

Reach the next level with PowerShell

Jaap Brasser
@jaap_brasser



jaapbrasser.com/about

- PowerShell Conf EU/Asia
- Dutch PowerShell User Group
- Blogging
- GitHub
- PowerShell Gallery
- TechNet Forums/Gallery

 @Jaap_Brasser

Agenda

P O W E R S H E L L



 **interface**
WWW.INTERFACETT.COM

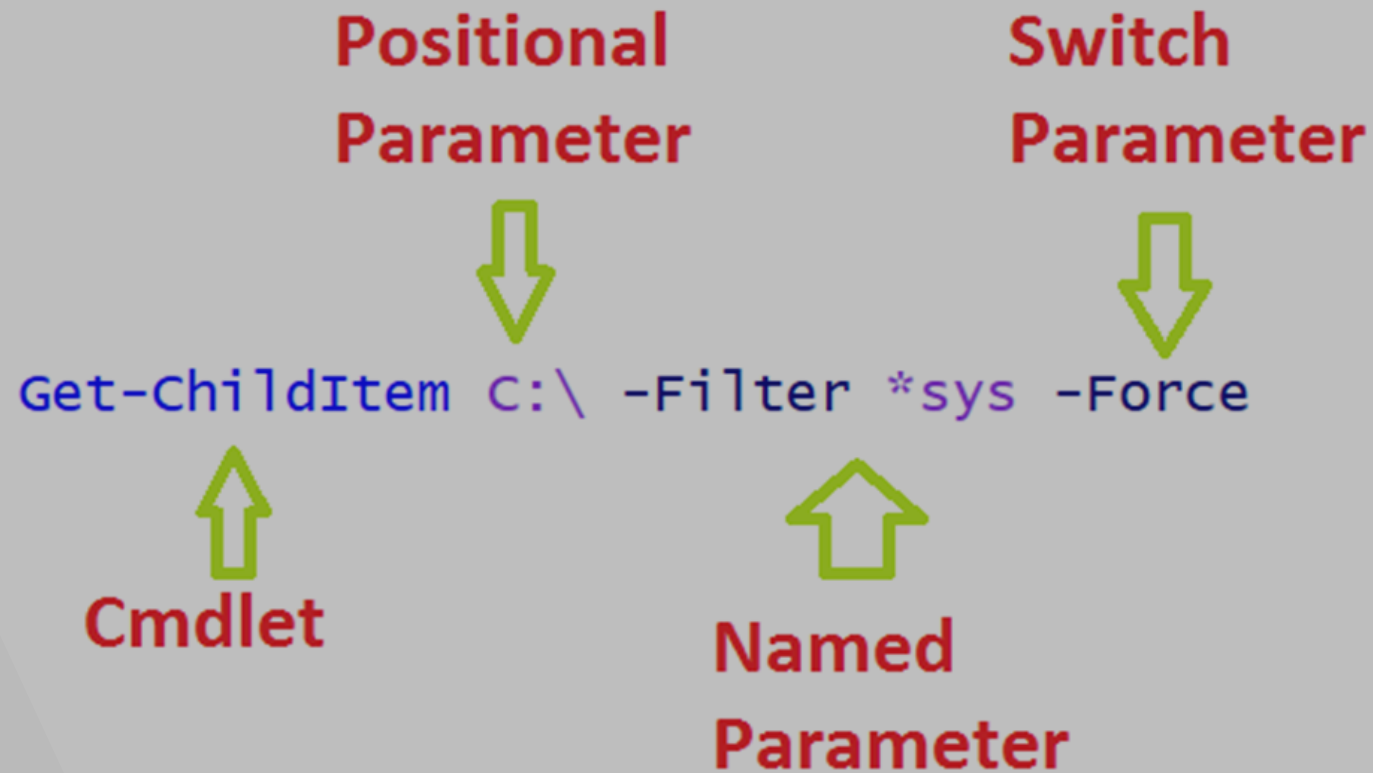
PowerShell versions

OS	PowerShell version
Windows 7, Server2008R2	2.0
Windows 8, Server2012	3.0
Windows 8.1, Server 2012R2	4.0
Windows 10, Server 2016	5.0
Windows 10 Creators Update	5.1

- How to check your PowerShell version:

```
$PSVersionTable
```

PowerShell language





What are we building?

Requirements

1. Structured output – Objects!
2. Hostname of the system
3. Domain name of system
4. OS info
5. Memory info
6. System disk
7. Pagefile
8. IP info
9. Create a function

Demo 1

Retrieve hostname



Demo Summary

- Found the host name of the system
- Looked at PowerShell drives
- Explored the Env: drive
- Found the hostname of the system
- Created a PowerShell object
- Exported the object to csv
- Opened csv in Excel

Demo 2

Get System Domain



Demo Summary

- Found the user domain
- Found the system domain
- Merged output with previous output

Demo 3

OS Information



Demo Summary

- Used the Win32_OperatingSystem class
- Retrieved the information
- Established Get-WmiObject is obsolete
- Used Get-CimInstance
- Help examples with array indexing
- Used WQL to speed up query
- Get-CimInstance hiding output

Demo 4

Memory information



Demo Summary

- Retrieved information from ComputerSystem
- Divided by GB
- Strong typed as integer to remove rounding
- Used format operator to create string

Demo 5

Timezones



Demo Summary

- Used the Get-TimeZone cmdlet
- Found out that the timezone description does not take DST into account
- Used [datetime] type accelerator
 - UtcNow
 - Now
- Calculated actual time difference
- Combined output into a single string

Demo 6

System disk size



Demo Summary

- Find out which disk is the system disk
- Find out how big the disk is
- Create the correct WQL query
- Format output into a string
- Set number of decimals for size in GB with `[math]::round`

Demo 7

Pagefile from registry



Demo Summary

- Found the registry provider PSDrive
- Looked at the 'Memory Management' key
- Retrieved the information using Get-ItemProperty
- Used the -join operator to collapse the array

Demo 8

Create the function



Demo Summary

- Create the Get-ComputerInformation function
- Create a hashtable
- Fill the values in the hashtable using output from the commands created
- Convert the hashtable to a PowerShell custom object
- Ran the function



Questions ?

[Github.com/jaapbrasser/Events/](https://github.com/jaapbrasser/Events/)
[Jaapbrasser.com](https://jaapbrasser.com)