PowerShell ABC's - O is for Output

Joe Pruitt, 2009-09-01

Welcome to this addition of the PowerShell ABC's where you'll find 26 posts detailing a component of the PowerShell scripting language, one letter at a time. Today's letter is the letter "O" and for this letter I'll discuss the **Output** system.

PowerShell, like most other scripting languages, wouldn't be much use without an input and output system for retrieving or sending data. PowerShell has various output systems.

The Pipeline

A pipeline is a series of commands separated by the pipe operator "|". Each command in the pipeline receives an object from the previous command, performs some operation on that object, and then passes it along to the next command in the pipeline. One of the distinguishing features of Pipelines in PowerShell vs. other shells is that it uses objects through the pipeline as opposed to strings. This enables the commands in the pipeline chain to have full access to the object as well as all of it's attributes and members.

Consider the example of a simple directory listing with the Get-ChildItem Cmdlet.

```
PS D:\Dev\PoshTweet> (Get-ChildItem).GetType()
IsPublic IsSerial Name                                     BaseType
-------- -------- ----                                     --------
True     True     Object[]                                 System.Array

PS D:\Dev\PoshTweet> (Get-ChildItem)[0].GetType()

IsPublic IsSerial Name                                     BaseType
-------- -------- ----                                     --------
True     True     FileInfo                                 System.IO.FileSystemInfo

PS D:\Dev\PoshTweet> Get-ChildItem

    Directory: D:\Dev\iControl\PoshTweet

Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---         1/7/2009  12:44 PM      29167 PoshTweet.ps1

PS D:\Dev\PoshTweet> Get-Childitem | ForEach-Object -Process { Write-Host $_.CreationTime : $_.Name; }
1/9/2009 8:32:40 AM : PoshTweet.ps1
```

By having access to the object, I was able to extract the CreationTime which I wouldn't have been able to do with a traditional shell that just passed the text representation of the output of a directory listing.

The Cmdlets

Now that you've learned how to pass the output of a command to another command, you'll likely want to be able to make that output available to you. This can be done by writing the output to the console, a file, a variable, a printer, or some other form of persistent storage. The main output mechanisms can be found in the Out-* Cmdlets listed below:

> Out-Default - Send the output to the default formatter and the default output cmdlet.
>
> Out-File - Sends output to a file.
>
> Out-GridView - Sends output to an interactive table in a separate window.
>
> Out-Host - Sends the output to the command line.

> Out-Null - Deletes output instead of sending it to the console.
>
> Out-Printer - Sends output to a printer.
>
> Out-String - Sends objects to the host as a series of strings.

PowerShell provides a mechanism in which the PowerShell runtime can be embedded inside other applications. The host you are probably most familiar with is the Interactive PowerShell console but the host could just as easily be your favorite GUI application that "hosts" the PowerShell runtime.

The Out-Host Cmdlet, sends the output to the default host but you may want more control with the format of your output. This can be accomplished with the the following Cmdlets.

> Format-Custom - Uses a customized view to format the output.
>
> Format-List - Formats the output as a list of properties in which each property appears on a new line.
>
> Format-Table - Formats the output as a table.
>
> Format-Wide - Formats objects as a wide table that displays only one property of each object.
>
> Tee-Object - Saves command output in a file or variable and then displays it to the console.
>
> Write-Debug - Writes a debug message to the console.
>
> Write-Error - Writes a warning message.
>
> Write-EventLog - Writes an event to the Windows Event Log.
>
> Write-Host - Writes customized output to a host.
>
> Write-Progress - Displays a progress bar within a Windows PowerShell command window.
>
> Write-Verbose - Writes text to the verbose message stream.
>
> Write-Warning - Writes a warning message.
>
> Write-Output - Sends the specified objects to the next command in the pipeline.

Note that some of these Cmdlet's will not display anything to the console by default. The Write-Verbose and Write-Debug Cmdlet's require you to set the $VerbosePreference and $DebugPreference variables to see their output.

Redirection Operators

No shell language would be complete without allowing for easy input/output redirection. PowerShell has the Out-File Cmdlet that covers the outbound redirections, but the following traditional operators have been added that are aliases to the various formats of the Out-File Cmdlet.

Operator	Example	Results	Description
>	dir > out.txt	Contents of out.txt are replaced	Redirect pipeline output to a file, overwriting the current contents.
>>	dir >> out.txt	Contents of out.txt are appended to.	Redirect pipeline output to a file, appending to the existing content.
2>	dir nosuchfile.txt 2> err.txt	Contents of err.txt are replaced by the error messages	Redirect error output to a file, overwriting the current contents.
2>>	dir nosuchfile.txt 2>> err.txt	Contents of err.txt are appended with the error messages	Redirect error output to a file, appending to the current contents.
2>&1	dir nosuchfile.txt 2>&1	The error message is	The error messages are written

2> &1	dir nosuchfile.txt 2> &1	The error message is written to the output.	The error messages are written to the output pipe instead of the error pipe.
<	Not implemented in PowerShell v1 or v2		The operator is reserved for input redirection which is not in PowerShell v1 or v2. This will result in a syntax error.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com