EXPLORING SQLPS

The SQL Server Powershell Module In Depth

UpSearch

Protect, unlock and optimize your data's value

Thank you, sponsors!

UpSearch

Microsoft® SQL Server®

www.mikefal.net

IDERA ACE

Microsoft CERTIFIED Solutions Expert

Data Platform

Mike Fal

UPSEARCH

What is SQLPS?

The Provider

Cmdlets/Functions

Practical Use

Don't Panic!

UPSEARCH

```
    $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!

In the beginning...

UpSearch

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
```

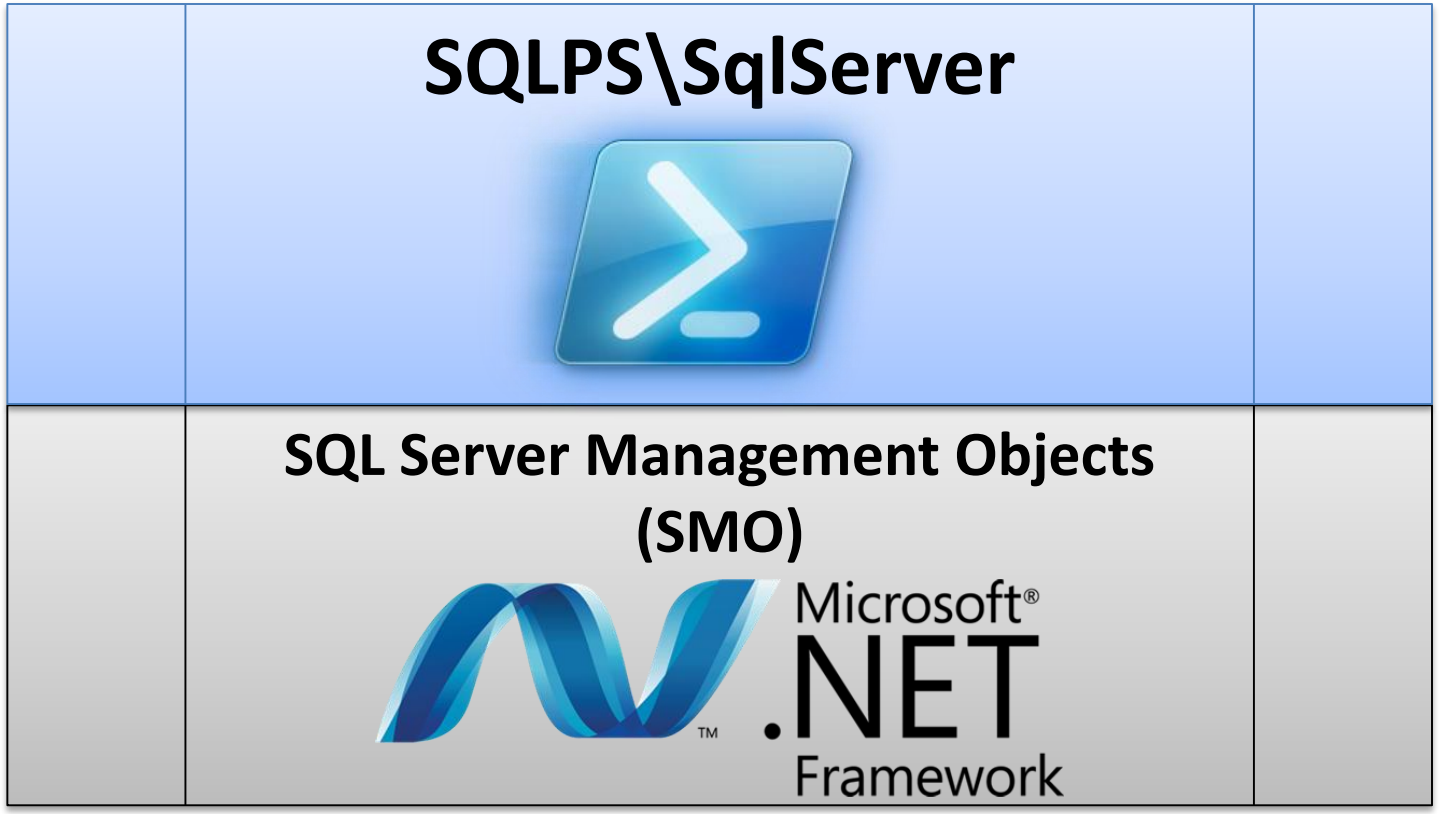
UPSEARCH

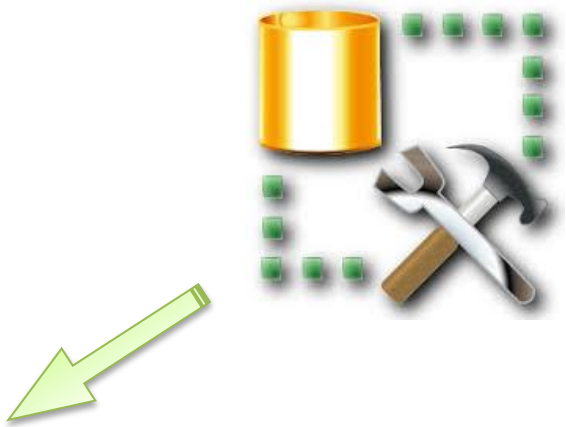
- Not installed specifically
- Included anytime you install ANY SQL Server component
- SQLPS Installation Path:

```
C:\Program Files (x86)
   \Microsoft SQL Server
   \130\Tools\PowerShell\Modules
```

- SqlServer Installation Path

```
C:\Program Files\WindowsPowerShell\Modules
```





.Net Library

requires .Net 2.0

UPSEARCH

SMO ASSEMBLIES	
Microsoft.SqlServer.Management.Common	Microsoft.SqlServer.SqlEnum
Microsoft.SqlServer.Smo	Microsoft.SqlServer.RegSvrEnum
Microsoft.SqlServer.Dmf	Microsoft.SqlServer.WmiEnum
Microsoft.SqlServer.Instapi	Microsoft.SqlServer.ServiceBrokerEnum
Microsoft.SqlServer.SqlWmiManagement	Microsoft.SqlServer.ConnectionInfoExtended
