

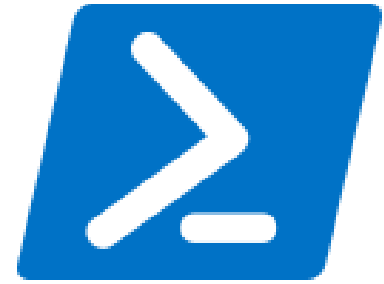
VALIDATE YOUR POWERSHELL PARAMETERS WITH ENUMS

Paul Broadwith

 [@pauby](https://twitter.com/pauby)

 pauby.com

 github.com/pauby



ABOUT ME

- Paul Broadwith
 - Freelancer since 2001
 - 25 years in IT in the defence, government, financial services and nuclear industry sectors;

• Contact Me

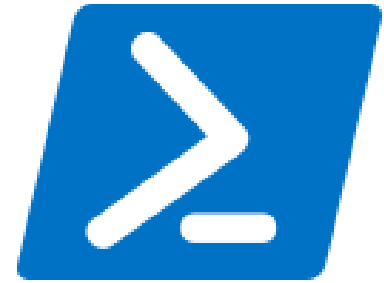
 <https://pauby.com>

 @pauby

 <https://github.com/pauby>

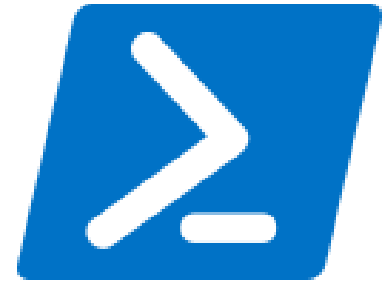
 <https://www.linkedin.com/in/paulbroadwith/>





WHAT IS AN ENUM

- Short for enumerated (data) type;
- Defines a set of members that a data type can have and restrict its value to those members;
- Rules:
 - An enum can only hold one member;
 - The value of an enum is restricted to the defined set of members;
 - Each member of the defined set has an associated integer value starting from zero for the first member;



COMMON POWERSHELL ENUMS

- **Get-ChildItem**
 - Enum: <Attributes>
- **Set-ExecutionPolicy:**
 - Enum: <ExecutionPolicy>
 - Enum: <ExecutionPolicyScope>
- **Set-Service**
 - Enum: <ServiceStartMode>
- **Write-Host**
 - Enum: <ConsoleColor>



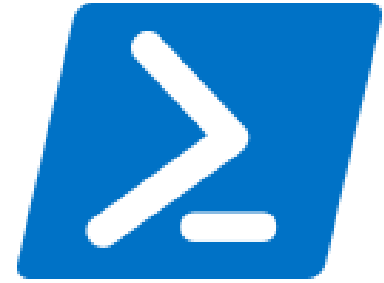
COMMON POWERSHELL ENUMS

CODE



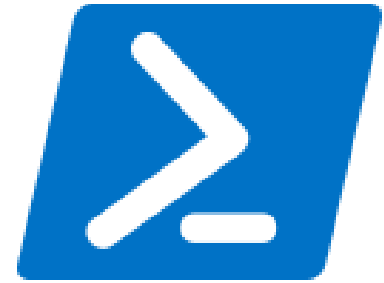
ENUMS HAVE INTEGER VALUES

- Each enum member has an integer value;
- The integer value is determined automatically or manually:
 - Automatically:
 - Each enum member has an integer value starting at zero for the first in the set, 1 for the second and so on;
 - Manually:
 - This is set when creating the enum members set;
 - When a member in the set is given an integer value, numbering starts at that and continues until the end of the member list or another integer is provided;



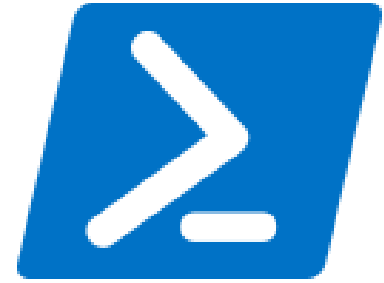
ENUMS HAVE INTEGER VALUES

CODE



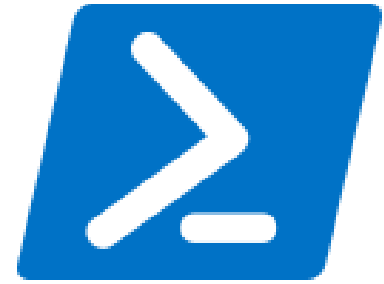
CREATING YOUR OWN ENUMS

- There are two methods to create an enum depending on which PowerShell version you are using:
 - Pre-5
 - Use Add-Type
 - 5 onwards
 - Use 'enum' keyword
 - Rules for creating enums:
 - Semicolons are only required when creating an enum on one lines but it's good practice to always use them;
 - Member names cannot include spaces or special characters (except an underscore)
 - don't be smart and put it in quotes to get around this as it won't work;
 - The enum members cannot be numbers;



