Power Shell 101

**Module 4 Hands-on Activity – PowerShell in VSCode**

5/4/2019 Developed by Jin Chang, Sion Yoon

5/2/2019 Tested by Sion Yoon, Jin Chang

Center for Information Assurance (CIAE) at City University of Seattle



**Learning Outcomes**

* Learn about objects and EventLog analysis

**Resources**

* Objects

<https://www.computerworld.com/article/2954261/understanding-and-using-objects-in-powershell.html>

<https://mva.microsoft.com/en-us/training-courses/getting-started-with-powershell-3-0-jump-start-8276?l=TCQ8JQWy_404984382>

* Get-Process

<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-process?view=powershell-6>

* EventLog

<https://evotec.xyz/powershell-everything-you-wanted-to-know-about-event-logs/>

<https://www.howtogeek.com/123646/htg-explains-what-the-windows-event-viewer-is-and-how-you-can-use-it/>

**Activities**

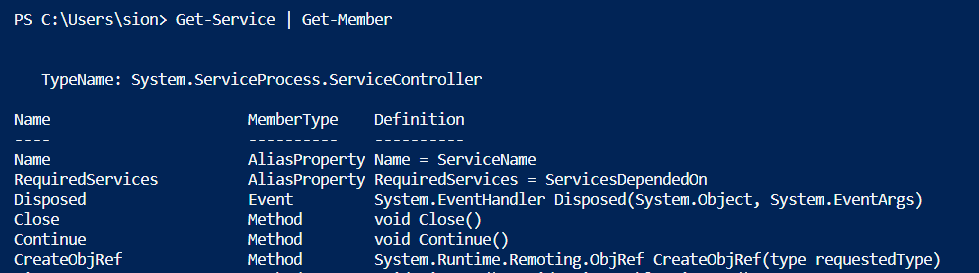
* Objects
* Get-EventLog
* Q&A

**Object**

1. Open PowerShell in VSCode
2. PowerShell is based on the .NET Framework, meaning that PowerShell can be considered a programming language. **Objects** are known quantities of something that programming languages can use and perform computations on. There are two types that objects: properties and methods. **Properties** are attributes of what the .NET objects are representing and **methods** describe the type of actions that the .NET object can undertake.

In PowerShell, you can see an object’s properties and methods by piping the output of a cmdlet with Get-Member. Let’s test it out with get-service cmdlet. Type:

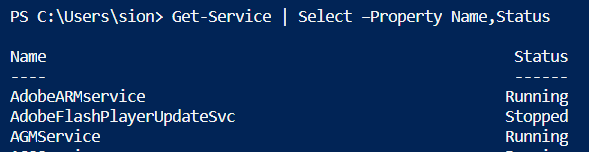
% **get-service | get-member**



This provides you with a list of properties and methods for get-service. You can see the start and stop methods that we used for Get-Service in last week’s activity.

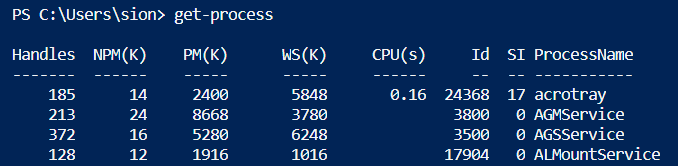
1. What if you want to select properties that you want to see such as name and status of get-service properties? Type:

% **get-service | select –property name,status**



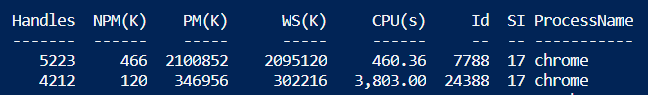
1. Let’s continue with objects with a different cmdlet, Get-Process, which is a cmdlet that gets the processes on a local computer. Type:

% **get-process**



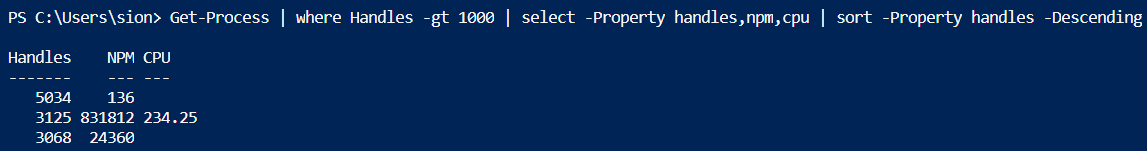
1. Since it is a long list, what if you want to get the processes where the handles is greater than 1000. Type:

% **get-process | where handles –gt 1000**



1. What if you want to select to see only the handles, NPM, and CPU and sort the processes by handles in a descending order? Type:

% **get-process | where handles –gt 1000 | select –property handles,npm,cpu |sort –property handles –descending**

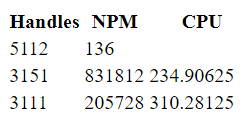


1. What if you want convert this into an html file and have an out file for it. Type:

% **get-process | where handles –gt 1000 | select –property handles,npm,cpu |sort –property handles –descending | convertto-html | out-file c:\users\sion\test.htm**

Direct the path to the directory that you are on. You can check if it worked by typing the directory of the file:

**% c:\users\sion\test.htm**



1. Let’s now try something different to compare objects. The following command produces a snapshot (.XML file) of all the process objects that are running in at the very moment.