Microsoft.SqlServer.ConnectionInfo	Microsoft.SqlServer.Management.Collector
Microsoft.SqlServer.SmoExtended	Microsoft.SqlServer.Management.CollectorEnum
Microsoft.SqlServer.SqlTDiagM	Microsoft.SqlServer.Management.Dac
Microsoft.SqlServer.SString	Microsoft.SqlServer.Management.DacEnum
Microsoft.SqlServer.Management.RegisteredServers	Microsoft.SqlServer.Management.Utility
Microsoft.SqlServer.Management.Sdk.Sfc	

DEMO!


Breeding
ken Problems
Get a new Puppy
Chores
2014

UPSEARCH

```
Name     Provider                                    Root                CurrentLocation
----     --------                                    ----                ---------------
A        Microsoft.PowerShell.Core\FileSystem        A:\
Alias    Microsoft.PowerShell.Core\Alias
C        Microsoft.PowerShell.Core\FileSystem        C:\                 Users\Administrator
Cert     Microsoft.PowerShell.Security\Certificate   \
D        Microsoft.PowerShell.Core\FileSystem        D:\
Env      Microsoft.PowerShell.Core\Environment
Function Microsoft.PowerShell.Core\Function
HKCU     Microsoft.PowerShell.Core\Registry          HKEY_CURRENT_USER
HKLM     Microsoft.PowerShell.Core\Registry          HKEY_LOCAL_MACHINE
Variable Microsoft.PowerShell.Core\Variable
WSMan    Microsoft.WSMan.Management\WSMan
```

