POWERSHELL JUMPSTART FOR SQL SERVER DBAS

www.mikefal.net

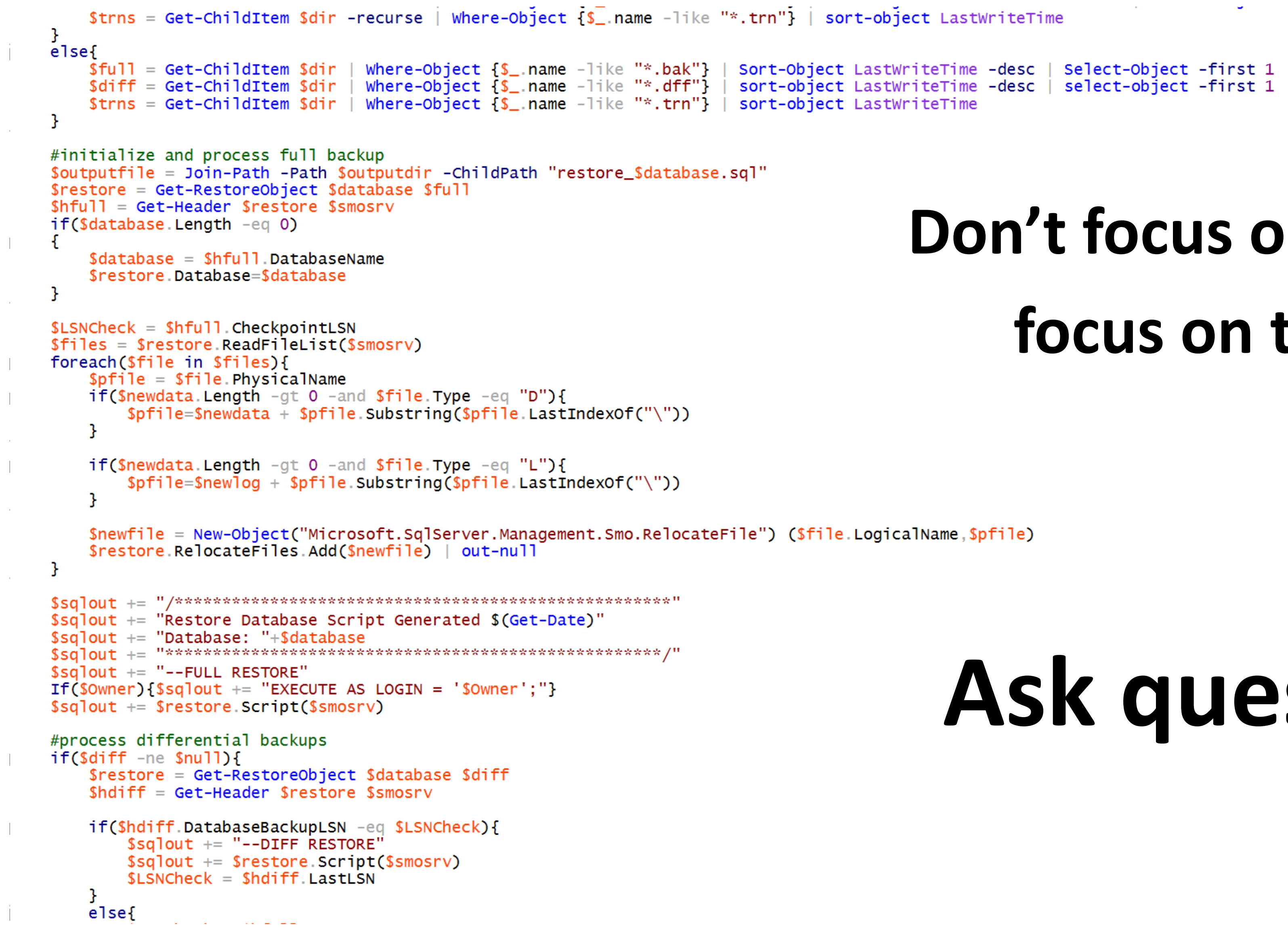
Mike Fal

Microsoft SQL Server

IDERA ACE

Microsoft CERTIFIED Solutions Expert Data Platform

Don’t Panic!



```
        $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
    $hfull = Get-Header $restore $smosrv
    if($database.Length -eq 0)
    {
        $database = $hfull.DatabaseName
        $restore.Database=$database
    }

