

# PowerShell Jumpstart for SQL Server DBAs





[www.mikefal.net](http://www.mikefal.net)



Mike Fal

**Microsoft**  
**CERTIFIED**  
Solutions Expert

Data Platform



```

    $trns = Get-ChildItem $dir -recurse | Where-Object {$_.name -like "*.trn"} | sort-object LastWriteTime
}
else{
    $full = Get-ChildItem $dir | Where-Object {$_.name -like "*.bak"} | Sort-Object LastWriteTime -desc | Select-Object -first 1
    $diff = Get-ChildItem $dir | Where-Object {$_.name -like "*.dff"} | sort-object LastWriteTime -desc | select-object -first 1
    $trns = Get-ChildItem $dir | Where-Object {$_.name -like "*.trn"} | sort-object LastWriteTime
}

#initialize and process full backup
$outputfile = Join-Path -Path $outputdir -ChildPath "restore_$database.sql"
$restore = Get-RestoreObject $database $full
$shfull = Get-Header $restore $smosrv
if($database.Length -eq 0)
{
    $database = $shfull.DatabaseName
    $restore.Database=$database
}

$LSNCheck = $shfull.CheckpointLSN
$files = $restore.ReadFileList($smosrv)
foreach($file in $files){
    $pfile = $file.PhysicalName
    if($newdata.Length -gt 0 -and $file.Type -eq "D"){
        $pfile=$newdata + $pfile.Substring($pfile.LastIndexOf("\"))
    }

    if($newdata.Length -gt 0 -and $file.Type -eq "L"){
        $pfile=$newlog + $pfile.Substring($pfile.LastIndexOf("\"))
    }

    $newfile = New-Object("Microsoft.SqlServer.Management.Smo.RelocateFile") ($file.LogicalName,$pfile)
    $restore.RelocateFiles.Add($newfile) | out-null
}

$ssqlout += "/******"
$ssqlout += "Restore Database Script Generated $(Get-Date)"
$ssqlout += "Database: "+$database
$ssqlout += "*****/"
$ssqlout += "--FULL RESTORE"
If($owner){$ssqlout += "EXECUTE AS LOGIN = '$owner';"}
$ssqlout += $restore.Script($smosrv)

#process differential backups
if($diff -ne $null){
    $restore = Get-RestoreObject $database $diff
    $hdiff = Get-Header $restore $smosrv

    if($hdiff.DatabaseBackupLSN -eq $LSNCheck){
        $ssqlout += "--DIFF RESTORE"
        $ssqlout += $restore.Script($smosrv)
        $LSNCheck = $hdiff.LastLSN
    }
    else{

```

Don't focus on the code,  
focus on the concepts.

Ask questions!

