

The Pipeline in Depth



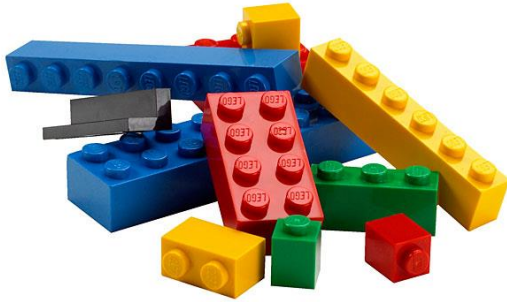
Jeff Hicks

AUTHOR ~ TEACHER ~ SENSEI

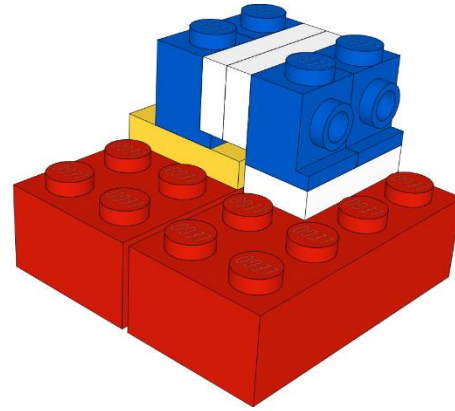
@jeffhicks <http://blog.jdhitsolutions.com>



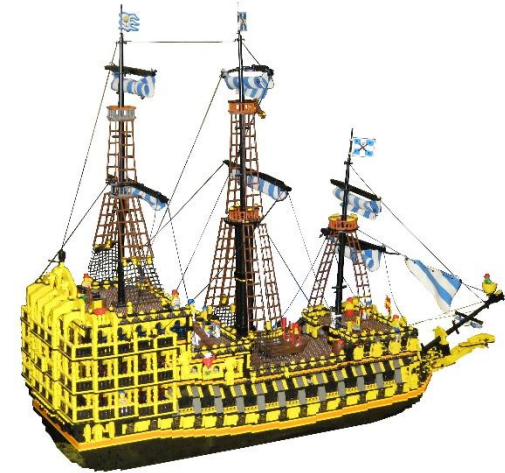
Pipeline Building



Cmdlets in the pipeline are like building blocks



Cmdlets are designed to snap objects together



You can create elaborate structures



Parameter Binding

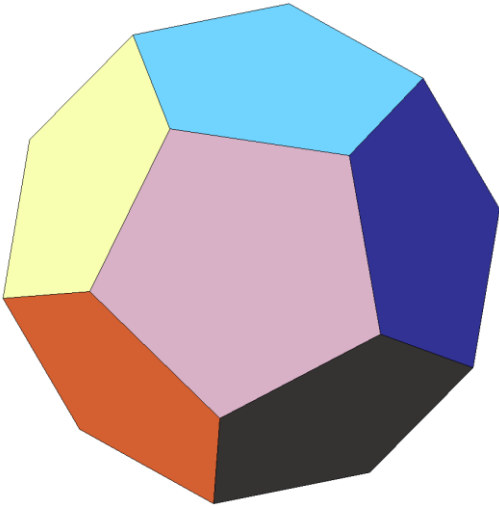
**Objects passed
down the pipeline**

**Next cmdlet can
consume the
incoming objects**

**PowerShell makes
connections**



Parameter Binding



By value



By property name

Get-Service winrm | Restart-Service

By Value

```
PS C:\> help restart-service -Parameter inputobject  
  
-Inputobject <ServiceController[]>  
    Specifies ServiceController objects that represent the services to restart. Enter a variable that contains the  
    objects, or type a command or expression that gets the objects.  
  
    Required?                true  
    Position?                0  
    Default value            None  
    Accept pipeline input?    True (ByValue)  
    Accept wildcard characters? false
```



Any object

```
[pscustomobject]@{Name='winrm';Computername='srv1'} |  
get-service
```

By Property Name

```
PS C:\> help get-service -parameter computername  
-ComputerName <String[]>  
  Gets the services running on the specified computers. The default is the local computer.  
  
  Type the NetBIOS name, an IP address, or a fully qualified domain name (FQDN) of a remote computer. To specify the  
  local computer, type the computer name, a dot (.), or localhost.  
  
  This parameter does not rely on Windows PowerShell remoting. You can use the ComputerName parameter of Get-Service  
  even if your computer is not configured to run remote commands.  
  
  Required?                false  
  Position?                named  
  Default value            None  
  Accept pipeline input?   True (ByPropertyName)  
  Accept wildcard characters? false
```



Demo



Dive! Dive! Dive!



Property Details

**Know your object
in the pipeline**

**Create custom
properties**

Expand properties




```
@{Name = 'something' ; Expression = { <your code>  }}
```

Custom properties

Create a special hashtable

Name and Expression are required keys

Use `$_` to reference current object in the pipeline



```
Get-Process | Select ID, Name, WS, StartTime,  
@{Name='Runtime';Expression={(Get-Date) - $_.starttime}}
```

Custom properties

Rename an existing property

Define something completely new



Id : 4200
Name : ZeroConfigService
WS : 19357696
StartTime : 5/31/2017 3:10:16 PM
Runtime : 00:29:29.7926163

Custom properties

Rename an existing property

Define something completely new



Property Expanding

**Objects come down the
pipeline**

**Sometimes it is a nested
object**

...or an array

**...or you may just want the
value**



```
Get-Service winrm | Select -ExpandProperty RequiredServices
```

Property expanding

You can only expand one property at a time



Status	Name	DisplayName
-----	----	-----
Running	RPCSS	Remote Procedure Call (RPC)
Running	HTTP	HTTP Service

Property expanding

You can only expand one property at a time



Demo



Perfect properties



Summary



Take advantage of parameter binding

Create the output you need with custom properties

Expand properties as needed