    $LSNCheck = $hfull.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
    }

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

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

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

Don’t focus on the code,

focus on the concepts.

Ask questions!

Mike Fal - www.mikefal.net

Get-Agenda

The What and Why of Powershell

Language Basics

Working with SQL Server

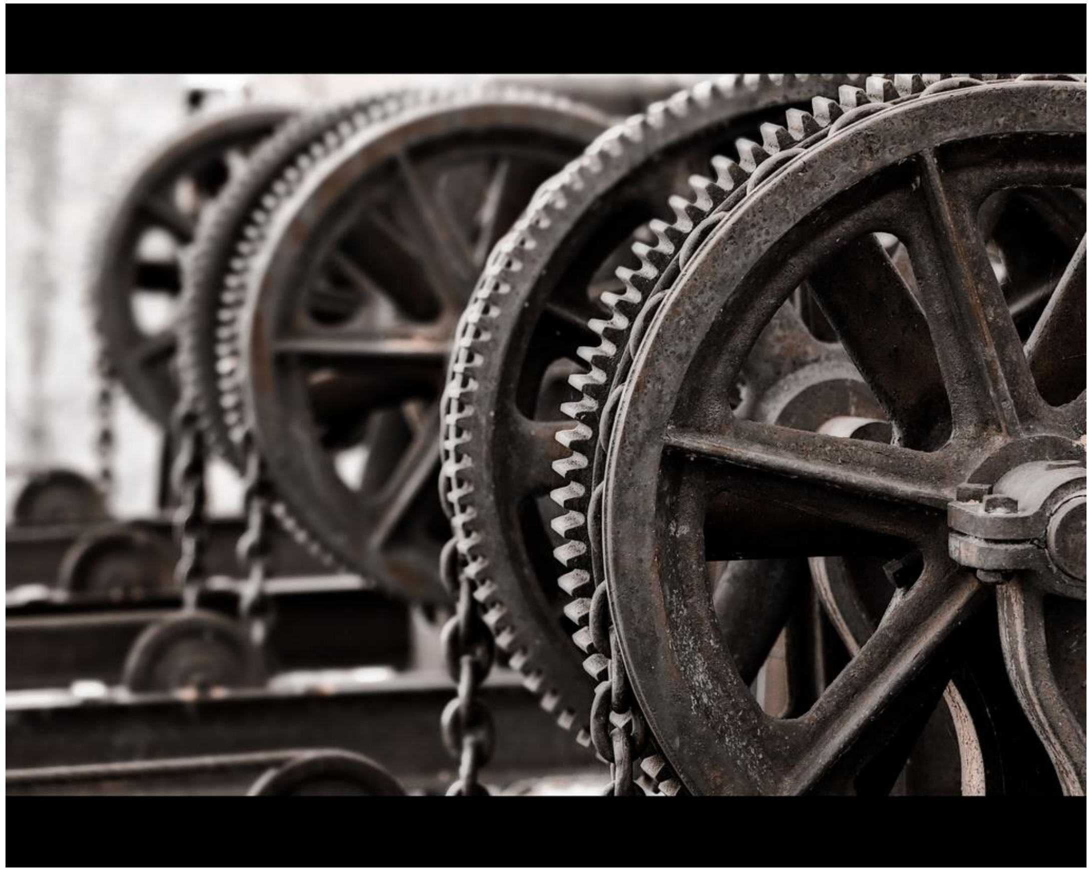
And Then What?

