PowerShell Basics

* **PowerShell is a framework to provide automation across the Windows platform.**
* **Cmdlets are the building blocks for PowerShell scripts**
* **Everything in PowerShell is a .NET object**

**Get-verb**

returns what verbs are available and gives you information to find certain commands

**3 most important cmdlets**

**Get-command – dictionary lookup this will help you find cmdlets you can use:**

Simply typing get-command will bring back all PS commands available on your computer

**Get-Command | Where-Object {$\_.source –eq ‘SQLPS’}**

This command will return all the commands in the sqlps module

**Get-command –name \*new\***

This will returns a list of commands that have verb like ‘new’

**Get-Help – help information (similar to ‘man’ pages in UNIX)**

**Get-help –name get-command -full**

This provides help information on the get-command cmdlet.

**Get-help –name get-command –examples**

Provides examples on using the cmdlet

**Get-help –name get-command -showwindow**

This provides the help information in a separate window

**Get-help get-command –online**

Will provide technet webpage on the command

**Get-Member – explores methods and properties within powershell**

Other

**Get-childitem C:\**

Returns all contents of a directory

Alias – You can have aliases for cmdlets too

Dir is an alias of Get-Childitem

Sal is an alias of set-alias

**Get-alias –defintion get-childitem**

This returns all aliases for get-childitem

Variables

Variables all start with $

Variables are based on .NET objects. PowerShell variables are strongly typed.

Objects are more than just data. There are Functions that can be used by the object.

PS C:\Users\pepperg> $tea = 'hot'

PS C:\Users\pepperg> $tea.Length

3

PS C:\Users\pepperg> $tea.IndexOf('o')

1

Get-Member can be used to find information on our objects

$tea |Get-Member

$file = New-Item –itemType File –Path C:\Temp\JunkFile.txt’

$File | Get-Member

$file.name

$file.FullName

$file.extension

$file.lastwritetime

Remove-item $file

Pipelines

$string = “Earl Grey, Hot.”

$string | get-member

Get-help about\* | measure-object

PS C:\Users\pepperg> get-help about\* | Measure-Object

Count : 117

Average :

Sum :

Maximum :

Minimum :

Property :

You could pipe a string of text to a text file (good for creating error logs)

The below example creates a new ‘test’ directory and a file in here to add a string too.

PS C:\Users\pepperg> new-item -ItemType directory -Path C:\Test

Directory: C:\

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 26/10/2018 20:24 Test

PS C:\Users\pepperg> 'The Quick brown fox jumps over the lazy dog.' | out-file -FilePath 'C:\test\Dummy.txt'

PS C:\Users\pepperg> notepad 'C:\Test\Dummy.txt'

The below command appends data to a file

PS C:\Users\pepperg> 'AHHHHHHH' | Out-File -FilePath C:\Test\Dummy.txt -Append

You can delete all items in a folder using a pipeline

PS C:\Users\pepperg> Dir C:\Test | Remove-Item

The following line could be used to delete old transaction logs older than 3 hours

PS C:\Users\pepperg> Get-ChildItem '\\Picard\C$\Backups' -recurse | where-object {$\_.Extension -eq ".trn" -and $\_.LastWriteTime -lt (get-date).AddHours(-3)} | remove-item -Whatif

-recurse will list all sub-folders and items from a folder

-whatif shows you what the results are without performing the action. Handy for testing file deletions

# Administering SQL Server

SQLPS – The SQL Server PowerShell Module

2 areas of functionality – SQL server Provider and CmdLets

You need to import module

PS C:\Users\pepperg> Import-Module sqlps

SQLPS is being deprecated.

