Objects and the Pipeline

Jeff Hicks

AUTHOR ~ TEACHER ~ SENSEI

@jeffhicks   http://blog.jdhitsolutions.com

```
Get-Process |
Measure WS –sum |
Select Count,Sum
```

Pipeline Principles

Cmdlets write objects to the pipeline

Cmdlets can be designed to take pipelined input

Repeat as needed

Pipeline Exceptions

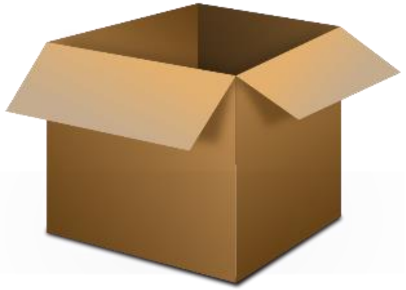
Normally, we let cmdlets do their thing

Pipelines are typically better for managing at scale

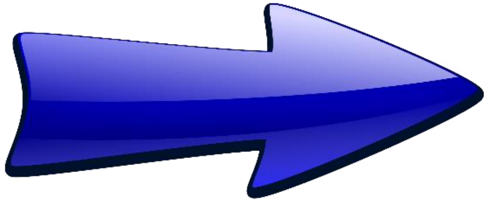
But sometimes you need to process objects individually

Foreach-Object

Cmdlet emits a bunch of objects



For each object do something



Results go down the pipeline

Out into PowerShell

Process Objects One at a Time

```
dir -file | where {$_.length -ge 1mb -AND $_.extension
-notmatch ".zip|.exe"} |
```

Process Objects One at a Time

```
dir -file | where {$_.length -ge 1mb -AND $_.extension
-notmatch ".zip|.exe"} |
Foreach-object {
 Compress-Archive –path $_.fullname
-DestinationPath "$($_.directory)\$($_.basename).zip"
-CompressionLevel Optimal
}
```

Process Objects One at a Time

Do this command for each file

```
dir -file | where {$_.length                       n
-notmatch ".zip|.exe"} |
Foreach-object {
 Compress-Archive -path $_.fullname
-DestinationPath "$($_.directory)\$($_.basename).zip"
-CompressionLevel Optimal
}
```

Process Objects One at a Time

```
dir -file | where {$_.length -ge 1mb -AND $_.extension
-notmatch ".zip|.exe"} |
Foreach-object {
 Compress-Archive –path $_.fullname
-DestinationPath "$($_.directory)\$($_.basename).zip"
-CompressionLevel Optimal
}
```

The current object

This is a one line PowerShell expression

ForEach-Object

Don’t confuse the ForEach alias with the ForEach enumerator

$_ represents current object in the pipeline

Use a semi-colon (;) to separate commands on the same line

Advanced Stuff

-Begin {}
commands that run once *before* anything is processed

-Process {}
commands that run *once* for each incoming object

Default

-End {}
commands that run once *after* everything is processed

Advanced ForEach-Object

```
dir c:\work -Directory |
foreach -begin { $h=@{} ; $results=@() }
-process {
$stat = dir $_.FullName -file -Recurse |
Measure-Object -Property length –sum ;
$h.Path = $_.fullname ; $h.Files = $stat.count ;
$h.TotalSize =$stat.sum ;
$results+=[pscustomobject]$h
}
-end {
$results
}
```

Initialize an empty hashtable

Initialize an empty array

Write the results to the pipeline

```
$numbers = 1..10
foreach ($n in $numbers) {
 $max = Get-Random –minimum 50 –maximum 500
 Set-Content –value (1..$max) –path "TestFile-$n.txt"
}
```

ForEach Enumerator

“For each thing in a collection of things, run these commands”

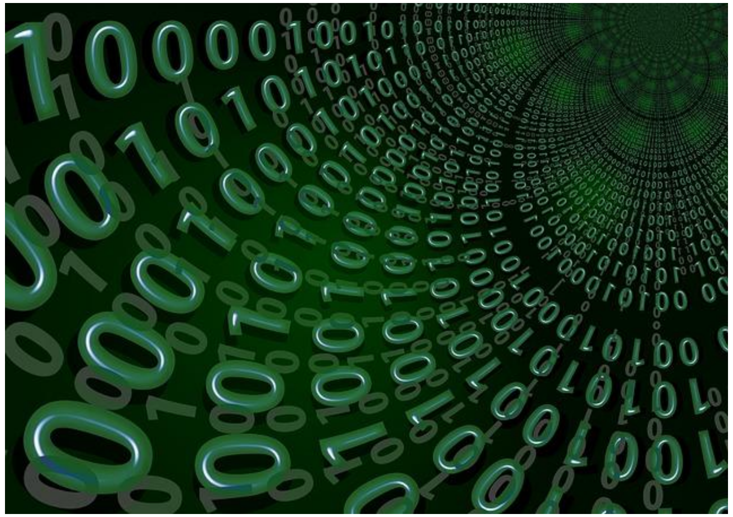
You pick the variable name

Does not write to the pipeline after the scriptblock

Demo

“I like to move it, move it!”

At the end…



PowerShell automatically handles formatting and output

We'll cover this later

But…you may want to send output beyond PowerShell

Out Options

Out-File

Out-Gridview

Out-Printer

Demo

“Out With It!”

Summary

Read full cmdlet help on ForEach-Object

Help about_foreach

The Out-* cmdlets are handy viewing alternatives