What is Powershell?

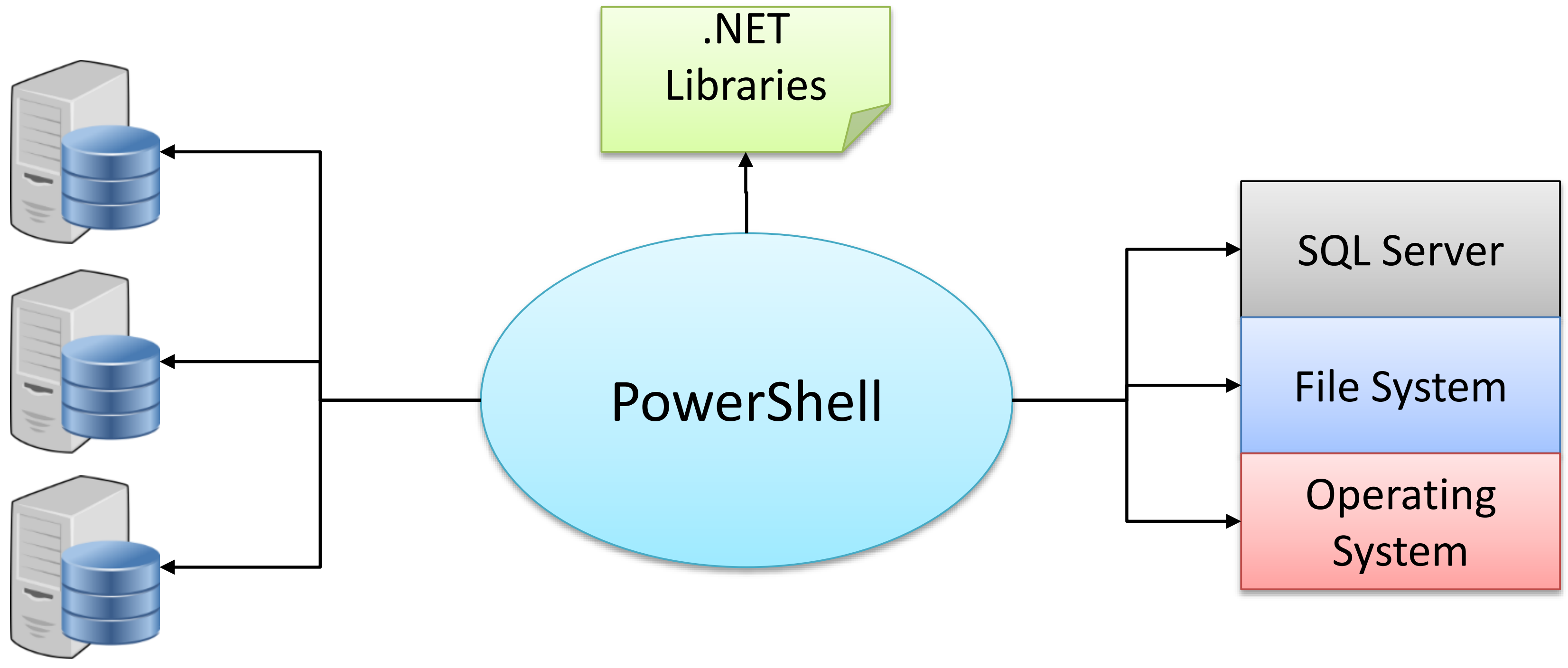
The Facts

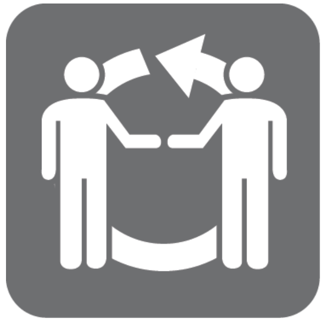
- Envisioned by Jeffery Snover – 2002
  - The Monad Manifesto
- Released as Powershell RC 1 – April 2006
  - Originally called Project “Monad”
- Current available version: 4.0
  - 5.0 is in preview!

But for what?



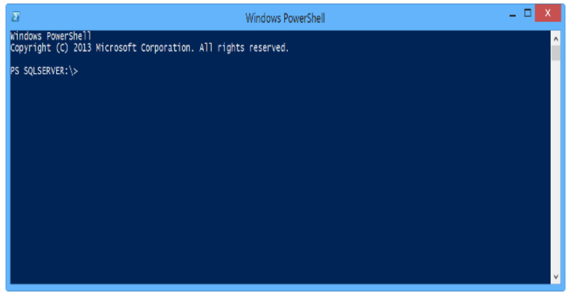
Why PowerShell?





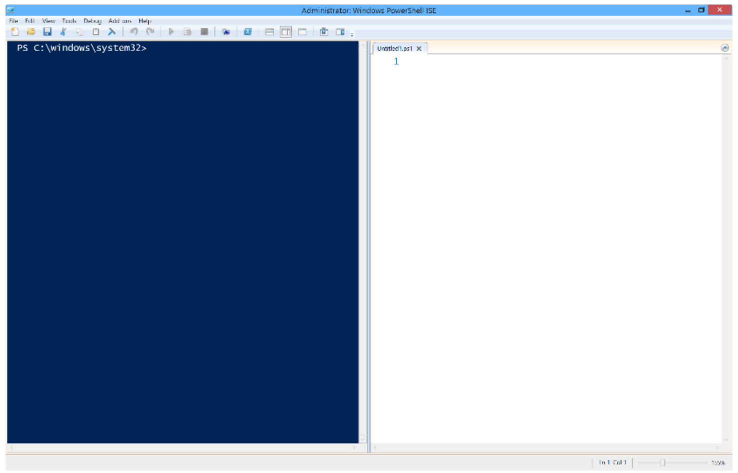
Pieces and Parts

Hosts