% **get-process | export-clixml c:\users\sion\process.xml**

Try changing the process that are running by opening the following apps:

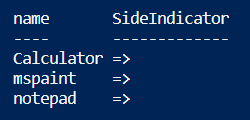
**% calc**

**% mspaint**

**% notepad**

To compare the process from the process.xml to the current process. Type:

**% Compare-Object -ReferenceObject(import-clixml process.xml) -DifferenceObject(Get-Process) -Property name**



Since the reference set is on the left and the difference set on the right, this means that the calculator, mspaint, and notepad is pointing to the right with an arrow towards the difference set. This can be used for change management and change reporting.

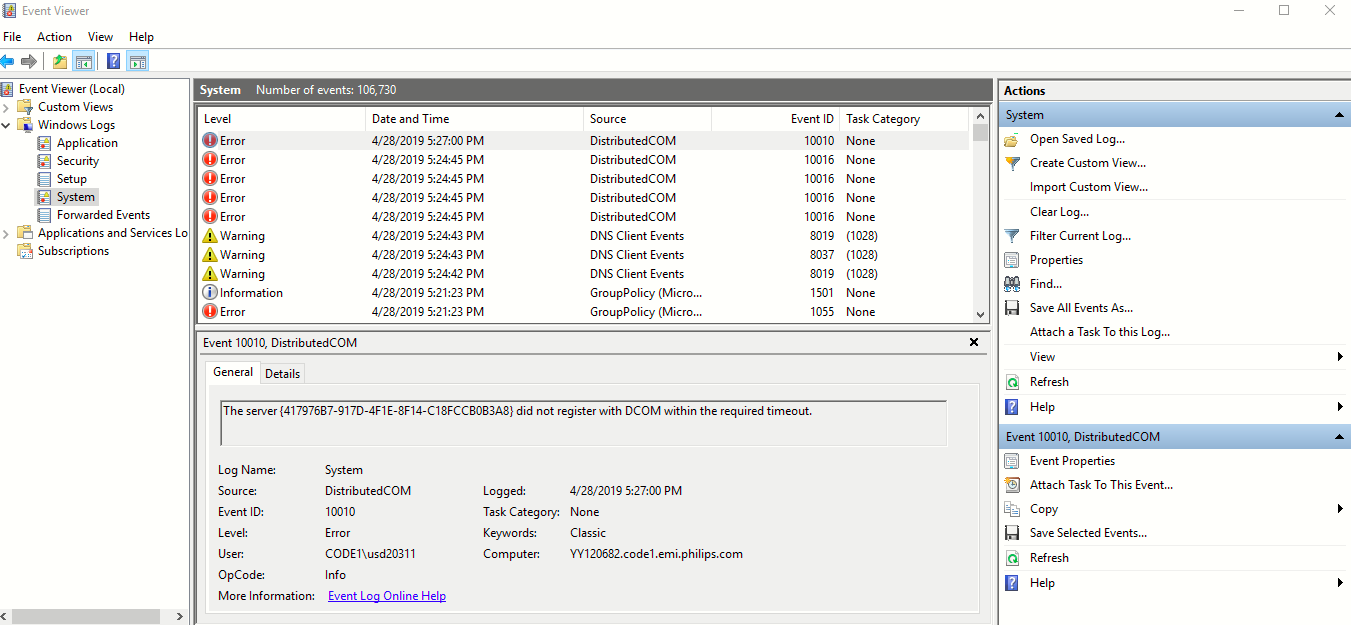
**EventLog Analysis Using Cmdlets**

1. Now, let’s examine Event Log in your computer. The event viewer (command name: **eventvwr**) is designed to help system admin keep tabs on their computers and troubleshoot relevant problems. If there is not a problem with your computer, the error in here are unlikely to be important. For example, you will often see errors that indicate a program crashed at a specific time which may have been weeks ago or that a service failed to start with Windows, but was likely started on a subsequent attempt. And we will learn very useful PowerShell cmdlets dealing with Event Logs. The user is created, windows update is installed time to time, service is restarted sometimes or application is crashed on you. Most of these events are logged in Event Logs. Four main categories;

* **Application**: events related to software installed
* **Security**: contains events related to the security such as permission changes
* **Setup**: performance events or events logged during installations
* **System**: shutdown, restart, group policy changes, etc.
* **Forward Events**: log depository to store event logs that are forwarded from different computers in the network.