Windows Components as Drives

Get-PSDrive

Explore/Use like a FileSystem

UPSEARCH

Windows PowerShell

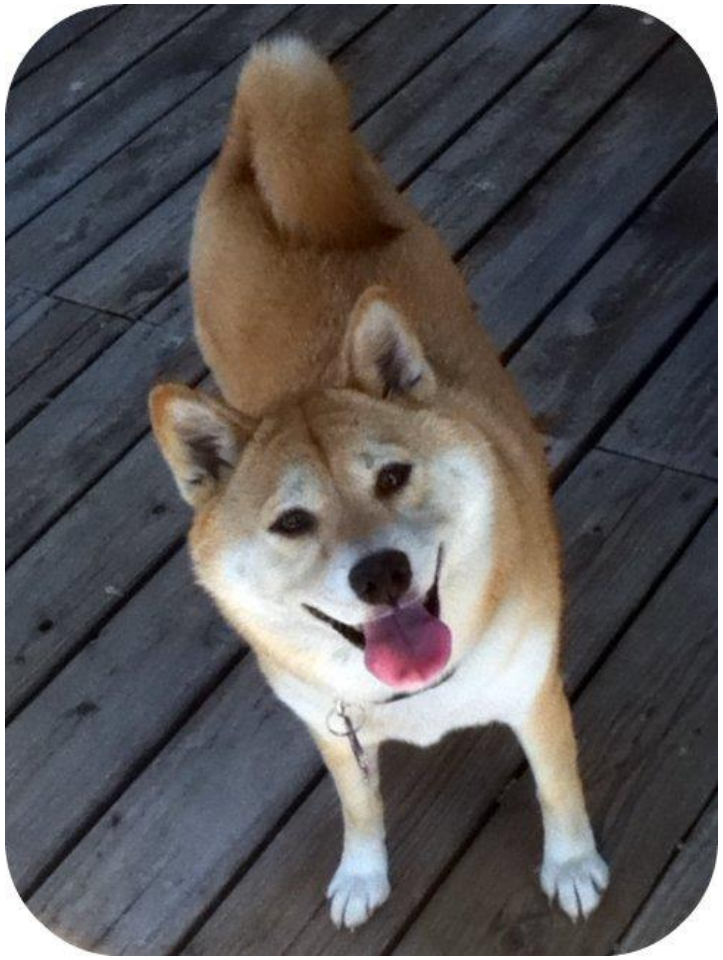
```
PS SQLSERVER:\> dir

Name             Root                      Description
----             ----                      -----------
DAC              SQLSERVER:\DAC            SQL Server Data-Tier Application
                                           Component
DataCollection   SQLSERVER:\DataCollection SQL Server Data Collection
SQLPolicy        SQLSERVER:\SQLPolicy      SQL Server Policy Management
Utility          SQLSERVER:\Utility        SQL Server Utility
SQLRegistration  SQLSERVER:\SQLRegistration SQL Server Registrations
SQL              SQLSERVER:\SQL            SQL Server Database Engine
SSIS             SQLSERVER:\SSIS           SQL Server Integration Services
XEvent           SQLSERVER:\XEvent         SQL Server Extended Events
DatabaseXEvent   SQLSERVER:\DatabaseXEvent SQL Server Extended Events
SQLAS            SQLSERVER:\SQLAS          SQL Server Analysis Services
```

DEMO!

Timeouts!

