Windows Event Logs

"Event logs record events taking place in the execution of a system to provide an audit trail that can be used to understand the activity of the system and to diagnose problems. They are essential to understand the activities of complex systems, particularly in applications with little user interaction (such as server applications)."

Event Viewer

- Windows Event Logs are not text files that can be viewed using text editor
- events are stored in a proprietary binary format with a .evt or .evtx extension
- log files with .evtx extension typically reside in C:\Windows\System32\winevt\Logs

Elements of a Windows Event Log

- System Logs : events associated with the OS. They may includes information about hardware changes, device drivers, system changes, and other activities related to the device.

- Security Logs : events connected to logon and logoff activities on a device.

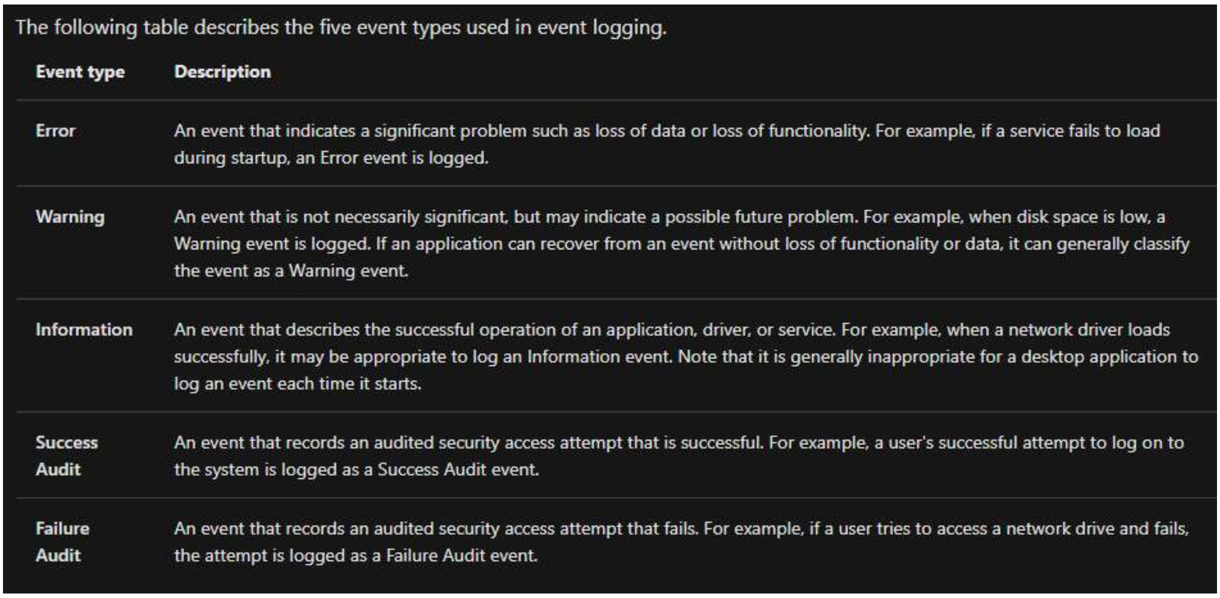
- Application Logs : events related to applications installed on the system.
(application errors/events/warnings)

- Directory Service Events : AD changes and activities are recorded in these logs, mainly on domain controllers.

- DNS Event Logs : DNS servers use these logs to record domain events and to map out.

- Custom Logs : events are logged by applications that require custom data storage. This allows applications to control the log size/attach other parameters, such as ACLs.

Event logs categorised into types



The following table describes the five event types used in event logging.

Event type	Description
Error	An event that indicates a significant problem such as loss of data or loss of functionality. For example, if a service fails to load during startup, an Error event is logged.
Warning	An event that is not necessarily significant, but may indicate a possible future problem. For example, when disk space is low, a Warning event is logged. If an application can recover from an event without loss of functionality or data, it can generally classify the event as a Warning event.
Information	An event that describes the successful operation of an application, driver, or service. For example, when a network driver loads successfully, it may be appropriate to log an Information event. Note that it is generally inappropriate for a desktop application to log an event each time it starts.
Success Audit	An event that records an audited security access attempt that is successful. For example, a user's successful attempt to log on to the system is logged as a Success Audit event.
Failure Audit	An event that records an audited security access attempt that fails. For example, if a user tries to access a network drive and fails, the attempt is logged as a Failure Audit event.