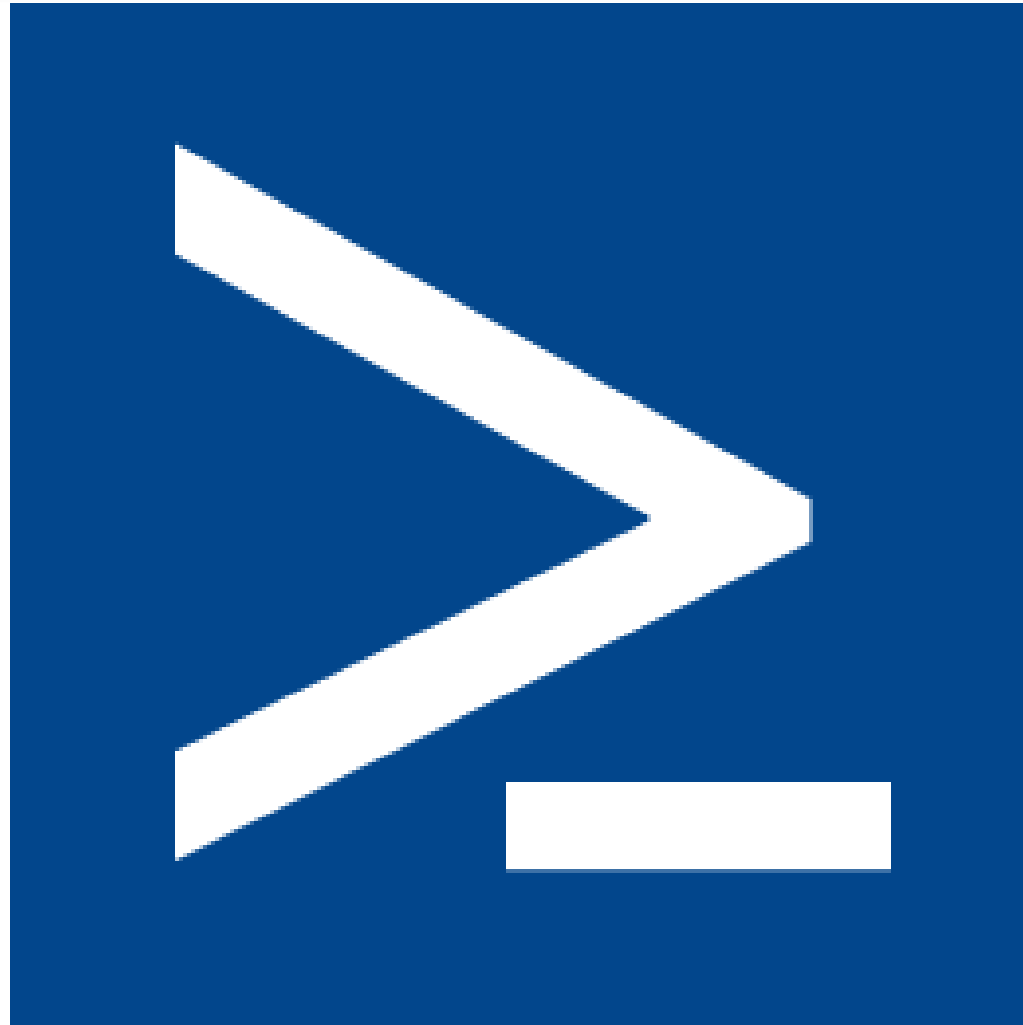
**What and Why of PowerShell**

**Language Basics**

**Working with SQL Server**

**And then what?**

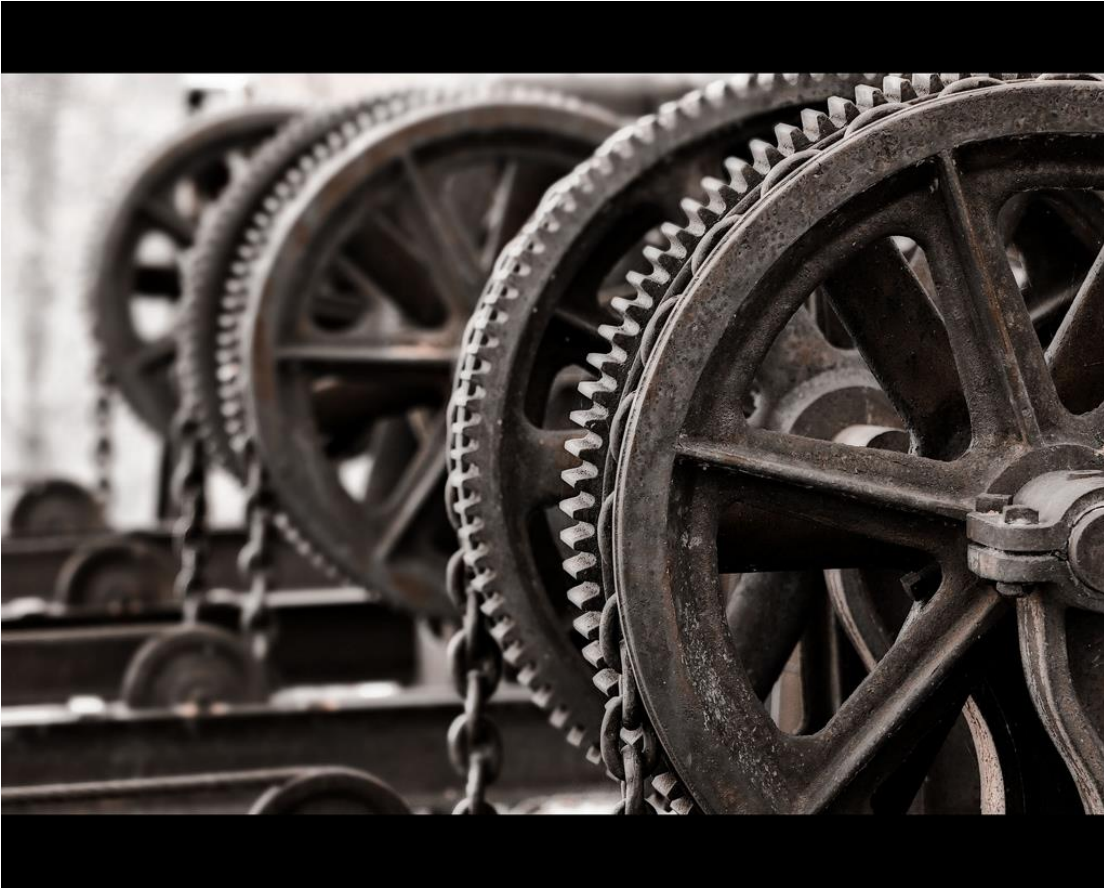
---

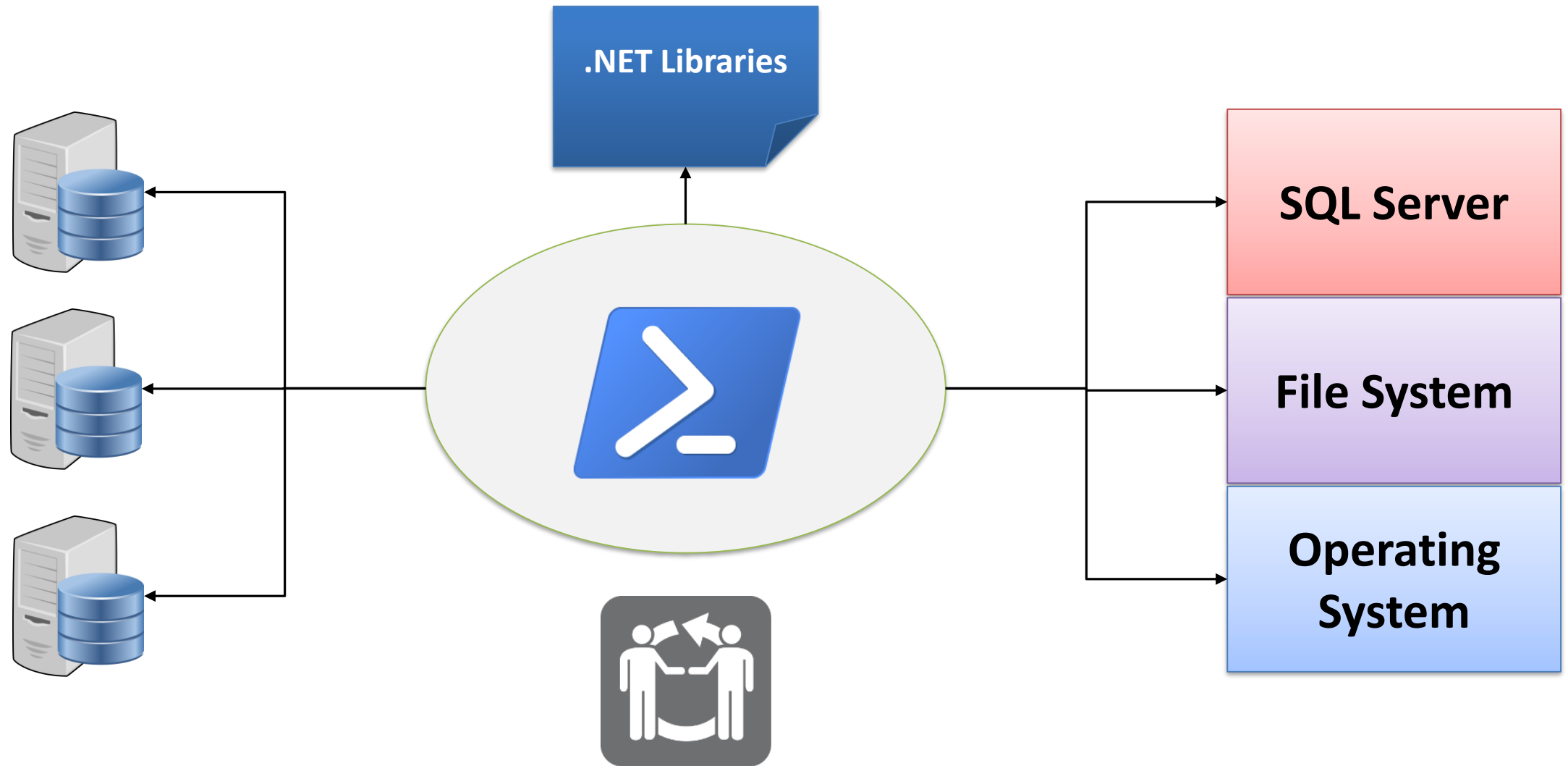


- Envisioned by Jeffery Snover – 2002
    - The Monad Manifesto
  - Powershell RC 1 Released April 2006
    - Originally called Project “Monad”
  - Current available version: 5.0
    - Just released, 4.0 has been out for a bit
-