SQLPS gets installed with SQL Server by default

What’s in SQLPS

PS C:\Users\pepperg> get-command -Module sqlps

CommandType Name Version S

o

u

r

c

e

----------- ---- ------- -

Alias Decode-SqlName 1.0 s

Alias Encode-SqlName 1.0 s

Function SQLSERVER: 1.0 s

Cmdlet Add-SqlAvailabilityDatabase 1.0 s

Cmdlet Add-SqlAvailabilityGroupListenerStaticIp 1.0 s

Cmdlet Add-SqlFirewallRule 1.0 s

Cmdlet Backup-SqlDatabase 1.0 s

Cmdlet ConvertFrom-EncodedSqlName 1.0 s

Cmdlet ConvertTo-EncodedSqlName 1.0 s

Cmdlet Convert-UrnToPath 1.0 s

Cmdlet Disable-SqlAlwaysOn 1.0 s

Cmdlet Enable-SqlAlwaysOn 1.0 s

Cmdlet Get-SqlCredential 1.0 s

Cmdlet Get-SqlDatabase 1.0 s

Cmdlet Get-SqlInstance 1.0 s

Cmdlet Get-SqlSmartAdmin 1.0 s

Cmdlet Invoke-PolicyEvaluation 1.0 s

Cmdlet Invoke-Sqlcmd 1.0 s

Cmdlet Join-SqlAvailabilityGroup 1.0 s

Cmdlet New-SqlAvailabilityGroup 1.0 s

Cmdlet New-SqlAvailabilityGroupListener 1.0 s

Cmdlet New-SqlAvailabilityReplica 1.0 s

Cmdlet New-SqlBackupEncryptionOption 1.0 s

Cmdlet New-SqlCredential 1.0 s

Cmdlet New-SqlHADREndpoint 1.0 s

Cmdlet Remove-SqlAvailabilityDatabase 1.0 s

Cmdlet Remove-SqlAvailabilityGroup 1.0 s

Cmdlet Remove-SqlAvailabilityReplica 1.0 s

Cmdlet Remove-SqlCredential 1.0 s

Cmdlet Remove-SqlFirewallRule 1.0 s

Cmdlet Restore-SqlDatabase 1.0 s

Cmdlet Resume-SqlAvailabilityDatabase 1.0 s

Cmdlet Save-SqlMigrationReport 1.0 s

Cmdlet Set-SqlAuthenticationMode 1.0 s

Cmdlet Set-SqlAvailabilityGroup 1.0 s

Cmdlet Set-SqlAvailabilityGroupListener 1.0 s

Cmdlet Set-SqlAvailabilityReplica 1.0 s

Cmdlet Set-SqlCredential 1.0 s

Cmdlet Set-SqlHADREndpoint 1.0 s

Cmdlet Set-SqlNetworkConfiguration 1.0 s

Cmdlet Set-SqlSmartAdmin 1.0 s

Cmdlet Start-SqlInstance 1.0 s

Cmdlet Stop-SqlInstance 1.0 s

Cmdlet Suspend-SqlAvailabilityDatabase 1.0 s

Cmdlet Switch-SqlAvailabilityGroup 1.0 s

Cmdlet Test-SqlAvailabilityGroup 1.0 s

Cmdlet Test-SqlAvailabilityReplica 1.0 s

Cmdlet Test-SqlDatabaseReplicaState 1.0 s

Cmdlet Test-SqlSmartAdmin 1.0 s

Prior to SQL2016 there are two cmdlets that don’t meet the naming standard (not approved verbs) .

Also prior to SQL 2016, loading the module would automatically change your location to the SQLSERVER:\ provider ‘drive’

## SQL Provider

Providers expose windows components as if they are files and folders

Get-PSDrive – Lists your providers.

PS SQLSERVER:\> Get-PSDrive

Name Used (GB) Free (GB) Provider Root

---- --------- --------- -------- ----

Alias Alias

C 117.13 1.62 FileSystem C:\

Cert Certificate \

Env Environment

Function Function

HKCU Registry HKEY\_CURRENT\_USER

HKLM Registry HKEY\_LOCAL\_MACHINE

SQLSERVER SqlServer SQLSERVER:\

Variable Variable

WSMan WSMan

Below you can change to the SQL Provider on the command line.

PS C:\Users\pepperg> CD SQLSERVER:\

PS SQLSERVER:\>

[DV-SQLCLN-02]: PS SQLSERVER:\> cd SQL

[DV-SQLCLN-02]: PS SQLSERVER:\SQL> ls

MachineName

-----------

DV-SQLCLN-02

[DV-SQLCLN-02]: PS SQLSERVER:\SQL> cd DV-SQLCLN-02

[DV-SQLCLN-02]: PS SQLSERVER:\SQL\DV-SQLCLN-02> dir

Instance Name

-------------

DEFAULT

List databases

Dir databases | format-table –Autosize

[DV-SQLCLN-02]: PS SQLSERVER:\SQL\DV-SQLCLN-02> cd default

[DV-SQLCLN-02]: PS SQLSERVER:\SQL\DV-SQLCLN-02\default> Dir databases | format-table –Autosize