At the command line, type:

% **eventvwr**



There are different categories available per each event.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Log Name**: Name of event log to view | | **Source**: Source of event e.g. app. name | | **Event ID**: Paired with Log Name | | **Level**: Six levels.  {Verbose, Informational, Warning,  Error, Critical, LogAlways} | | **User**: Owner of the event | | **Logged**: Time the event was logged | | **Task Category**: Event source | | **Keywords**: 9 Keywords  {AuditFailure, AuditSuccess, Classic, CorrelationHint, Response Time, SQM, WDI Context, WDI Diag.} | | **Computer**: Computer name reported the event | |  |

1. To displays available log names in the event viewer, type:

**% Get-EventLog –List**

1. Type the following cmdlet to count the number of log names.

**% (Get-EventLog –List).Count**

1. To get a list of events in the system eventlog, type the following. You can break out early with ‘ctrl-c’ while being displayed.

**% Get-EventLog –LogName System**

1. Type the following cmdlet, and let’s take a closer look at the headers.

**% Get-EventLog –LogName System –newest 1**

Note that “newest X” shows the first X incidents.

|  |
| --- |
|  |

1. We have Index, Time, EntryType, Source, InstanceID and Message. As we have learned from above, these are property objects in the cmdlet. We can very effectively parse out if we can utilize these object names. For instance, can we parse out only the events that have “Error” EntryType from “DCOM” source? Type and run the following cmdlet.

**% Get-EventLog –LogName System –EntryType Error –Source DCOM –newest 10**

|  |
| --- |
|  |

1. To detect brute-force, dictionary, and other password guess attacks, which are characterized by sudden spike in failed logons. To detect abnormal and possibly malicious internal activity, like a logon attempt from a disabled account or unauthorized workstation, users logging on outside of normal working hours. Therefore, it is a very useful tool to find out if there is anyone (suspicious attacker) who attempted to log in your computer ended up failed. You can of course look at the event, but it will be way easier if you think about performing this search over 1,000 computers in a network. So, using the PowerShell, we can confidently display the logon failure listed in a given Event Log. The following 2 cmdlets will provide the attempts. You would first need administrator privilege to perform the operations. Run the command below in prior to executing the cmdlets.

**% Start-Process PowerShell –Verb RunAs**

**% Get-EventLog –LogName Security**

And we can narrow down to the failures.

**% Get-EventLog –LogName Security –instanceId 4625**

**instanceID(4625) = 4625** (viewed in Windows **Event** Viewer) documents user’s logon failure to a local computer if there is any.

**% Get-EventLog –LogName Security –instanceId 5061**

**instanceID(5061) = 5061** (viewed in Windows **Event** Viewer) documents every cryptographic operations on to a local computer.

1. To find any substring search listed in the Message object, type the following example.

**% Get-EventLog –LogName Security –Message “crypto\*”**

1. To Count the number of instances, then use the member object called **.Count**

**% (Get-EventLog –LogName Security –Message “crypto\*”).Count**

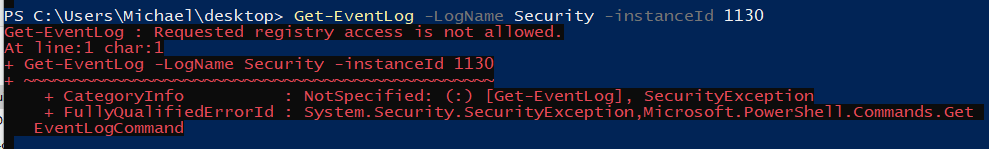
**Q&A**

1. **Which log name would have the group policy object (GPO) related events? Hint: see activity #9.**

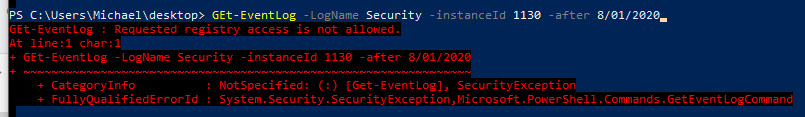
System

1. **Find a cmdlet to display GPO start up script failures for instance ID 1130. Get the screen capture of your command and its output and paste here. [If there is no listing for instance ID 1130, then find any InstanceID of your choice (ex. InstanceID 10016) and answer the following questions].**

Get-EventLog cmdlet was working at first but for some reason it stopped working.



1. **Provide the cmdlet that displays the events (for the instance ID that you used in question 2) that occurred for the last 3 months. Get the screen capture of your command and its output and paste here. Hint: use the “-after” parameter (ex. “-after 2/01/2019)”.**



1. **Your supervisor wants to know “*how many times*” the events (for the instance ID that you used in question 2) actually occurred in your computer for the last 3 months. Get the screen capture of your command and its output and paste here.**



1. **Your supervisor wants to know the events (for the instance ID that you used in question 2) that occurred in your computer for the last 3 months but wants to see only the property index and time generated. Get the screen capture of your command and its output and paste here.**
2. **Your supervisor wants to know the events (for the instance ID that you used in question 2) that occurred in your computer for the last 3 months but want to see only the property index and time generated and want it to be sorted by the time generated (oldest to newest). Get the screen capture of your command and its output and paste here.**
3. **Your supervisor wants to see what you have done on question 6 into on the web. Get the screen capture of your command and the screen capture of the output on your web browser and paste here.**