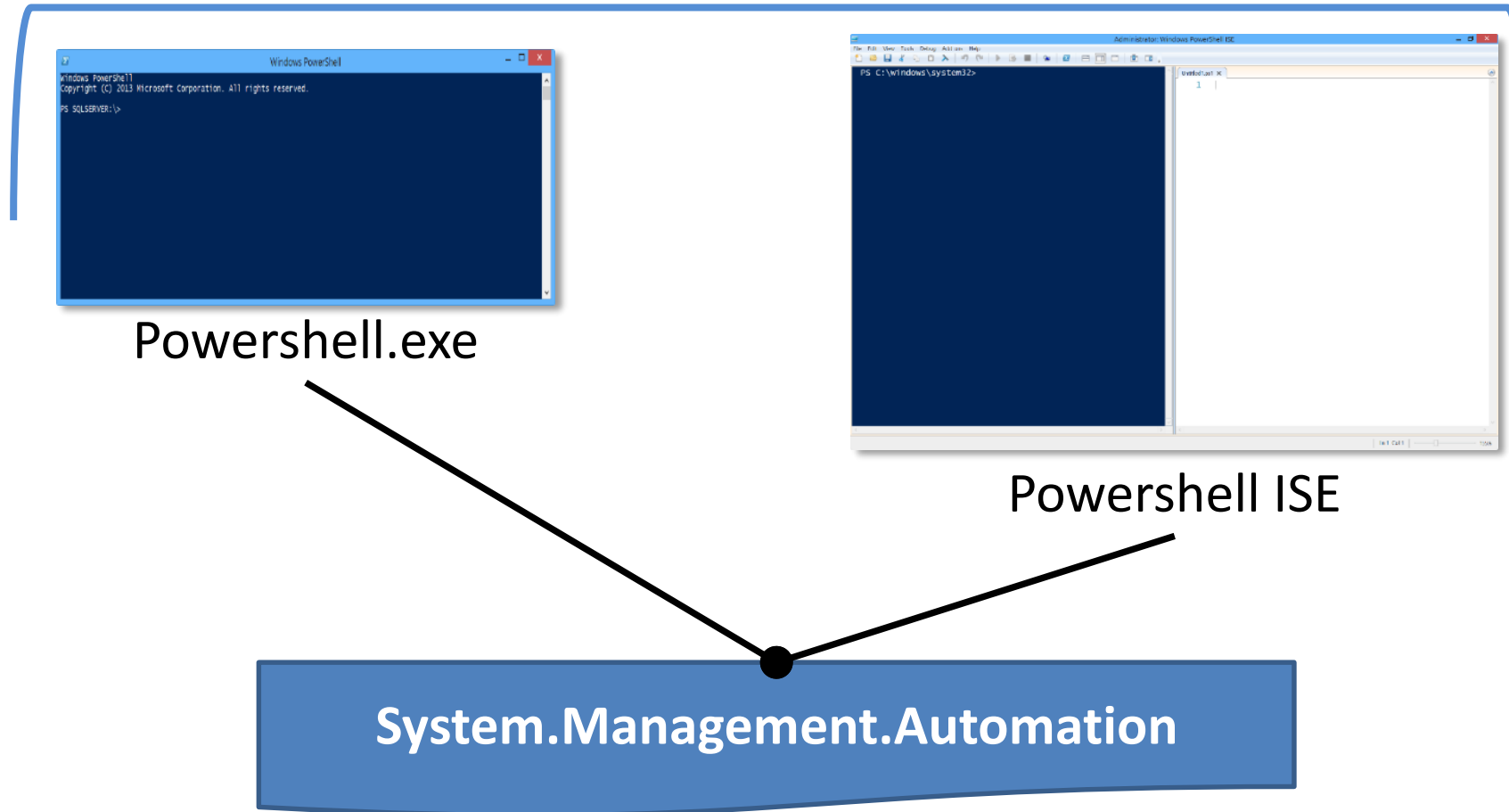
But for what?