3 main ways of accessing these event logs within a Windows system:

1. Event-Viewer (GUI-based application)
2. Wevtutil.exe (command-line tool)
3. Get-WinEvent (PowerShell cmdlet)

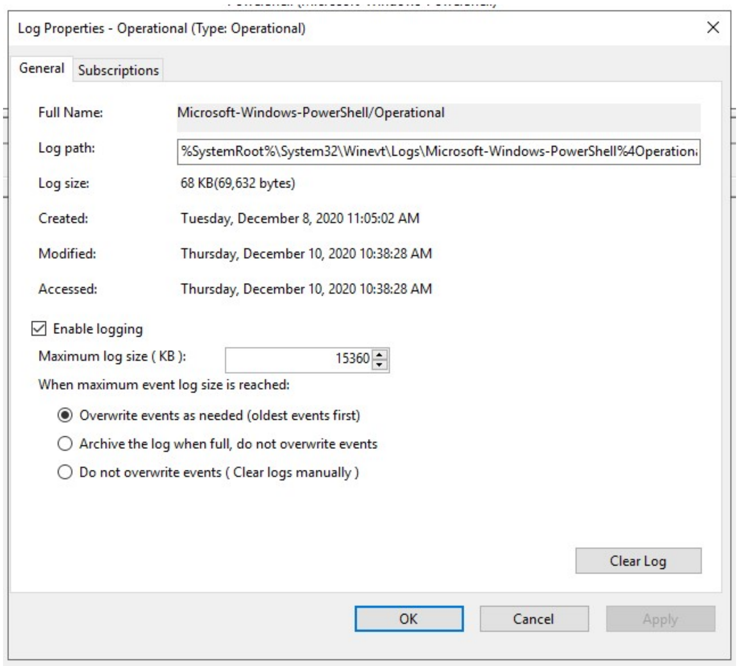
Event Viewer

Following section is the **Applications and Services Logs**

Microsoft > Windows > PowerShell > Operational

(PowerShell will log operations from the engine, providers, and cmdlets to the Windows event log)

Log Properties



Log Properties - Operational (Type: Operational)

Full Name: Microsoft-Windows-PowerShell/Operational

Log path: %SystemRoot%\System32\Winevt\Logs\Microsoft-Windows-PowerShell%4Operation

Log size: 68 KB(69,632 bytes)

Created: Tuesday, December 8, 2020 11:05:02 AM

Modified: Thursday, December 10, 2020 10:38:28 AM

Accessed: Thursday, December 10, 2020 10:38:28 AM

☑ Enable logging

Maximum log size (KB): 15360

When maximum event log size is reached:

● Overwrite events as needed (oldest events first)

○ Archive the log when full, do not overwrite events

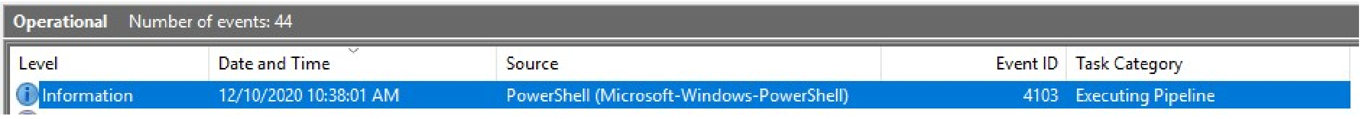
○ Do not overwrite events (Clear logs manually)

Within Properties:
- Log path/location
- Log size
- Log creation
- Modified
- Last accessed

Log Rotation
- Maximum set log size
- What action to take once the criteria is met

Clear Log

> attackers will likely attempt to clear the logs to go undetected



Operational Number of events: 44

Level	Date and Time	Source	Event ID	Task Category
Information	12/10/2020 10:38:01 AM	PowerShell (Microsoft-Windows-PowerShell)	4103	Executing Pipeline

- Level: highlights the log recorded type
- Date/Time : records the time at which the event was logged
- Source : name of the software that logs the event is identified > PowerShell from above screenshot
- Event ID : predefined numerical value that maps to a specific operation or event based on log source.