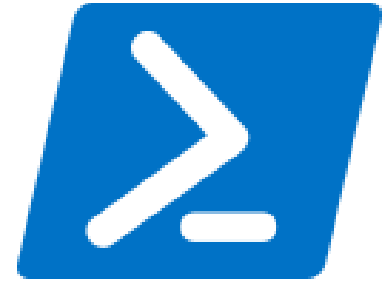
CREATING YOUR OWN ENUMS

CODE



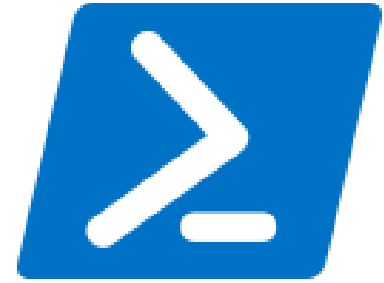
ASSIGNING INTEGER VALUES TO MEMBERS

- Members are automatically assigned an integer value starting from zero for the first member;
- You can assign your own integer values;
- When you assign an integer value to a member it restarts the counter to that value and each member thereafter will get a value incremented by one;



ASSIGNING INTEGER VALUES TO MEMBERS

CODE



USING ENUMS IN SWITCH STATEMENTS

- The enum member name is converted to a string so can be used as a match in your switch statement;



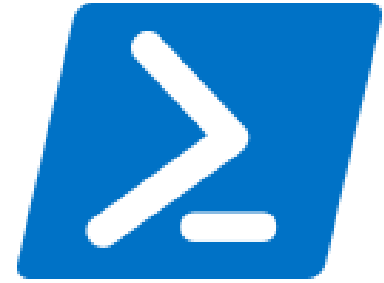
USING ENUMS IN SWITCH STATEMENTS

CODE



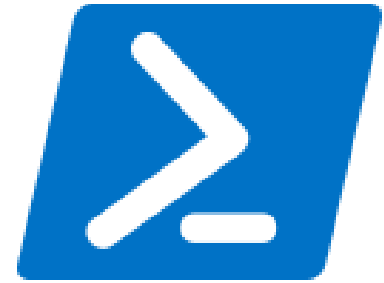
TRADITIONAL PARAMETER VALIDATION

- To perform parameter validation we traditionally use `[ValidateSet()]`
- If multiple functions will use the same set of values for validation then we must define it each time;
- If we update the list we need to update it in several places potentially across several scripts within a module;



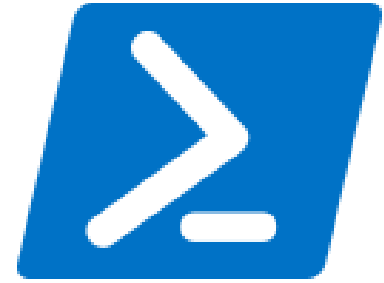
TRADITIONAL PARAMETER VALIDATION

CODE



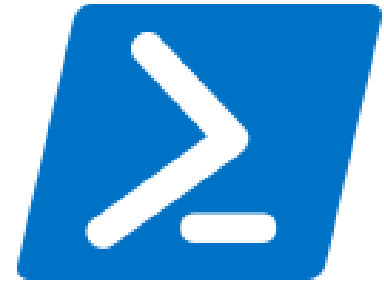
PARAMETER VALIDATION WITH ENUMS

- Create an enum with a member list of values;
- Use the enum as the data type for your function parameter:
 - No need for `[ValidateSet()]`
 - No need for `[ValidateNotNullOrEmpty()]`
 - The only values accepted are the enum members;



PARAMETER VALIDATION WITH ENUMS

CODE

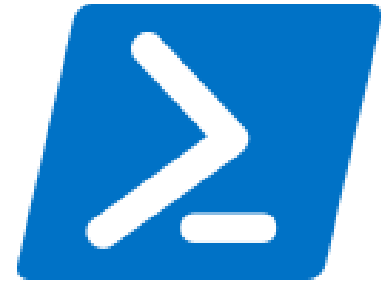


SUMMARY

- Enums are simple to create;
- Help prevent errors by only accepting a predefined set of values;
- Easily replace `[ValidateSet()]` and `[ValidateNotNullOrEmpty()]` when working with more than one function using the same set of parameter validation values;

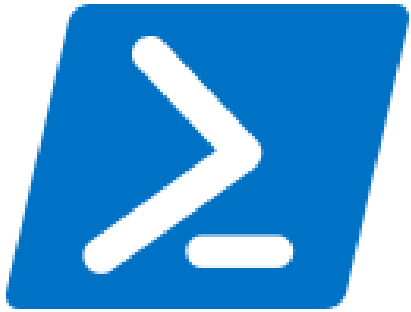


QUESTIONS?



RESOURCES

- Working With Enums In PowerShell
 - <https://blog.pauby.com/post/working-with-enums-in-powershell/>
- Creating Enums In PowerShell
 - <https://blog.pauby.com/post/creating-enums-in-powershell/>



THANK YOU!

Paul Broadwith

 [@pauby](https://twitter.com/pauby)

 pauby.com

 github.com/pauby