```
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS SQLSERVER:\>
```

Powershell.exe


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1

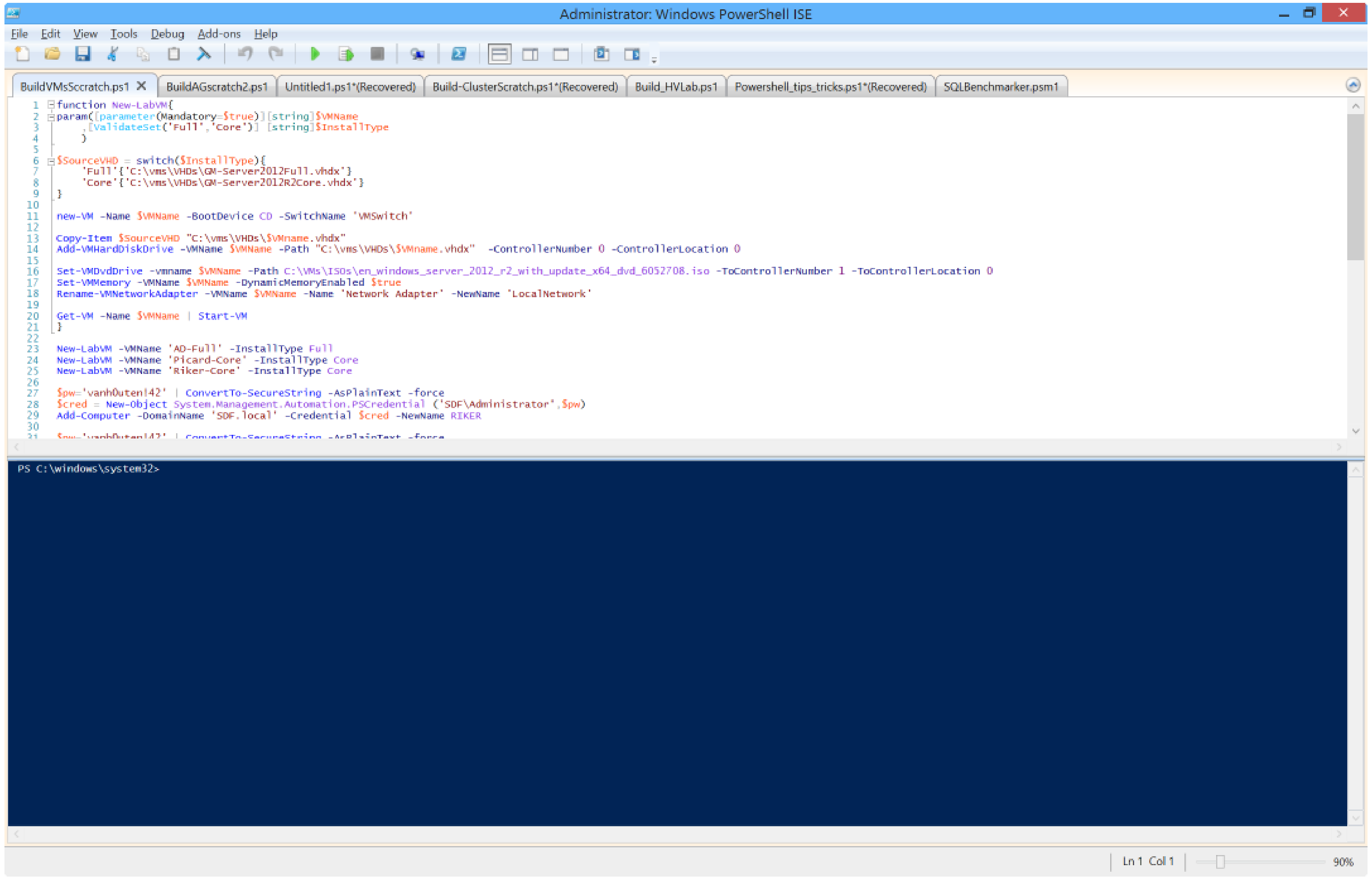
```
PS C:\windows\system32>
```

Powershell ISE

System.Management.Automation

Mike Fal - www.mikefal.net

Demo – The ISE


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
BuildVMsSccratch.ps1
BuildAGscratch2.ps1
Untitled1.ps1*(Recovered)
Build-ClusterScratch.ps1*(Recovered)
Build_HVLab.ps1
Powershell_tips_tricks.ps1*(Recovered)
SQLBenchmarker.psm1
Ln 1 Col 1
90%

