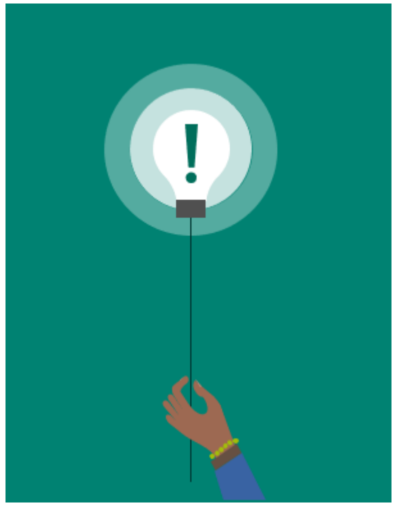
Microsoft

PsCustomObject

Objectives

After completing this learning, you will be able to:

- Understand what is PSCustomObject
- When to work with PSCustomObject



About PsCustomObject (By Don Jones)

[PSCustomObject] is a type accelerator.
It constructs a PSObject, bot does so in a way that results in hash table keys becoming properties.

PsCustomObject isn’t an object type per se – it’s a process shortcut.

PsCustomObject is a placeholder that’s used when PSObject is called with no constructor parameters.

PSCustomObject

```
$person = [pscustomobject]@{
    Name      = “Gadi"
    age       = 48
    Office    = "Home"
    Title     = “CE Platform Active Directory"
    Planet    = "Earth"
    birthdate = Get-Date -Date "1974-11-23"
}
```

Simplification and Standardization

Using a PScustomObject

```
$person = [pscustomobject]@{
    Name        = “Gadi"
    age         = 48
    Office      = "Home"
    Title       = “CE Platform Active Dir
    Planet      = "Earth"
    birthdate = Get-Date -Date "1974-11
}

$person.GetType().fullname

System.Management.Automation.PSCustomObject
```

- Pscustomobject returned
- Pscustomobject full name

Listing Properties

We can list all of the properties by using these 2 methods

```
# Listing all properties
$person.psobject.Properties.Name

$person | get-member -MemberType
NoteProperty | Select-Object
-ExpandProperty name

Get-Member -InputObject $person
-MemberType NoteProperty | Select-Object
-ExpandProperty name
```

Add Change and Remove properties dynamically

- We can change properties to a PsCustomObject dynamically

- We can add new properties in runtime to an existing PsCustomObject

- We can remove property

- If we Add/Remove properties, we need to check if they exist first !

```
$person.Planet = ‘Mars’


$person | Add-Member -MemberType
NoteProperty -Name 'Favourite_Food’
-Value 'Steak’


$person.PSObject.Properties.Remove('favo
urite_Drink’)

Get-Member -InputObject $person
-MemberType Properties | Select-Object
-ExpandProperty Name
```

Adding Methods(!!!) to our PsCustomObject

We can add functionality to our PsCustomObject.

```powershell
$method = {
    "Hi, my name is $($this.name) and I like to write PowerShell Code!"
}

$params = @{
    MemberType = 'ScriptMethod'
    Name       = 'SayHi'
    Value      = $method
}

$person | Add-Member @params
$person.SayHi()
```

Converting PsCustomObject to HashTable

Example for a useful method, exporting our PsCustomObject to hashtable

```
$params = @{
    MemberType = 'ScriptMethod'
    Name       = 'OutHashTable'
    Value      = {
        $hash = @{}
        $this.psobject.properties.name.f
oreach({
            $hash[$_] = $this.$_
        })
        return $hash
    }
}
$person | Add-Member @params
$person.OutHashTable()
```

Changing out type

What are types?

Why should I care about the type of my PsCustomObject

How can I change it

```
$person.GetType()
IsPublic IsSerial Name                BaseType
True     True     PSCustomObject      System.Object

$person.psobject.TypeNames
System.Management.Automation.PSCustomObject
System.Object

$person.psobject.TypeNames.insert(0, "Gadla.Person")
$person | Get-Member
TypeName: Gadla.Person

OR

$person.psobject.typenames
Gadla.Person
System.Management.Automation.PSCustomObject
System.Object
```

Exporting and Importing PsCustomObject

- We can export our PsCustomObject to files
- Consider what is your use of the output file (csv for working with excel)
- Consider exporting to a JSON file format when export needs to be read by non-PowerShell language or needs to go into source control
- Consider exporting to XML file When transporting data between two PowerShell processes

Export and Import using Export-CSV

```
$person = [pscustomobject]@{
    Name      = 'Gadi'
    age       = 48
    Title     = 'CE Platform Active Directory'
    Planet    = 'Earth'
    birthdate = Get-Date -Date "1974-11-23"
}

$person | Export-Csv -Path 'C:\TEMP\psobject_to_csv.csv' -NoTypeInformation
$importedPerson = Import-Csv -Path 'C:\TEMP\psobject_to_csv.csv'
$importedPerson
```

Export and Import using JSON

Using a PScustomObject

```
$person = [pscustomobject]@{
    Name      = 'Gadi'
    age       = 48
    Title     = 'CE Platform Active Directory'
    Planet    = 'Earth'
    birthdate = Get-Date -Date "1974-11-23"
}

$Path = 'C:\TEMP\psobject_to_json.json'
$person | ConvertTo-Json -Depth 99 | Set-Content -Path $Path
$importedPerson = Get-Content -Path $Path
$importedPerson
$ImportedPersonJson = $importedPerson | ConvertFrom-Json
$ImportedPersonJson
```

Export and Import using XML

Using a PScustomObject

```
$person = [pscustomobject]@{
    Name      = 'Gadi'
    age       = 48
    Title     = 'CE Platform Active Directory'
    Planet    = 'Earth'
    birthdate = Get-Date -Date "1974-11-23"
}

# Saving to a file (XML)
$xmlpath = 'C:\temp\pso.xml'
$person | Export-Clixml -Path $xmlpath
$xmlPerson = Import-Clixml -Path $xmlpath
$xmlPerson
$xmlPerson.birthdate.GetType()
```

Demonstration

PsCustomObject Introduction

Questions?