



# SQLintersection

Session: Sunday, June 8, 2021 9:00 AM

## Tips and Tricks for the PowerShell DBA (1)

Ben J Miller

[ben@benmiller.net](mailto:ben@benmiller.net)

SQL Server & Azure SQL  
Conference

POWERED BY  
Microsoft & Next Gen



# Get *Whova* for all 4 conferences

## Official Event App

- **Access links** to join all sessions and keynotes virtually
- Explore the **professional profiles** of event speakers and attendees
- Send **in-app messages** and **exchange contact info**
- **Network and find attendees** with common affiliations, educations, shared networks, and social profiles
- Receive **update notifications** from organizers
- Access the **event agenda**, GPS guidance, maps, and parking directions at your fingertips

**Download Whova and take your event mobile.**

Get Whova from the App Store or Google Play.



Please sign up for the app with your social media account or email.

The event invitation code is:  
**damah**

You will be asked for an event invitation code after installing Whova.



# Sponsors & Exhibitors

## Platinum Sponsors



Booth 7



ORANGE BEES

Booth 6

Quest®

## Gold Sponsors



## Silver Sponsors



Booth 5

## Exhibitors



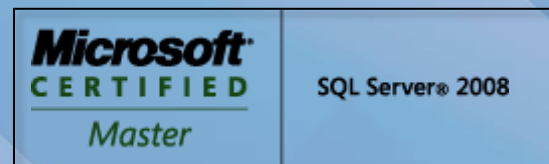
Booth 2



Booth 1

# Speaker: Ben Miller - DBADuck

- Data Platform MVP since 2009
- Microsoft Certified Master in SQL Server
- Author – <https://dbaduck.com>
  - MVP Deep Dives #2
  - PowerShell MVP Deep Dives
  - PowerShell Stairway on SQLServerCentral.com
- Email : [ben@benmiller.net](mailto:ben@benmiller.net)
- Twitter: @DBADuck
- GitHub: [github.com/dbaduck](https://github.com/dbaduck)



# Introduction

- **Transcripting**
- **Array vs. ArrayList**
- **Strings vs. StringBuilder**
- **1..X iteration**
- **.ForEach() usage**
- **Splatting (what is this?)**
- **-eq vs. -ceq**
- **Pipelining**
- **Object Creation tips**

# Transcripting

- Think of it as a log that is automatic. What IT sees, it records.
- Just a file
- Start-Transcript
- Path gets fully created even if it does not exist
- ***Make sure*** that you do Stop-Transcript before you exit
  - Content is buffered and may just not record if you X out of your window

# Array vs. ArrayList

- **[System.Collections.ArrayList]**
- **.NET type**
- **.Add(object) to add to the array list**
- **Memory Buffer**
  
- **Array (native to PowerShell)**
  - **@( )** is a blank Array, **@(1,2,3)** is Predefined, **1,2,3** is dynamic
- **\$ary += \$obj to add to the array**
- **Immutable – cannot be changed**



# String vs. StringBuilder

```
$obj = "String"
```

```
$obj = "String" + " " + "Builder"
```

```
$obj = "$obj Builder"
```

Strings are Immutable – cannot be changed

```
[System.Text.StringBuilder]
```

```
$obj = new-object -typename System.Text.StringBuilder -Args 4096
```

```
$obj.Append(" ") or $obj.AppendLine(" ")
```

```
$obj.AppendFormat("{0} {1}", "one", "two")
```



# 1..X Syntax For Iteration

- **1..5 produces an array of 1 – 5 one at a time**
- **Can be used with Foreach()**
- **Increments by 1 so you can start at any number**
- **Can be done in reverse order**
- **5..1 and it will produce 5,4,3,2,1**
- **Useful and kind of like GO 5 except that it does not always have to go in reverse, and you can use the number in PowerShell**
- **1..5 | Foreach { \$\_ }**

# ForEach() on an Object

- **Dynamic Properties and Methods**
- **`$obj.Foreach( { code block; } )`**
- **Iteration for each item in the collection in `$obj`**
- **\* Bonus: You can add your own as well.**

# Splatting in PowerShell

- Splatting is all about a Variable satisfying Parameters
- Variables look like this \$obj
- Splatting uses @obj
- Hash Table with multiple Keys/Values passed into a function/cmdlet
- *Rules: You can splat with a variable that has as many or fewer parameter satisfying keys/values but NOT more.*

# Case Sensitive Comparisons

- **PowerShell is Case Insensitive**
- **Comparing values (mostly Strings) is the same**
  - Ben = ben
- **Comparison Operators**
  - -eq -lt -gt etc.
- **Case Sensitive Operators**
  - -ceq -clt -cgt etc.
- **Great for password comparison and others that require Case Sensitivity**

# Using the Pipeline

- Using the Pipeline character | (pipe character)
- Objects are passed **By Value** not By Reference on the pipeline which means that the object passed is copied so now there are 2 of them in the memory space
- Keep the objects getting smaller on the left and getting smaller going to the right
- Powerful Tip to pass a set of objects over a pipe to a Cmdlet that handles a set of objects

# Object Creation Tips

- **Objects are used to store data for use in your automation**
- **1. `$obj = New-Object -TypeName PSObject (not my favorite)`**
  - `Add-Member`
- **2. `$obj = "" | Select Name, ID, Description`**
- **3. `$obj = @{ Name = Value; ID=6; Description = "Desc" }`**
- **The only one I would stay away from on custom objects is #1**

# Demo

## Demo Title

Subtitle

SQL Server & Azure SQL  
Conference

POWERED BY  
Microsoft & Next Gen

© 2021. All rights reserved.  
<http://www.MSSQLConf.com>



# Review

- **Array vs. ArrayList**
- **Get Assemblies in your session**
- **Strings vs. StringBuilder**
- **1..X iteration**
- **.ForEach() usage**
- **Splatting (what is this?)**
- **-eq vs. -ceq**
- **Pipelining**
- **Object Creation tips**

# Questions?

Don't forget to complete an online evaluation!

## Tips and Tricks for the PowerShell DBA (1)

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

SQL Server & Azure SQL  
Conference

POWERED BY  
Microsoft & Next Gen

Thank you!

IN-PERSON EVENT

DECEMBER 7–9  
2021

SQL Server & Azure SQL  
Conference

POWERED BY  
Microsoft & Next Gen

LAS VEGAS, NV  
MGM GRAND