```
function New-LabVM{
param([parameter(Mandatory=$true)][string]$VMName
     ,[ValidateSet('Full','Core')] [string]$InstallType
     )

$SourceVHD = switch($InstallType){
     'Full'{'C:\vms\VHDs\GM-Server2012Full.vhdx'}
     'Core'{'C:\vms\VHDs\GM-Server2012R2Core.vhdx'}
}

new-VM -Name $VMName -BootDevice CD -SwitchName 'VMSwitch'

Copy-Item $SourceVHD "C:\vms\VHDs\$VMname.vhdx"
Add-VMHardDiskDrive -VMName $VMName -Path "C:\vms\VHDs\$VMname.vhdx"  -ControllerNumber 0 -ControllerLocation 0

Set-VMDvdDrive -vmname $VMName -Path C:\VMs\ISOs\en_windows_server_2012_r2_with_update_x64_dvd_6052708.iso -ToControllerNumber 1 -ToControllerLocation 0
Set-VMMemory -VMName $VMName -DynamicMemoryEnabled $true
Rename-VMNetworkAdapter -VMName $VMName -Name 'Network Adapter' -NewName 'LocalNetwork'

Get-VM -Name $VMName | Start-VM
}

New-LabVM -VMName 'AD-Full' -InstallType Full
New-LabVM -VMName 'Picard-Core' -InstallType Core
New-LabVM -VMName 'Riker-Core' -InstallType Core

$pw='vanhOuten!42' | ConvertTo-SecureString -AsPlainText -force
$cred = New-Object System.Management.Automation.PSCredential ('SDF\Administrator',$pw)
Add-Computer -DomainName 'SDF.local' -Credential $cred -NewName RIKER

$pw='vanhOuten!42' | ConvertTo-SecureString -AsPlainText -force
```

```
PS C:\windows\system32>
```

cmdlets

Fundamental unit of “getting stuff done”

Verb-Noun

Limited by Microsoft

Unlimited values,
Should be descriptive

cmdlet examples

