



Learning PowerShell as a DBA

Ben Miller – ben@benmiller.net

He/Him

SQL Server Architect

Ben Miller & Associates, Inc.

Ben Miller

He/Him



SQL Server Architect DBADuck Consulting

@DBADuck – Twitter, Facebook, LinkedIn

www.dbaduck.com | github.com/dbaduck

ben@benmiller.net

Ben has been a member of the SQL Server Community since 1999. He also consults on SQL Server problems. He is a SQL Server MVP a SQL Server MCM, and has spent time in the field using SQL Server since 1997. He has worked at various companies throughout the US, as well as at Microsoft for 7 years. He is passionate about SQL Server and automation and integration and uses SMO and PowerShell regularly. He teaches DBAs how to use PowerShell as well. He is @DBADuck all around the web (Twitter, LinkedIn, Facebook)



What are we covering today?

- Environment
- PowerShell Syntax and Context for DBAs
- Objects for the DBA
- Practical Use for the DBA

Environment

- Windows PowerShell 5.1 (Native on 2016+)
- Modules (sqlserver, dbatools)
- VSCode or Azure Data Studio as a tool

PowerShell Syntax for DBAs

| DBA – SQL Server | PowerShell |
|---|---|
| Variables – DECLARE @variable int | \$variable |
| Strings – ‘string’ or N’string’ | ‘literal string’ or “replaceable string” |
| Parameters - @param = 3, @param2 = 4 | -Param 3 -Param2 4 |
| Escape Character - %%string% | ` (back tick) to escape characters |
| Arrays in TSQL – None | @() or @(1,2,3) or 1,2,3 |
| Hash Tables in TSQL – None | @{ } or @{name=“value”; name2=“value2” } |
| Comparison Operators - <>, <, >, =, ! (NOT) | -ne, -lt, -gt, -eq, -contains, -notcontains, -not |
| Pipelines in TSQL – None | (Pipe symbol) |
| SELECT in TSQL – SELECT column, column2 | \$obj Select-Object column1, column2 |
| WHERE in TSQL – SELECT col1 FROM tbl WHERE | \$obj Where-Object { \$_.Name -eq “DEMO” } |

Context for DBAs

- Data (tables, rows, columns)
- Objects (Database, Table, Index)
 - New-Object
 - Properties \$var.Name
 - Methods \$var.Alter()

What is next for a DBA?

- Get-Help
- Get-Command
- Get-Member
- Modules (dbatools, sqlserver)
- Prebuilt commands for managing SQL Server
 - Import-Module dbatools
 - Get-DbLogin

Cheat Sheet for Modules

- Verb-Noun (Get, Set, Backup) – (Login, Database)
- Dbatools module has Db* and Db*Db prefixes
 - Get-Db*Login
 - Get-Db*DbFile
- SqlServer module has Sql prefix
 - Get-SqlLogin
 - Get-SqlInstance

Practical Use for a DBA

- Transcripts – Capture all statements and output from statements executed in Session
- Profile – Preconfiguration of your session to allow you to do some pre-setup
- Objects that directly affect SQL objects (SMO)

Practical Use for a DBA

- Grow a log file
- Backup a Database
- Copy logins from server to server
- Create a new login
- Get logins on a server with a filter (find a login)
- ETC.....

Cheat Sheet for SMO

- Almost all objects in SMO have `Alter()`, `Drop()` methods to allow you to change settings and affect the object in SQL Server
- Collections – Just like SSMS there are collections in SMO. Databases, Tables, Columns, Indexes
- Learning to navigate SMO is much like SSMS as it uses SMO for its actions
- Collection Syntax involves `collection["name"]` or `[1]`

PowerShell Demo for DBAs

Review

- Environment
- PowerShell Syntax and Context for DBAs
- Objects for the DBA
- Practical Use for the DBA

Learning PowerShell as a DBA

Ben Miller – ben@benmiller.net

@DBAduck everywhere

Twitter, LinkedIn, Facebook, etc.

www.dbaduck.com | github.com/dbaduck