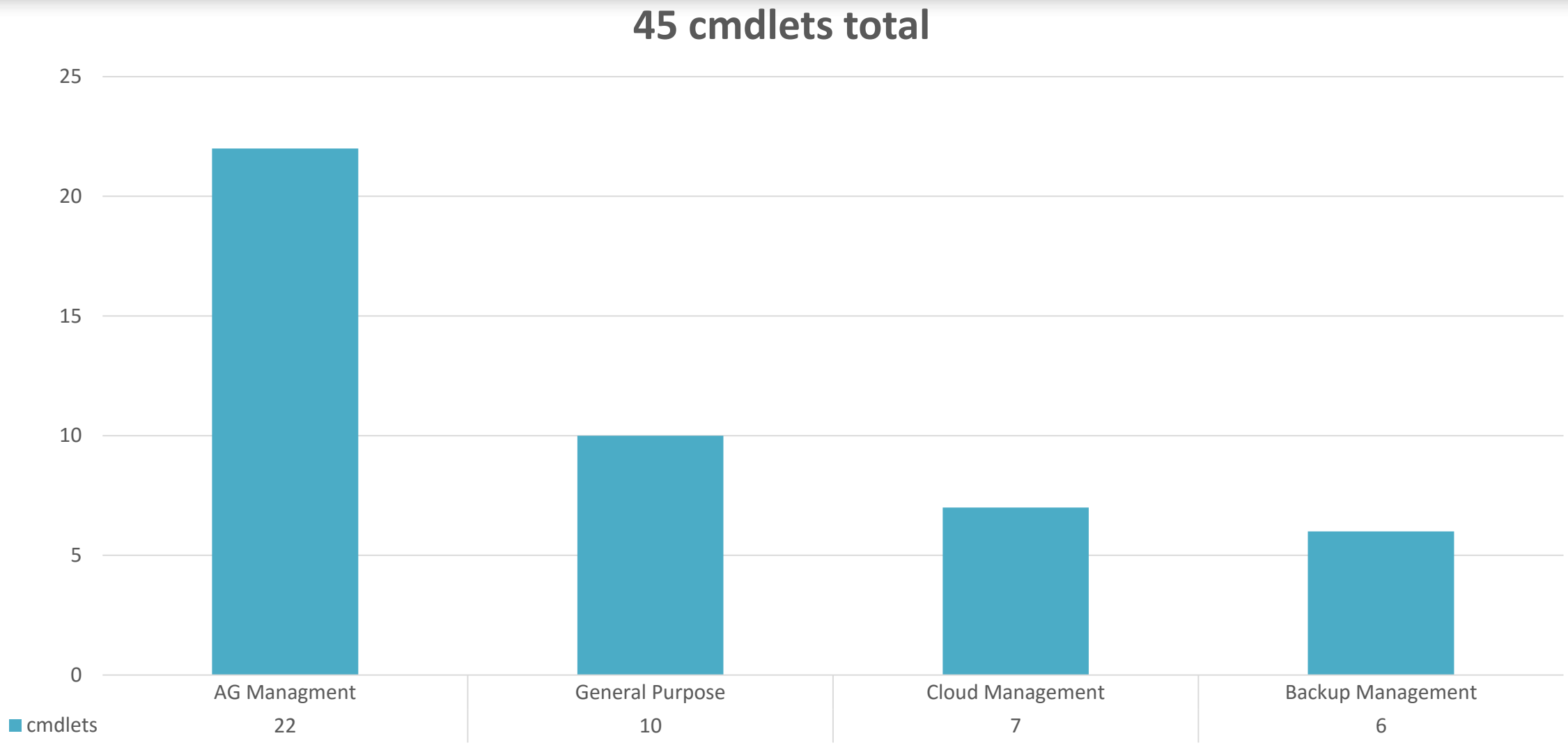
Tab completion and Intellisense

Contextual behavior



Verb-Noun

UPSEARCH


45 cmdlets total
25
20
15
10
5
0
AG Managment
General Purpose
Cloud Management
Backup Management
cmdlets
22
10
7
6

UPSEARCH

Convert-UrnToPath
Decode-SqlName
Encode-SqlName
Get-SqlCredential
Get-SqlDatabase
Invoke-PolicyEvaluation
Invoke-Sqlcmd
New-SqlCredential
Remove-SqlCredential
Set-SqlCredential

Do not match Powershell naming standards

Useful for checking Policy Based Management

“Replacement” for sqlcmd, heavily used

UPSEARCH

Backup-SqlDatabase
Restore-SqlDatabase
New-SqlBackupEncryptionOption
Get-SqlSmartAdmin
Set-SqlSmartAdmin
Test-SqlSmartAdmin

Standard
Backup and Restore
operations

SqlSmartAdmin cmdlets
for Azure Managed Backups

Add-SqlFirewallRule
Remove-SqlFirewallRule
Get-SqlInstance
Set-SqlAuthenticationMode
Set-SqlNetworkConfiguration
Start-SqlInstance
Stop-SqlInstance

These cmdlets all require the Cloud Adapter Service

Careful, this doesn't return what you think it does…

These do the same as Start-Service and Stop-Service

Add-SqlAvailabilityDatabase	Remove-SqlAvailabilityReplica
Add-SqlAvailabilityGroupListenerStaticIp	Resume-SqlAvailabilityDatabase
Disable-SqlAlwaysOn	Set-SqlAvailabilityGroup
Enable-SqlAlwaysOn	Set-SqlAvailabilityGroupListener
Join-SqlAvailabilityGroup	Set-SqlAvailabilityReplica
New-SqlAvailabilityGroup	Set-SqlHADREndpoint
New-SqlAvailabilityGroupListener	Suspend-SqlAvailabilityDatabase
New-SqlAvailabilityReplica	Switch-SqlAvailabilityGroup
New-SqlHADREndpoint	Test-SqlAvailabilityGroup
Remove-SqlAvailabilityDatabase	Test-SqlAvailabilityReplica
Remove-SqlAvailabilityGroup	Test-SqlDatabaseReplicaState