Name Status Containment Type Recovery Model CompatLvl Collation

---- ------ ---------------- -------------- --------- ---------

AdventureWorks2012 Normal None Full 110 SQL\_Latin1\_

General\_CP1

\_CI\_AS

MuckleDB Normal None Full 110 Latin1\_Gene

ral\_CI\_AS

StackOverflow2010 Normal None Full 110 SQL\_Latin1\_

General\_CP1

\_CS\_AS

Tools Normal None Full 130 Latin1\_Gene

ral\_CI\_AS

list tables in a database:

[DV-SQLCLN-02]: PS SQLSERVER:\SQL\DV-SQLCLN-02\default> dir databases\muckledb\tables

Schema Name Created

------ ---- -------

dbo ACC\_TRANSACTION 18/06/2018 11:12

dbo ACCOUNT 18/06/2018 11:12

dbo BRANCH 18/06/2018 11:12

dbo BUSINESS 18/06/2018 11:12

dbo CUSTOMER 18/06/2018 11:12

dbo DEPARTMENT 18/06/2018 11:12

dbo EMPLOYEE 18/06/2018 11:12

dbo INDIVIDUAL 18/06/2018 11:12

dbo OFFICER 18/06/2018 11:12

dbo PRODUCT 18/06/2018 11:12

dbo PRODUCT\_TYPE 18/06/2018 11:12

Check for last database backups

$instances = @(‘Default’) #this array can allow you to look at multiple instances

$instances | foreach-object {get-childItem “SQLSERVER:\SQL\$\_\Default\Databases” –force} | sort-object size –descending | select-object @{n=’Server’:e={$\_.parent.name}}.Name.lastbackupdate.size

[DV-SQLCLN-02]: PS> $instances = @(‘Default’)

$instances | foreach-object {get-childItem “SQLSERVER:\SQL\$\_\Default\Databases” –force} | sort-object size –descending | select-object @{n=’Server’;e={$\_.parent.name}}.Name.lastbackupdate.size

## Cmdlets

Focus areas:

General purpose (working with basics of SQL Server)

Invoke-sqlcmd – allows you to run t-sql scripts on an instance

Cloud/azure administration (not used as much as others)

Backups/Restores – very useful

The cmdlets such as Backup-SQLDatabase is simply running T-SQL in the background

PS SQLSERVER:\> Backup-SqlDatabase –Serverinstance DV-SQLCLN-02 –Database MuckleDB –BackupFile ‘C:\Temp\MuckleDB.bak’ –whatif

What if: Performing the operation "Backup-SqlDatabase" on target "[DV-SQLCLN-02]

".

A good example for this would be to set this command to run on multiple instances and backup to the same location. If the folder didn’t exist (test-path), you could create it (new-item).

Availability Group Management (advanced feature)

## SQL Server Management Objects

SMO has been around since 2002

It is the .NET library for accessing SQL Server.

When you load SQLPS module you automatically get lots of SMO libraries loaded with it.

You can interact with a server as if it is an object:

PS SQLSERVER:\> $SqlInstance = new-object Microsoft.SqlServer.Management.Smo.Server 'DV-SQLCLN-01'

PS SQLSERVER:\> $SqlInstance.databases

Name

----

AdventureWorks2012

master

MockData

model

msdb

MuckleDB

StackOverflow2010

tempdb

Tools

You can drill down to great detail:

PS SQLSERVER:\> $sysjobs = $SqlInstance.databases["msdb"].tables["sysjobs"]

PS SQLSERVER:\> $sysjobs | Get-Member

You could script out all your logins for a server

PS SQLSERVER:\> $SqlInstance.Logins.script()

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [##MS\_PolicyEventProcessingLogin##] WITH PASSWORD=N'qA0YZV/7TKOHD34

TB1pkZ7I+Fm8bIiqXhiSPaiBKwtY=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_

english], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [##MS\_PolicyEventProcessingLogin##] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [##MS\_PolicyTsqlExecutionLogin##] WITH PASSWORD=N'/CZZ7QIrO+GeNTERW

6PBtLRpS5aPUTSniDqVhXOBapA=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_en

glish], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [##MS\_PolicyTsqlExecutionLogin##] DISABLE

CREATE LOGIN [BHF-ADS\adminlg] FROM WINDOWS WITH DEFAULT\_DATABASE=[master], DEFA

ULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [BHF-ADS\adminlg]

CREATE LOGIN [BHF-ADS\Domain Admins] FROM WINDOWS WITH DEFAULT\_DATABASE=[master]

, DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [BHF-ADS\Domain Admins]