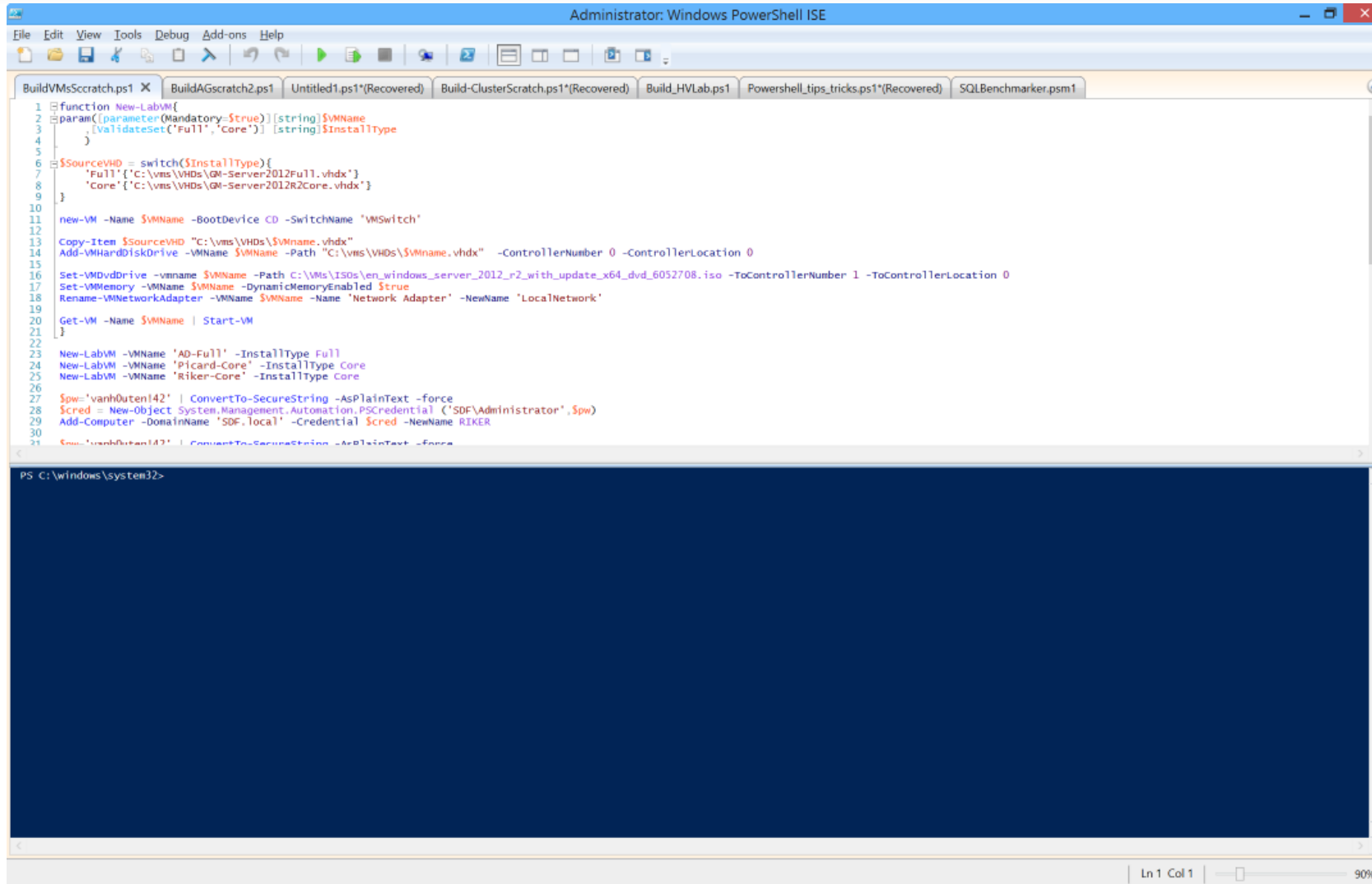






## Hosts





```
1 function New-LabVM{
2 param([parameter(Mandatory=$true)][string]$VMName
3       ,[ValidateSet('Full','Core')][string]$InstallType
4       )
5
6 $SourceVHD = switch($InstallType){
7     'Full' {'C:\vms\VHDs\GM-Server2012Full.vhdx'}
8     'Core' {'C:\vms\VHDs\GM-Server2012R2Core.vhdx'}
9 }
10
11 new-VM -Name $VMName -BootDevice CD -SwitchName 'VMSwitch'
12
13 Copy-Item $SourceVHD "C:\vms\VHDs\$VMName.vhdx"
14 Add-VMHardDiskDrive -VMName $VMName -Path "C:\vms\VHDs\$VMName.vhdx" -ControllerNumber 0 -ControllerLocation 0
15
16 Set-VMVDvdDrive -vmname $VMName -Path C:\VMS\ISOs\en_windows_server_2012_r2_with_update_x64_dvd_6052708.iso -ToControllerNumber 1 -ToControllerLocation 0
17 Set-VMMemory -VMName $VMName -DynamicMemoryEnabled $true
18 Rename-VMNetworkAdapter -VMName $VMName -Name 'Network Adapter' -NewName 'LocalNetwork'
19
20 Get-VM -Name $VMName | Start-VM
21 }
22
23 New-LabVM -VMName 'AD-Full' -InstallType Full
24 New-LabVM -VMName 'Picard-Core' -InstallType Core
25 New-LabVM -VMName 'Riker-Core' -InstallType Core
26
27 $pw='vanh0uten142' | ConvertTo-SecureString -AsPlainText -force
28 $cred = New-Object System.Management.Automation.PSCredential ('SDF\Administrator',$pw)
29 Add-Computer -DomainName 'SDF.local' -Credential $cred -NewName RIKER
30
31 $pw='vanh0uten142' | ConvertTo-SecureString -AsPlainText -force
```

PS C:\windows\system32>

Ln 1 Col 1 90%

# Verb-Noun

Limited by  
Microsoft

Unlimited values,  
Should be  
descriptive

Get-Help

New-Item

Remove-Module

---

## Get-Command

List Commands:

\*New\*

-Module SQLPS/SqlServer

## Get-Help

Show help info:

man, help

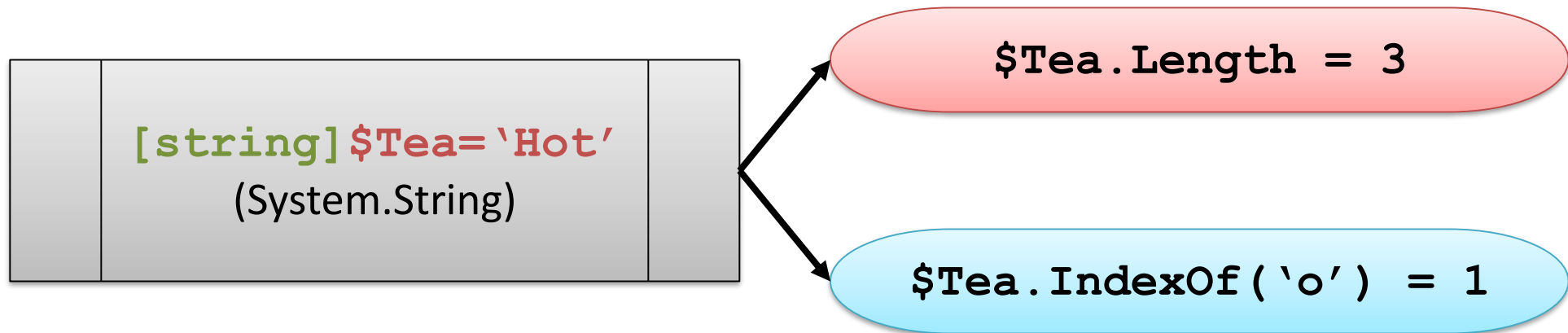
-ShowWindow

## Get-Member

Methods and  
properties

# Everything is a .Net object!

- Properties (attributes)
- Methods (functions, do things)





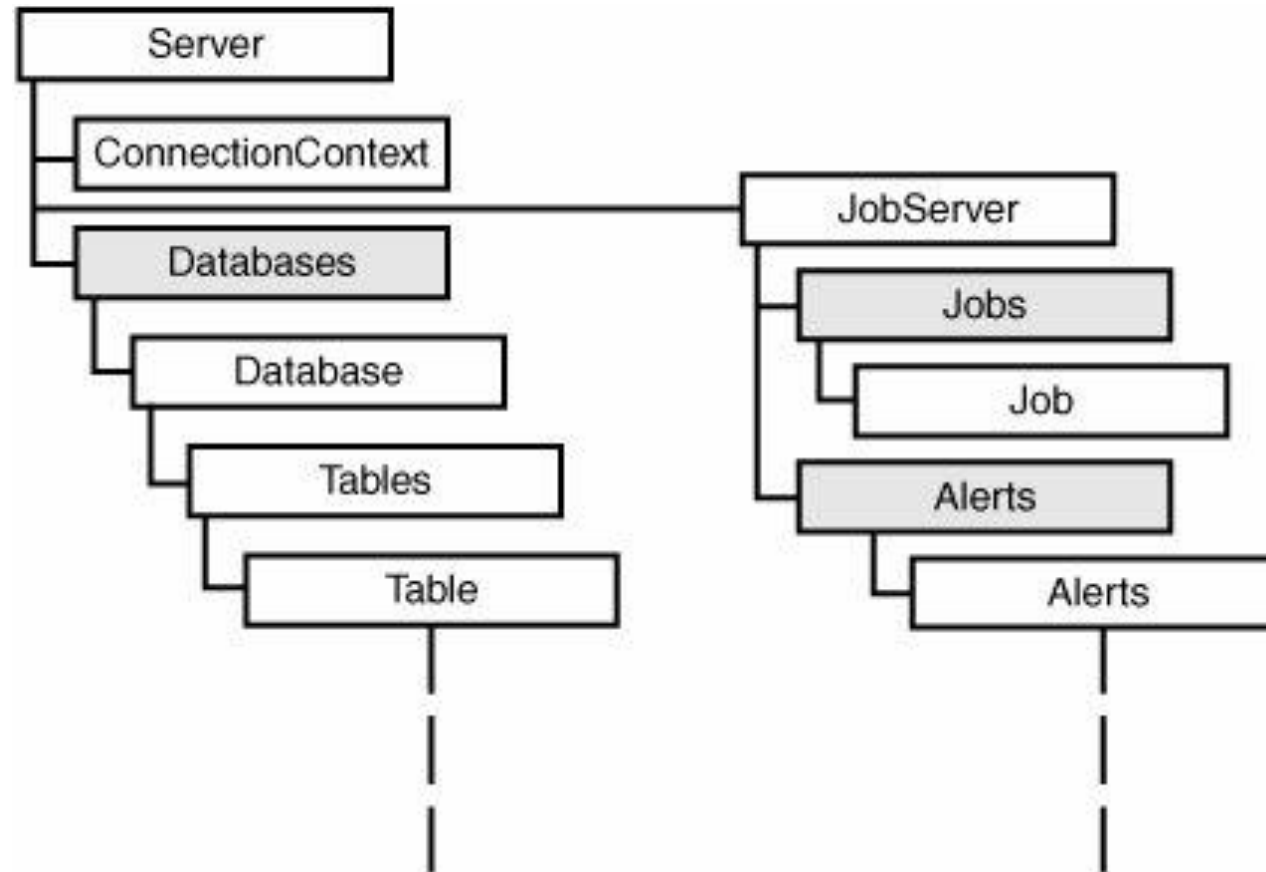
‘\$’ indicates a variable

```
[string]$Tea = 'Hot'
```

```
$Tea = 'Hot'
```

```
$TeaTime = Get-Date
```

