Microsoft® Most Valuable Professional

PoShDBA: Practical PowerShell for the DBA

Microsoft® CERTIFIED Master | SQL Server® 2008

Ben Miller - @DBAduck
Email: ben@benmiller.net
http://www.powershelldba.com
http://dbaduck.com

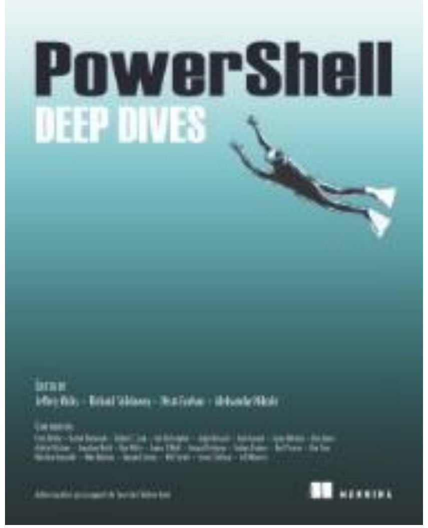
SQL intersection

DBADuck ben miller

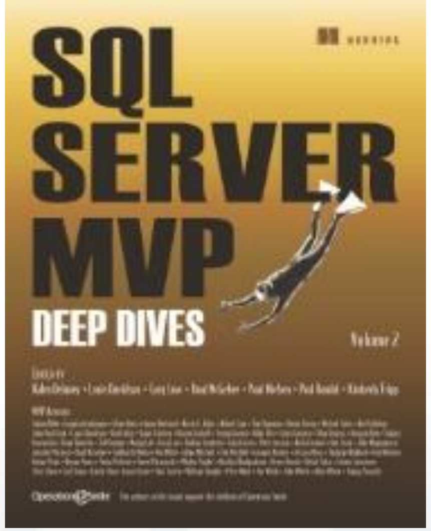
About Me!

- Data Platform MVP (7 years and running)
- Certified Master SQL Server
- Author
- SQL Server Chapter Leader in Utah County, Utah
- Independent Consultant
- PowerShell enthusiast
- @DBAduck
- SQL PowerShell Stairway on SQL Server Central
- Avid speaker at SQL Saturdays

MVP
Microsoft® Most Valuable Professional



Microsoft® CERTIFIED Master | SQL Server® 2008



SQL intersection

Reminder: Intersect with Speakers and Attendees

- **Tweet tips and tricks that you learn and follow tweets posted by your peers!**
  - Follow: #SQLintersection and/or #DEVintersection
- **Join us – Tuesday Evening – for SQLafterDark**
  - Doors open at 7:00 pm
  - Trivia game starts at 7:30 pm

    Winning team receives something fun!
  - Raffle at the end of the night

    Lots of great items to win including a seat in a SQLskills Immersion Event!
  - The first round of drinks is sponsored by SentryOne and SQLskills

Agenda

- Foundations of PowerShell
- Setting the Stage for using PowerShell and SMO together
- Loading Assemblies and using SMO
- Common Operations
- SQL Server Provider for Power in the Shell
- Where to go from here?

Foundation for PowerShell

- $pshome – c:\windows\system32\WindowsPowerShell\v1.0
- $home - $home\Documents\WindowsPowerShell
- Local $profile – In Documents folder
- Global Profile – In $pshome
- Global is first – Local Second

- X64 PSHome = c:\Windows\System32\WindowsPowerShell\v1.0
- X86 PSHome = C:\Windows\SysWOW64\WindowsPowerShell\v1.0

Foundation for PowerShell (part 2)

- **$ - preface for Variable**
- **` escape character**
- **String Substitution “My name is $name”**
  - With “Ben” in $name you get “My name is Ben”
- **Literal Strings ‘My name is $name’**
  - You get “My name is $name” as output
- **-Parameters**
- **Get-Command**
- **Get-Help**

SQL intersection

Foundation for PowerShell and the DBA

- **SQL Objects Installed**
  - SQL Tools from SQL Setup
  - Feature Pack for your platform – bingle “SQL 2016 Feature Pack”
- **PowerShell installed (v3.0 or above)**
- **PowerShell objects installed**