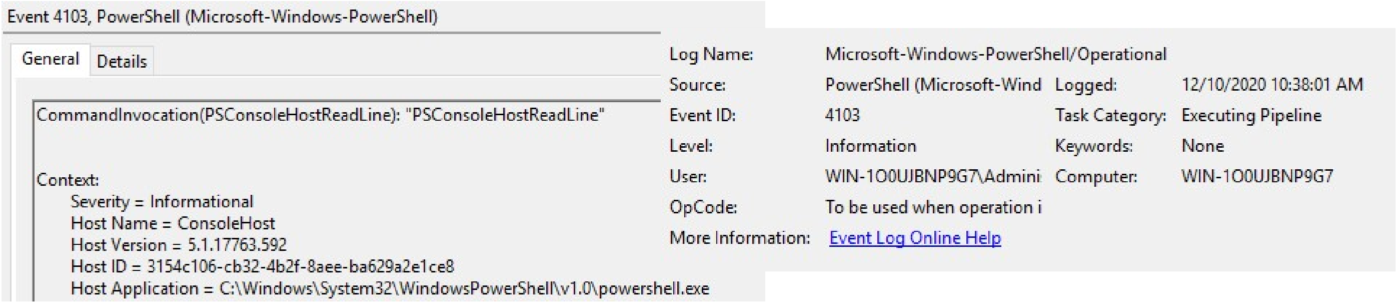
Event ID 4103 > related to Executing Pipeline but will have a different meaning in another event log

- Task Category : helps organize events so the Event Viewer can filter

More information is displayed at the bottom when selecting an event.

Section has 2 tabs: General and Details

- General : default view, and the rendered data is displayed
- Details : friendly view / XML view