CREATE LOGIN [BHF-ADS\pepperg] FROM WINDOWS WITH DEFAULT\_DATABASE=[master], DEFA

ULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [BHF-ADS\pepperg]

CREATE LOGIN [BHF-ADS\RES-DBS-KILO-BHFSTAG-ROA] FROM WINDOWS WITH DEFAULT\_DATABA

SE=[master], DEFAULT\_LANGUAGE=[us\_english]

CREATE LOGIN [BHF-ADS\ROLE-Database Administrators] FROM WINDOWS WITH DEFAULT\_DA

TABASE=[master], DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [BHF-ADS\ROLE-Database Administrators]

CREATE LOGIN [BHF-ADS\Run\_SQL\_PShell] FROM WINDOWS WITH DEFAULT\_DATABASE=[master

], DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [BHF-ADS\Run\_SQL\_PShell]

CREATE LOGIN [bhf-ads\svcCL\_DEV\_SQL\_1\_SQL] FROM WINDOWS WITH DEFAULT\_DATABASE=[m

aster], DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [bhf-ads\svcCL\_DEV\_SQL\_1\_SQL]

CREATE LOGIN [BHF-ADS\svcPD-VEEGUS-01] FROM WINDOWS WITH DEFAULT\_DATABASE=[maste

r], DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [dbcreator] ADD MEMBER [BHF-ADS\svcPD-VEEGUS-01]

CREATE LOGIN [BHF-ADS\TableauAdmin] FROM WINDOWS WITH DEFAULT\_DATABASE=[master],

DEFAULT\_LANGUAGE=[us\_english]

CREATE LOGIN [BHF-ADS\Team-HO IT SQL Server Developers] FROM WINDOWS WITH DEFAUL

T\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english]

CREATE LOGIN [BHF-ADS\younisa] FROM WINDOWS WITH DEFAULT\_DATABASE=[master], DEFA

ULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [BHF-ADS\younisa]

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [BHFDataDW\_Load] WITH PASSWORD=N'+jhxQVWM7d+0VPmN4DzAdPbbV7vlttwXFi

AZOeQxA8k=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_EXP

IRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [BHFDataDW\_Load] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [BHFDataDW\_Reporting] WITH PASSWORD=N'jb61wNTAMUyqSK7REse2fU9bQ0okW

4z7hvt6fDJya44=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHEC

K\_EXPIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [BHFDataDW\_Reporting] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [BHFDataDW\_Staging] WITH PASSWORD=N'zYhxraILT40p7FMIflTXBhcbzyNyLXR

WJPxu5Smr1ZA=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_

EXPIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [BHFDataDW\_Staging] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [BHFStagingAdmin] WITH PASSWORD=N'asEPIpS8XmYpuWtIAxqzGXsxRFEGY9Wp9

9W7paGP4R0=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_EX

PIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [BHFStagingAdmin] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [BHFStagingHRProFSS] WITH PASSWORD=N'zIk07OTWArRuDgtIMblNyJfg2ai3wF

IBkARumfdHH7g=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK

\_EXPIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [BHFStagingHRProFSS] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [BHFStagingIntel] WITH PASSWORD=N'mW+iXCqb5sNukpD98EmG2EhEPSpLTtZ6R

Ud4D6DigBE=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_EX

PIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [BHFStagingIntel] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [BHFStagingLoad] WITH PASSWORD=N'KhjSP176hhOpA5KTRF/DFBa3t3k9aqPlPs

IFPNsP5TE=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_EXP

IRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [BHFStagingLoad] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [C4rb0n] WITH PASSWORD=N'+r10Hn+tgZLzP8FsUC5MTmnMrHDPrkEE5VZoY6KzqK

E=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_EXPIRATION=

OFF, CHECK\_POLICY=ON

ALTER LOGIN [C4rb0n] DISABLE

ALTER SERVER ROLE [sysadmin] ADD MEMBER [C4rb0n]

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [EventIncomeReport] WITH PASSWORD=N'fkVv2xByCLz40AyoLeV5lkIexrAdO2Q

cBofu11pRMqE=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_

EXPIRATION=OFF, CHECK\_POLICY=ON

ALTER LOGIN [EventIncomeReport] DISABLE

CREATE LOGIN [NT AUTHORITY\SYSTEM] FROM WINDOWS WITH DEFAULT\_DATABASE=[master],

DEFAULT\_LANGUAGE=[us\_english]

CREATE LOGIN [NT SERVICE\MSSQLSERVER] FROM WINDOWS WITH DEFAULT\_DATABASE=[master

], DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT SERVICE\MSSQLSERVER]