Advanced Topic

UPSEARCH

Always Encrypted
18 cmdlets

Add-SqlAzureAuthenticationContext	New-SqlColumnEncryptionKey
Add-SqlColumnEncryptionKeyValue	New-SqlColumnEncryptionKeyEncryptedValue
Complete-SqlColumnMasterKeyRotation	New-SqlColumnEncryptionSettings
Get-SqlColumnEncryptionKey	New-SqlColumnMasterKey
Get-SqlColumnMasterKey	New-SqlCspColumnMasterKeySettings
Invoke-SqlColumnMasterKeyRotation	Remove-SqlColumnEncryptionKey
New-SqlAzureKeyVaultColumnMasterKeySettings	Remove-SqlColumnEncryptionKeyValue
New-SqlCertificateStoreColumnMasterKeySettings	Remove-SqlColumnMasterKey
New-SqlCngColumnMasterKeySettings	Set-SqlColumnEncryption

SQL Agent
6 cmdlets

Get-SqlAgent
Get-SqlAgentJob
Get-SqlAgentJobHistory
Get-SqlAgentJobSchedule
Get-SqlAgentJobStep
Get-SqlAgentSchedule

SQL Error Log
2 cmdlets

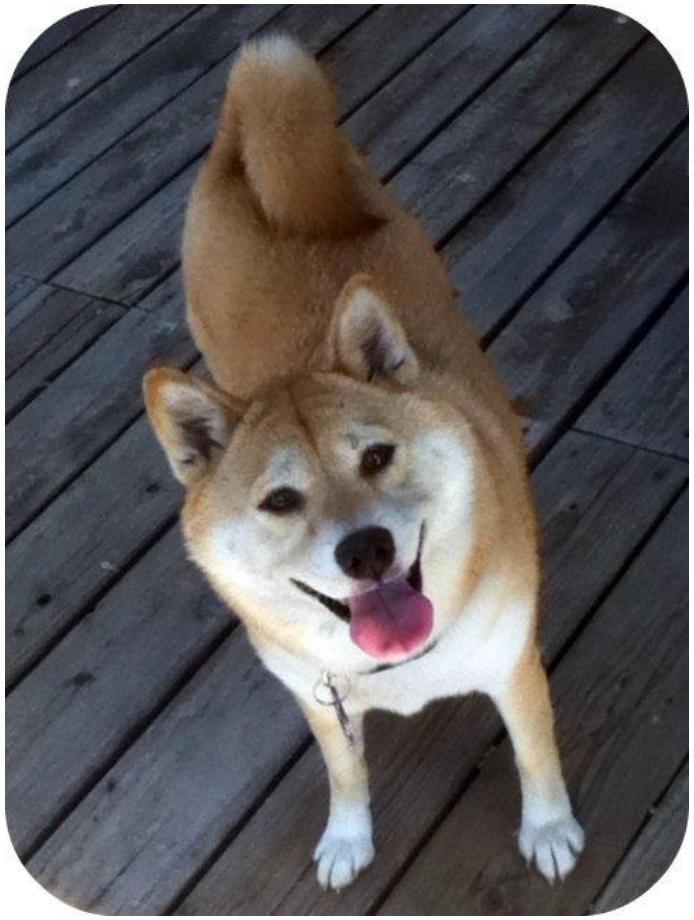
Get-SqlErrorLog
Set-SqlErrorLog

DEMO!

Poorly Documented *(Changing!)*

Unexpected behavior

Lack of support *(Changing!)*




WHAT'S
NEXT?

DEMO!

- ✓What is the SQLPS Module?
- ✓How do we use it?
- ✓The Provider, what is that all about?
- ✓What kind of cmdlets (functions) are there?
- ✓How do I use this darn thing, anyway?

UPSEARCH

mike@mikefal.net

www.mikefal.net

@Mike_Fal

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

https://github.com/MikeFal