```
Get-Help
New-Item
Remove-Module
```

Learn Within Powershell

Get-Command

List Commands:
- *New*
- -Module SQLPS

Get-Help

Show help info:
- man, help
- -ShowWindow

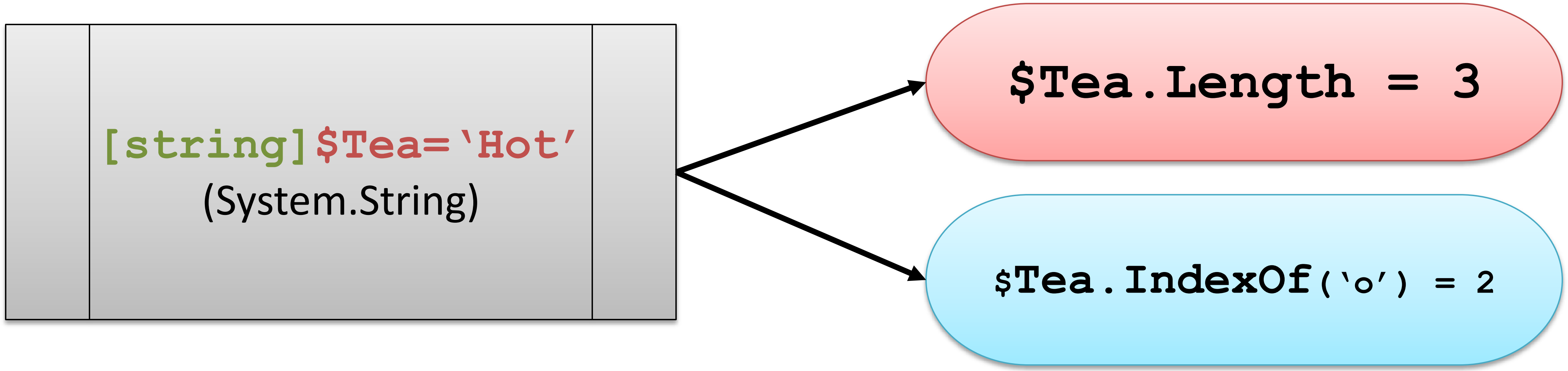
Get-Member

Methods and properties

Object Oriented Thinking

Everything is a .Net object!

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

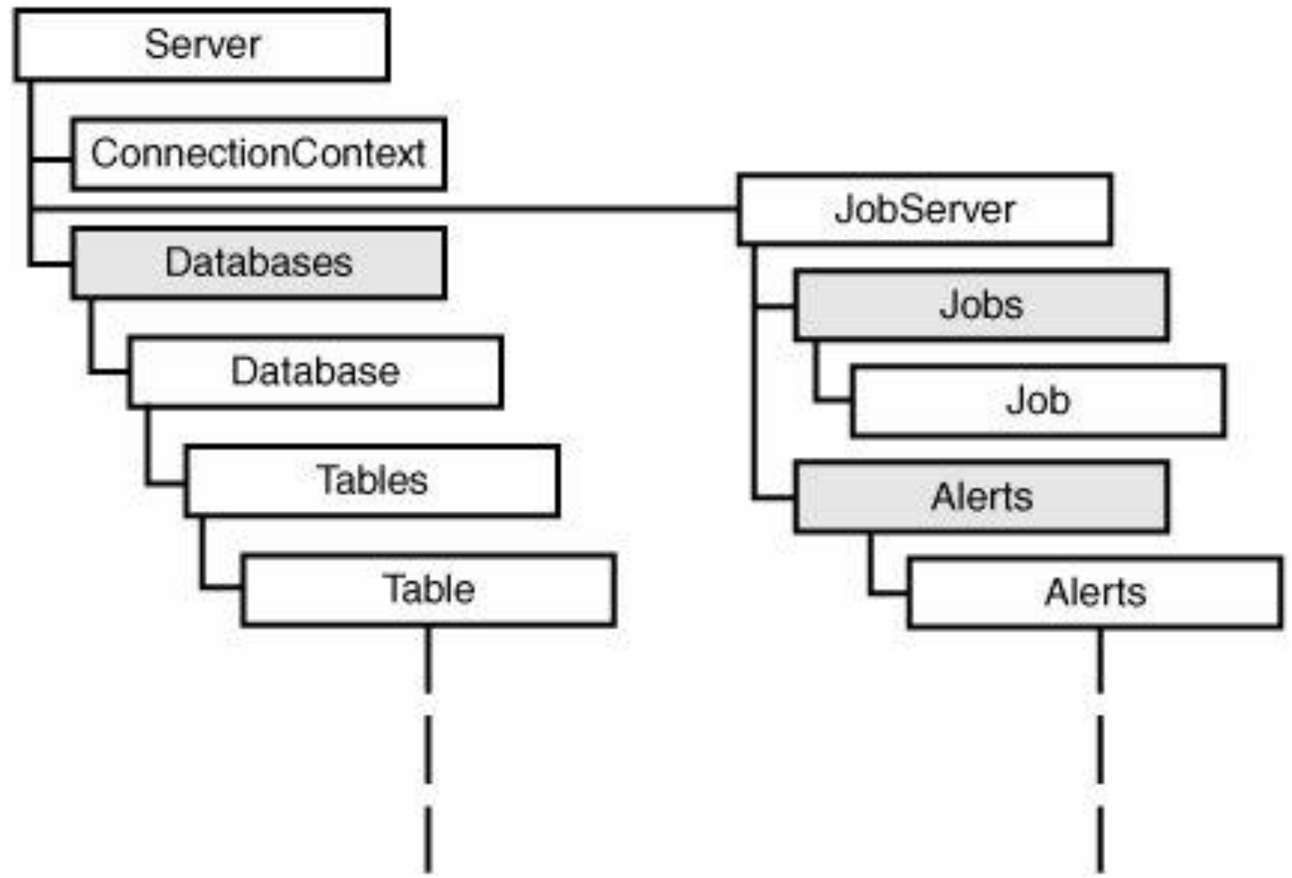


Variables

‘$’ indicates a variable