---





**SQL Server cmdlets**  
**SQL Server provider**

---

# SQLPS

- Powershell module for SQL Server
- Includes cmdlets and the provider
- SQL Server 2012 or greater
- Powershell 2.0 or greater

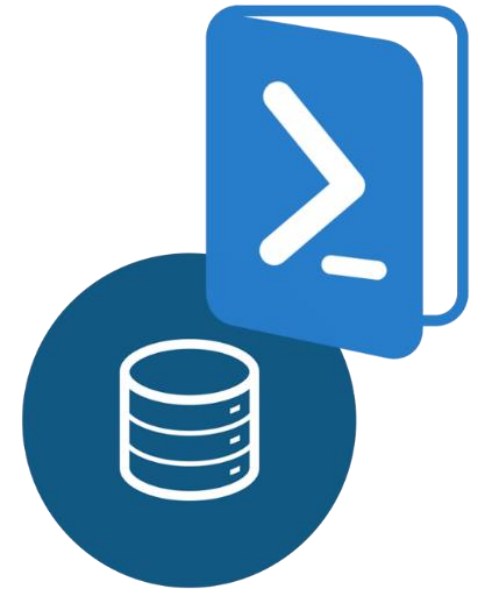
`Import-Module SQLPS`



# SqlServer

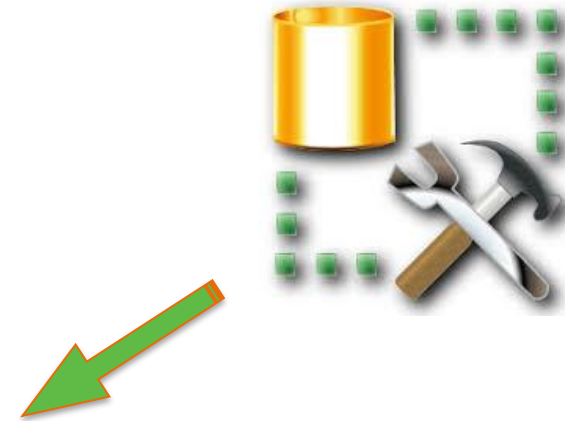
- Introduced with SSMS July 2016 update!
- Includes everything from SQLPS along with fixes
- Not compatible with SQLPS!

`Import-Module SqlServer`



- Not installed specifically
  - Included anytime you install ANY SQL Server component
  - SQLPS Installation Path:  
C:\Program Files (x86)  
  \Microsoft SQL Server  
  \**130**\Tools\PowerShell\Modules
  - SqlServer Installation Path  
C:\Program Files\WindowsPowerShell\Modules
-





```
PS SQLSERVER:\> Get-Command -Module SQLPS
```

