令，可以成功。

支持的参数名称和类型都是一样的，所以可以顺利的通过管道进行传递。

```
PS C:\powershellscript> Get-Service -Name BITS | Stop-Service

PS C:\powershellscript> Get-Service -Name BITS | Start-Service
```

(二) **bypropertyname** 的意思是如果管道的输入对象里面有一个属性，他的名字和类型都对输出命令的某一个参数的名字和类型都对上号，那么这样的管道也是成立的。

执行get-adcomputer ex1601 | get-service bits会失败

```
PS C:\Users\administrator.DEMO> get-adcomputer ex1601 | get-service bits
Get-Service 无法对参数 "ComputerName" 执行参数验证。参数为 Null 或空。 请提供一个不为 Null 或空的参数，然后重试该命令。
所在位置 行:1 字符: 25
+ get-adcomputer ex1601 | get-service bits
+ ~~~~~
+ CategoryInfo          : InvalidData: (CN=EX1601,CN=Computers,DC=demo,DC=com:PSObject) [Get-Service], ParameterBindingValidationException
+ FullyQualifiedErrorId : ParameterArgumentValidationError,Microsoft.PowerShell.Commands.GetServiceCommand
```

为什么不行呢？

因为我们前一个命令是AdComputer的类型，如图。

```
PS C:\ps> Get-ADComputer ex1601 | gm

TypeName:Microsoft.ActiveDirectory.Management.ADComputer

Name                MemberType          Definition
-----
Contains            Method              bool Contains(string propertyName)
Equals              Method              bool Equals(System.Object obj)
GetEnumerator        Method              System.Collections.IDictionaryEnumerator GetEnumerator()
GetHashCode          Method              int GetHashCode()
GetType             Method              Type GetType()
ToString             Method              string ToString()
```

而我们后面一条命令查看一下get-service 的帮助文档，byvalue需要的是serviceController类型，对不上，**这两个命令无法通过byvalue的方式进行传递。**

Get-Service Help

Find: byvalue Previous Next 2 matches

Required?	false
Position?	Named
Default value	
Accept pipeline input?	false
Accept wildcard characters?	

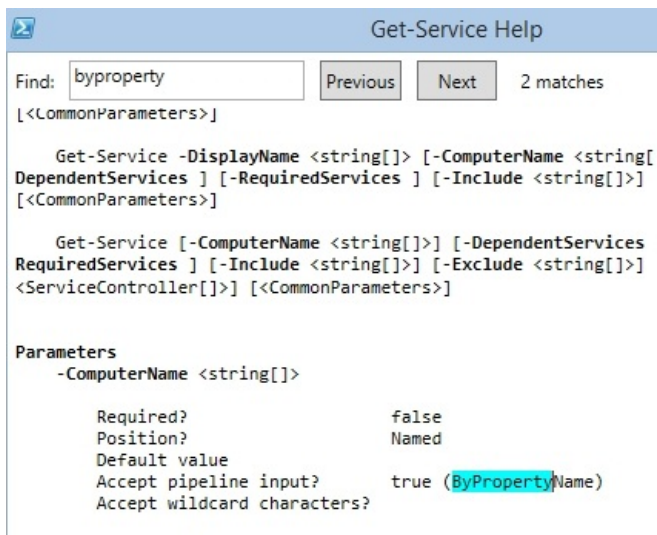
-Include <string[]>

Required?	false
Position?	Named
Default value	
Accept pipeline input?	false
Accept wildcard characters?	

-InputObject <ServiceController[]>

Required?	false
Position?	Named
Default value	
Accept pipeline input?	true (ByValue)

接下来我们看一下get-service命令支持bypropertyname,他接受一个管道对象的属性为computername，类型为字符串string的作为他的参数输入



而对于我们的get-adcomputer的属性里面，可以看见他有一个叫做name的属性，类型为字符串string。因为名字不匹配(不是get-service命令需要的computername属性)，所以管道仍然无法传输。

这时候我们只需要把名字属性匹配就行了。

get-adcomputer ex1601 | select name, @{name="computername";expression={\$_.name}} | Get-Service bits

```
PS C:\Users\administrator.DEMO> get-adcomputer ex1601 | select name, @{name="computername";expression={$_.name}} | Get-Service bits
```

Status	Name	DisplayName
Running	bits	Background Intelligent Transfer Ser...

总结（如果通过bypropertyname传递）：

- 1) 名字属性匹配
- 2) 类型匹配（例如都是字符串类型）