```
[string]$Tea = ‘Hot’
$Tea = ‘Hot’
$TeaTime = Get-Date
```

Demo – CMDLETS and Objects

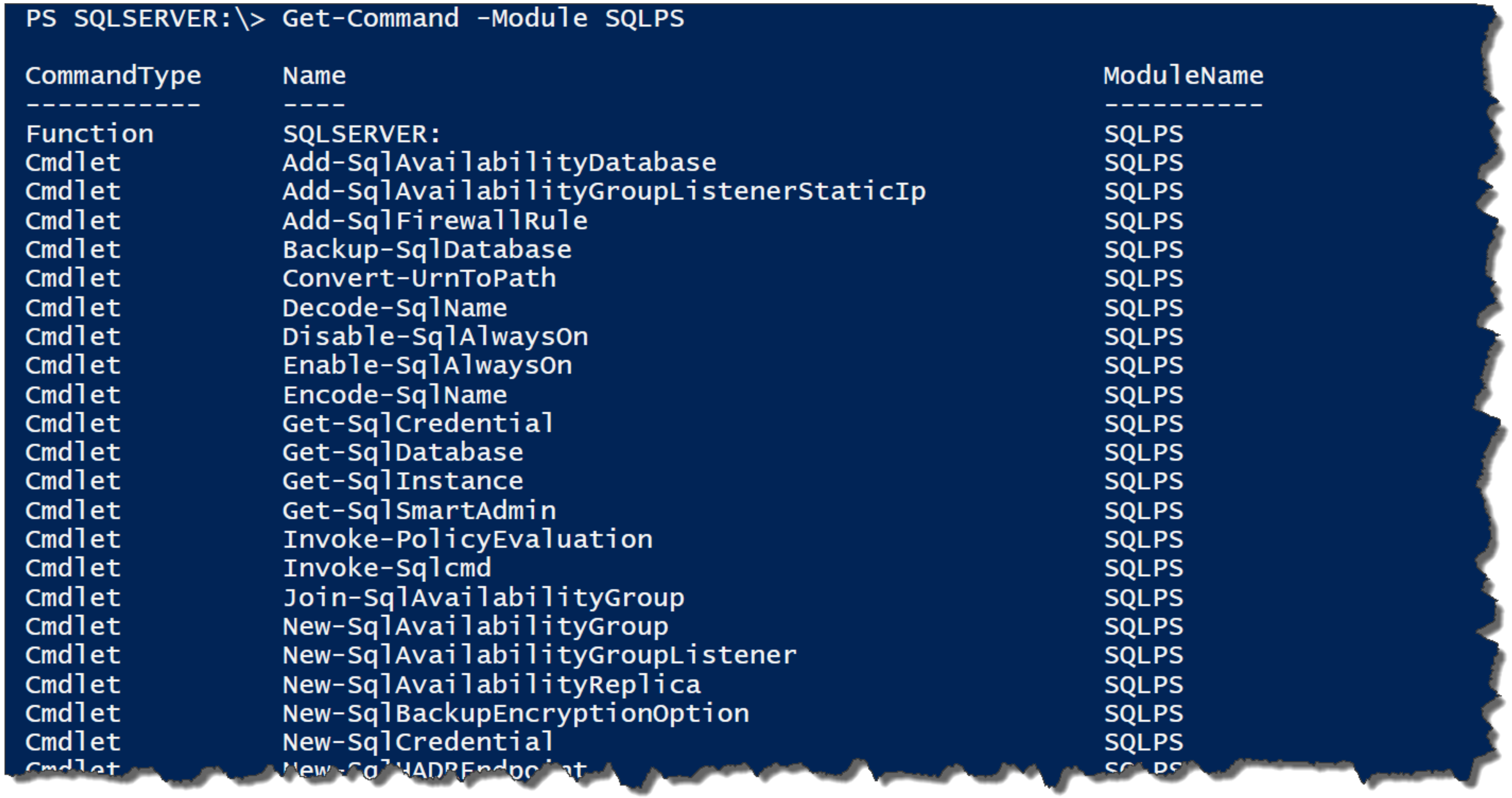


The SQL Server Module

`Import-Module SQLPS` (SQL 2012+ client)

SQL Server cmdlets **SQL Server Provider**

Demo – Working with SQL Server



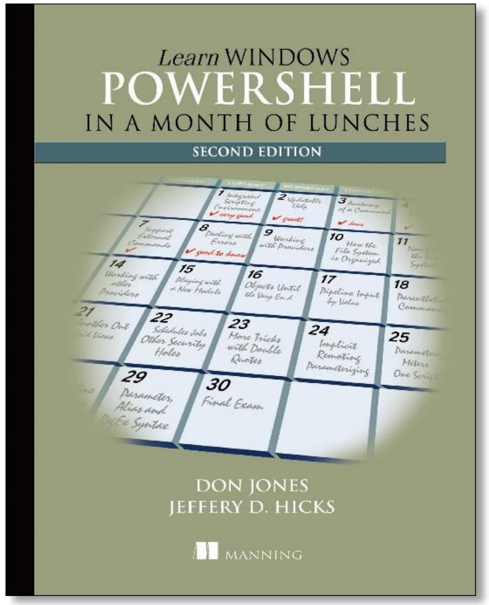
```
PS SQLSERVER:\> Get-Command -Module SQLPS