SQL intersection

Setting the Stage…

- SMO (Shared Management Objects)
- Management Studio uses SMO
- SMO emits TSQL
- PowerShell is a Scripting Language
- Objects live in Memory in current session
- Objects have Properties and Methods

SQL intersection

Loading Assemblies

- **Loading the SQL Provider or Module**
  - Loads almost all of the SMO assemblies
- **Loading SMO Assemblies**
  - Individually Load them

Loading SMO / Provider and Assemblies

- **Install Tools**
- **Install Feature Pack for your platform (x86 or x64)**
  - SqlClrTypes
  - Shared Management Objects
  - PowerShell tools
- **Import-Module sqlps -DisableNameChecking**
- **LoadWithPartialName**
- **LoadFrom, Load, LoadFile**
- **Add-Type**

SQL intersection

Load With PartialName

- [System.Reflection.Assembly]::LoadWithPartialName(name)
- Cannot specify Version of assembly
- Will get the Latest version of assembly on machine
- Deprecated method, don't use and convert old code

LoadFrom, LoadFile, Load

- **Very specific**
- **[system.reflection.assembly]::LoadFrom(Path)**
- **[system.reflection.assembly]::LoadFile(Path)**
- **[system.reflection.assembly]::Load(Path) ***
- **Must distribute assembly with .ps1 files**
- **Supportability issues going forward with framework versions**

- ***** ***You can use a strong name with this option as well.***

SQL intersection

Add-Type

- **Two ways to load**
- **Add-Type Weak Name**
  - Add-Type –AssemblyName “Microsoft.SqlServer.Smo”
  - Loads specific assembly defined by the PowerShell setup not you
  - For SQL v 9.0.242.0
- **Add-Type Strong Name**
  - Add-Type –AssemblyName "Microsoft.SqlServer.Smo, `
  - Version=10.0.0.0, Culture=neutral, `
  - PublicKeyToken=89845dcd8080cc91"

SQL *intersection*

Examples

```
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo")
[System.Reflection.Assembly]::Load("Microsoft.SqlServer.Smo")
[System.Reflection.Assembly]::Load("Microsoft.SqlServer.Smo,
                    Version=10.0.0.0, Culture=neutral,
                    PublicKeyToken=89845dcd8080cc91")
[System.Reflection.Assembly]:: LoadFrom("c:\Sample.Assembly.dll")
[System.Reflection.Assembly]:: LoadFile("c:\Sample.Assembly.dll")

Add-Type –AssemblyName "Microsoft.SqlServer.Smo"
Add-Type –AssemblyName "Microsoft.SqlServer.Smo,
                    Version=10.0.0.0, Culture=neutral,
                    PublicKeyToken=89845dcd8080cc91"
```

Practical Usage of PowerShell

- Profile
- Start-Transcript (not compatible with ISE before v5)
- Start-Service
- Get-Service
- Get-Command -module “Module Name”
- Get-History
- Modules, Modules, Modules

Common Operations

- Add Login
- Copy Login
- Change Configuration
- Get-* - Get Items from SQL to see how they are configured
- Add space to the database – grow file
- Get Database Stats and Files information

SQL intersection

SQL Server Provider

- **Path Like Structure**
- **SQLSERVER: drive**
- **SQLSERVER:\sql is the root of the Engine**
- **SQLSERVER:\sql\MACHINE\instancename (or default)**

PowerShell for the DBA

Demo

SQL intersection

What Now?

- Iteration with ForEach
- GetDefaultInitFields
- SetDefaultInitFields
- Use Profiler to continue to fish
- Be specific if performance is key
- Be aware of the Pipeline

Questions?

DBADuck
ben miller

Don’t forget to complete an online evaluation!

PoShDBA: Practical PowerShell for the DBA

Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

SQL intersection

Thank you!

Save the Date

Oct 30-Nov 2, 2017

We’re back in Vegas baby!

www.SQLintersection.com

SQL
intersection