CREATE LOGIN [NT SERVICE\SQLSERVERAGENT] FROM WINDOWS WITH DEFAULT\_DATABASE=[mas

ter], DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT SERVICE\SQLSERVERAGENT]

CREATE LOGIN [NT SERVICE\SQLTELEMETRY] FROM WINDOWS WITH DEFAULT\_DATABASE=[maste

r], DEFAULT\_LANGUAGE=[us\_english]

CREATE LOGIN [NT SERVICE\SQLWriter] FROM WINDOWS WITH DEFAULT\_DATABASE=[master],

DEFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT SERVICE\SQLWriter]

CREATE LOGIN [NT SERVICE\Winmgmt] FROM WINDOWS WITH DEFAULT\_DATABASE=[master], D

EFAULT\_LANGUAGE=[us\_english]

ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT SERVICE\Winmgmt]

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [SVCStackOverflow] WITH PASSWORD=N'dfeObb7pIYxQjbLqNPj0X6cu3ftfQ6GT

rj5kmAht2js=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_E

XPIRATION=OFF, CHECK\_POLICY=OFF

ALTER LOGIN [SVCStackOverflow] DISABLE

/\* For security reasons the login is created disabled and with a random password

. \*/

CREATE LOGIN [ToolsDBO] WITH PASSWORD=N'0FRbWhfsM532mz87b5XF3NwkPF7K+aOPQNq30MOE

PjA=', DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english], CHECK\_EXPIRATIO

N=OFF, CHECK\_POLICY=ON

ALTER LOGIN [ToolsDBO] DISABLE

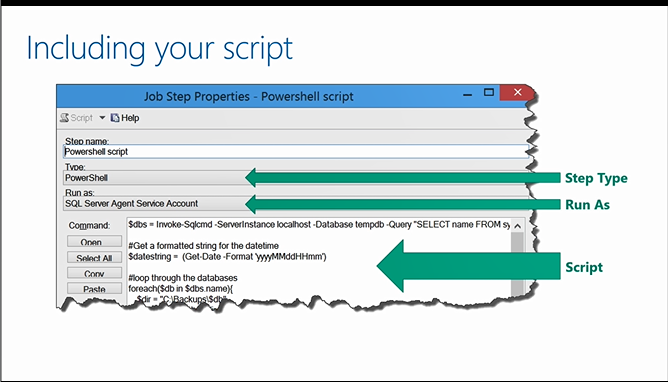
By adding an out-file pipe command you could easily append this to an SQL File.

You can create/drop databases with SMOs quite easily. Not only can this but you set filegrowths, names, locations quite easily.

T-SQL gives you more control over this but scripting this in PowerShell can be really useful for automation.

## SQL Jobs and PowerShell

SQL Server 2008, R2 and 2012 only uses PowerShell 2.0.

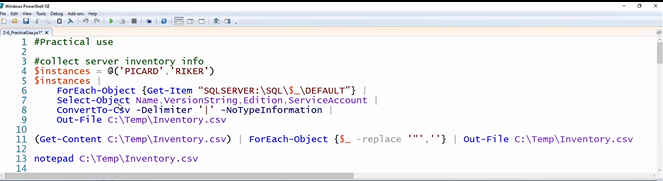


The account that runs the script (default to the Agent account) needs permissions to run the PSScript. For security purposes, it is best practice to set up a credential.

Practical uses include backup/restore scripts, moving backup files (might be handy for UAT restores).

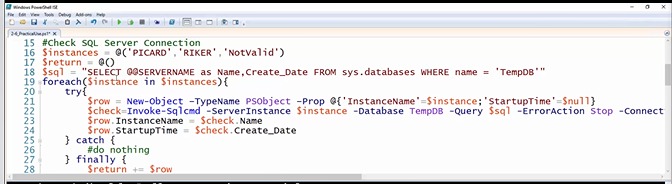
## Practical Uses

**Get SQL server Inventory Info through the SMO**



**Testing your SQL Server Connection**

If you query tempdb and look at create date, you get the date of your last SQL Server restart.



Create the script above and include this as part of my PowerShell profile

**Create a Point in Time Restore Script**

This is good for when you have lots of log backups that you need to restore. You can sort the output and use a foreach loop to restore all the log backups. You can also create a file with the SQL Statement on for the Point in Time restore in case you want to run the T-SQL separately and/or tweak it and use it in the future.