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
```

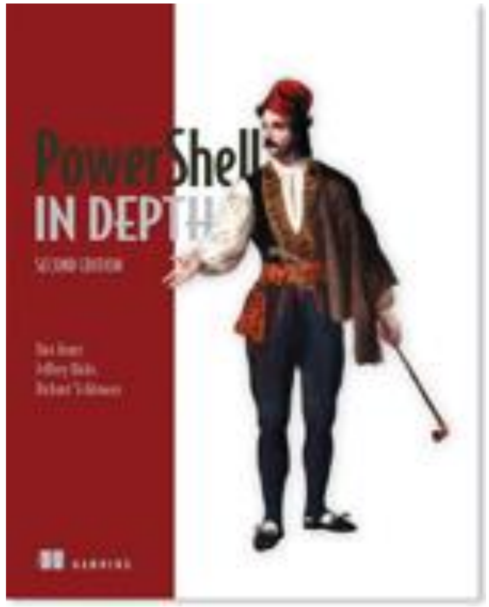
Mike Fal - www.mikefal.net

So now what?

Books

- Powershell In A Month of Lunches
- Powershell in Depth, 2nd Edition





General Powershell

- The Scripting Guys (http://blogs.technet.com/b/heyscriptingguy/)
- Jeff Hicks (http://jdhitsolutions.com/blog/)
- Powershell.org (http://powershell.org/)

Bloggers

- Ben Miller (http://www.dbaduck.com/)
- 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/)

Online

[Microsoft Virtual Academy]

[PluralSight ($$)]

Other scripts

- https://github.com/MikeFal
  - Powershell repository
    (all my scripts, including WIP)

Most of all, USE IT

Find tasks to automate

Manage the file system ONLY through Powershell

Rewrite a T-SQL or other task using Powershell

And so on…

Questions

mike@mikefal.net

www.mikefal.net

@Mike_Fal