

How PowerShell Pipelines works

What is a pipeline

- Cmdlet1 | Cmdlet2
- This sends (pipes) the result of Cmdlet1 to the input of Cmdlet2
- There are two ways PowerShell sends data over a pipeline
 - ByValue
 - ByPropertyName
- ByValue is always tried first
- If ByValue is not possible ByPropertyName is tried next
- If ByPropertyName is not possible the pipeline fails with an error

ByValue

- Cmdlet1 | Cmdlet2
- Sends the whole output of Cmdlet1 to a parameter of Cmdlet2
- The parameter of Cmdlet2
 - Must accept pipelines
 - Must accept the object type that Cmdlet1 is sending
- To test this
 - Cmdlet1 | Get-Member
 - Discover the “Type Name” at the top of the Get-Member result
 - Get-Help Cmdlet2 -ShowWindow
 - Look for a parameter that will satisfy the requirements above

ByValue Example

- Get-Service -Name Spooler,WinRM | Set-Service -StartType Automatic
- Get-Service -Name Spooler,WinRM | Get-Member

```
Get-Service -Name Spooler | Get-Member
```

```
TypeName: System.Service.ServiceController
```

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

- Get-Help Set-Service -ShowWindow
 - -InputObject (type matches)
 - -Name (type does not match)
- ByValue pipeline succeeds

```
-InputObject <System.ServiceProcess.ServiceController>
```

Specifies a ServiceController object that represents the service to change. Enter a variable that contains the object, or type a command or expression that gets the object, such as a 'Get-Service' command. You can use the pipeline to send a service object to 'Set-Service'.

Required?	true
Position?	0
Default value	None
Accept pipeline input?	True (ByValue)
Accept wildcard characters?	false

```
-Name <System.String>
```

Specifies the service name of the service to be changed. Wildcard characters aren't permitted. You can use the pipeline to send a service name to 'Set-Service'.

Required?	true
Position?	0
Default value	None
Accept pipeline input?	True (ByPropertyName, ByValue)
Accept wildcard characters?	false

ByPropertyName

- Cmdlet1 | Cmdlet2
- Sends values of properties
 - from the output of Cmdlet1
 - to corresponding input parameters in Cmdlet2
- Test this
 - Get-Help Cmdlet2 -ShowWindow
 - Locate the parameters in Cmdlet2 that will accept pipeline ByPropertyName
 - Check the object type each parameter can accept
 - Cmdlet1 | Get-Member
 - Look for properties that match the spelling of the parameter located above
 - Check what type of data the property is
 - Make sure the:
 - Spelling of the properties from Cmdlet1 and the parameters in Cmdlet2 match
 - Object types of the matching properties and parameters must also match

ByPropertyName Example

- `Import-Csv ServiceNames.csv | Set-Service -StartType Automatic`
- `Import-Csv ServiceNames.csv | Get-Member`



Name
Spooler
Winrm

```
Import-Csv ServiceNames | Get-Member
```

TypeName: System.Management.Automation.PSCustomObject

Name	MemberType	Definition
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
ToString	Method	string ToString()
Name	NoteProperty	String Name=Spooler

- `Get-Help Set-Service -ShowWindow`
 - Parameter and Property spelling match (`-Name` & `Name`)
 - `-Name` allows pipeline
 - `-Name` Pipes `ByPropertyName`
 - `[String]` is the same for `-Name` & `Name`
- `ByPropertyName` pipeline succeeds

```
-InputObject <System.ServiceProcess.ServiceController>  
Specifies a ServiceController object that represents the service to change. Enter a variable that contains the object, or type a command or expression that gets the object, such as a 'Get-Service' command. You can use the pipeline to send a service object to 'Set-Service'.
```

Required?	true
Position?	0
Default value	None
Accept pipeline input?	True (ByValue)
Accept wildcard characters?	false

`-Name <System.String>`

Specifies the service name of the service to be changed. Wildcard characters aren't permitted. You can use the pipeline to send a service name to 'Set-Service'.

Required?	true
Position?	0
Default value	None
Accept pipeline input?	True (ByPropertyName) ByValue
Accept wildcard characters?	false

If Neither ByValue or ByPropertyName works

- When pipelining fails, use one of the following
 - Parenthetical
 - Calculated Property

Parenthetical

- `Get-ADComputer -Filter * | Get-Service -Name Spooler`
 - This will not work because
 - the Name property and -ComputerName parameter
 - spelling do not match
- `Get-Service -Name Spooler -ComputerName (Get-ADComputer -Filter *).Name`
 - () tell PowerShell to do this first
 - The computer names get transposed where the (Get-ADComputer -Filter *).Name was
 - The command runs as if the computernames were typed as an array

Calculated Property

- `Get-ADComputer -Filter * | Get-Service -Name Spooler`
 - This will not work because
 - the Name property and -ComputerName parameter
 - spelling do not match
- `Get-ADComputer -Filter * |
Select-Object @{n='ComputerName';e={$_.Name}} |
Get-Service -Name Spooler`
 - The Select-Object creates a new Property called ComputerName
 - The ComputerName property holds the names of the computers
 - ComputerName Property and -ComputerName parameter spelling match
 - ByPropertyName pipeline now works