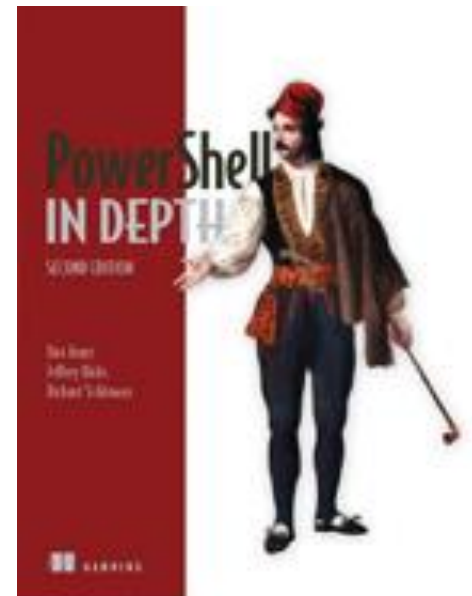
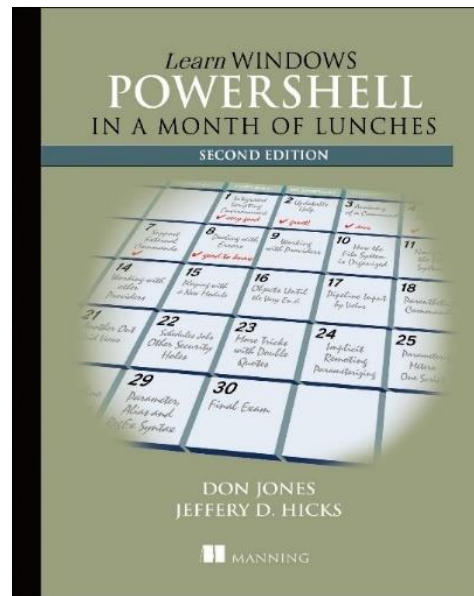
CommandType	Name	ModuleName
-----	----	-----
Function	SQLSERVER:	SQLPS
Cmdlet	Add-SqlAvailabilityDatabase	SQLPS
Cmdlet	Add-SqlAvailabilityGroupListenerStaticIp	SQLPS
Cmdlet	Add-SqlFirewallRule	SQLPS
Cmdlet	Backup-SqlDatabase	SQLPS
Cmdlet	Convert-UrnToPath	SQLPS
Cmdlet	Decode-SqlName	SQLPS
Cmdlet	Disable-SqlAlwaysOn	SQLPS
Cmdlet	Enable-SqlAlwaysOn	SQLPS
Cmdlet	Encode-SqlName	SQLPS
Cmdlet	Get-SqlCredential	SQLPS
Cmdlet	Get-SqlDatabase	SQLPS
Cmdlet	Get-SqlInstance	SQLPS
Cmdlet	Get-SqlSmartAdmin	SQLPS
Cmdlet	Invoke-PolicyEvaluation	SQLPS
Cmdlet	Invoke-Sqlcmd	SQLPS
Cmdlet	Join-SqlAvailabilityGroup	SQLPS
Cmdlet	New-SqlAvailabilityGroup	SQLPS
Cmdlet	New-SqlAvailabilityGroupListener	SQLPS
Cmdlet	New-SqlAvailabilityReplica	SQLPS
Cmdlet	New-SqlBackupEncryptionOption	SQLPS
Cmdlet	New-SqlCredential	SQLPS
Cmdlet	New-SqlHADREndpoint	SQLPS

So now what?



# Powershell In A Month of Lunches

## Powershell in Depth, 2<sup>nd</sup> Edition



The Scripting Guys

(<http://blogs.technet.com/b/heyscriptingguy/>)

Jeff Hicks

(<http://jdhitsolutions.com/blog/>)

Powershell.org

(<http://powershell.org/>)

---



Ben Miller

(<http://www.dbaduck.com/>)

Allen White

([http://sqlblog.com/blogs/allen white/](http://sqlblog.com/blogs/allen_white/))

Kendal Van Dyke

(<http://www.kendalvandyke.com/>)

Laerte Junior

(<https://www.simple-talk.com/author/laerte-junior/>)

Chrissy LeMaire

(<https://blog.netnerds.net/author/chrissy/> )

---

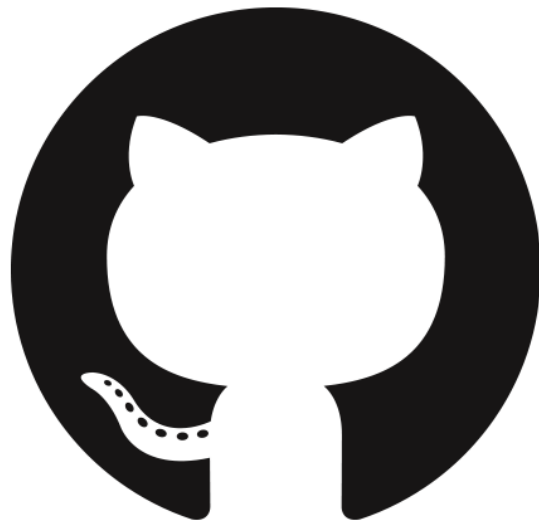


# Microsoft Virtual Academy PluralSight (\$\$)



<https://github.com/MikeFal>

- Powershell repository  
(all my scripts, including WIP)



Find tasks to automate  
Manage the file system ONLY through PowerShell  
Rewrite a T-SQL or other task using PowerShell  
And so on...

---



[mike@mikefal.net](mailto:mike@mikefal.net)



[www.mikefal.net](http://www.mikefal.net)



[@Mike\\_Fal](https://twitter.com/Mike_Fal)



<https://upsearch.com/connect-with-us/>



<https://github.com/MikeFal>