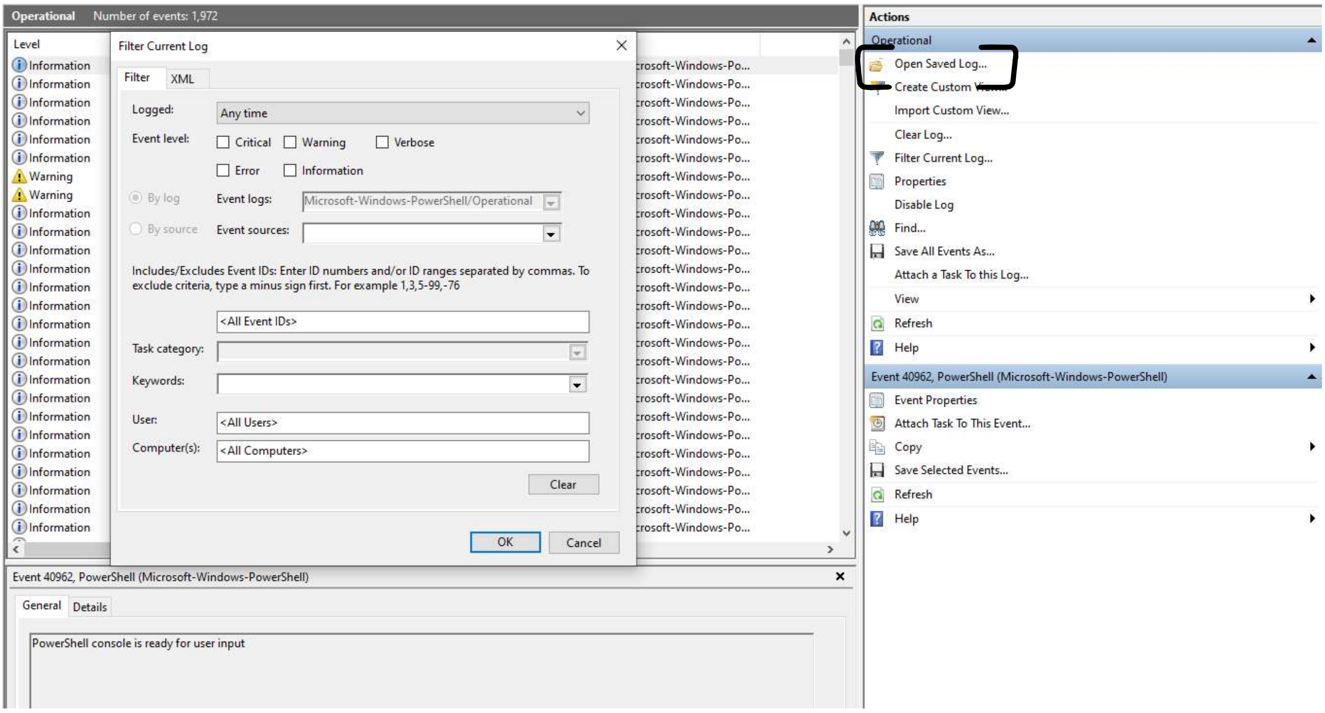
Event 4103, PowerShell (Microsoft-Windows-PowerShell)

```
CommandInvocation(PSConsoleHostReadLine): "PSConsoleHostReadLine"


Context:
     Severity = Informational
     Host Name = ConsoleHost
     Host Version = 5.1.17763.592
     Host ID = 3154c106-cb32-4b2f-8aee-ba629a2e1ce8
     Host Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Log Name:	Microsoft-Windows-PowerShell/Operational		
Source:	PowerShell (Microsoft-Wind	Logged:	12/10/2020 10:38:01 AM
Event ID:	4103	Task Category:	Executing Pipeline
Level:	Information	Keywords:	None
User:	WIN-1O0UJBNP9G7\Admini	Computer:	WIN-1O0UJBNP9G7
OpCode:	To be used when operation i		
More Information:	Event Log Online Help		

Actions pane

- can open a saved log within the Actions pane



Operational Number of events: 1,972

Filter Current Log

Logged: Any time

Event level: □ Critical □ Warning □ Verbose □ Error □ Information

● By log Event logs: Microsoft-Windows-PowerShell/Operational

○ By source Event sources:

Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76

<All Event IDs>

Task category:

Keywords:

User: <All Users>

Computer(s): <All Computers>

Event 40962, PowerShell (Microsoft-Windows-PowerShell)

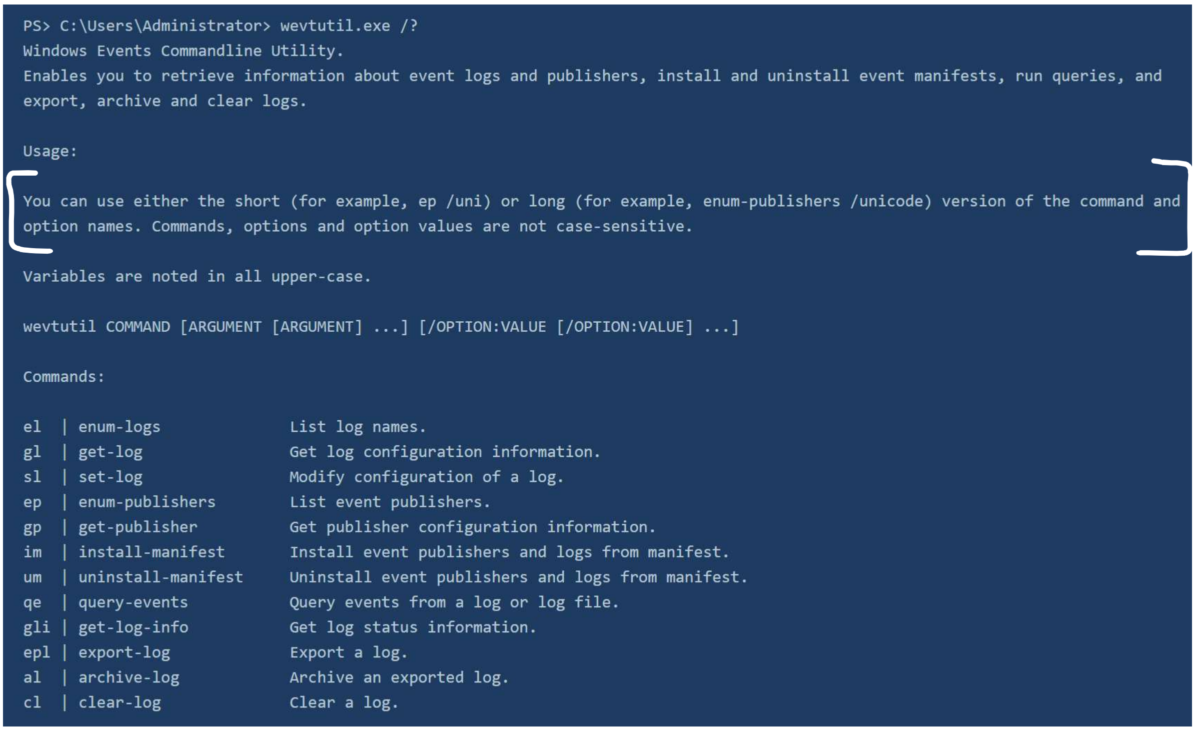
PowerShell console is ready for user input

View Event Logs from another computer

Right click Event Viewer (Local) > Connect to another computer...

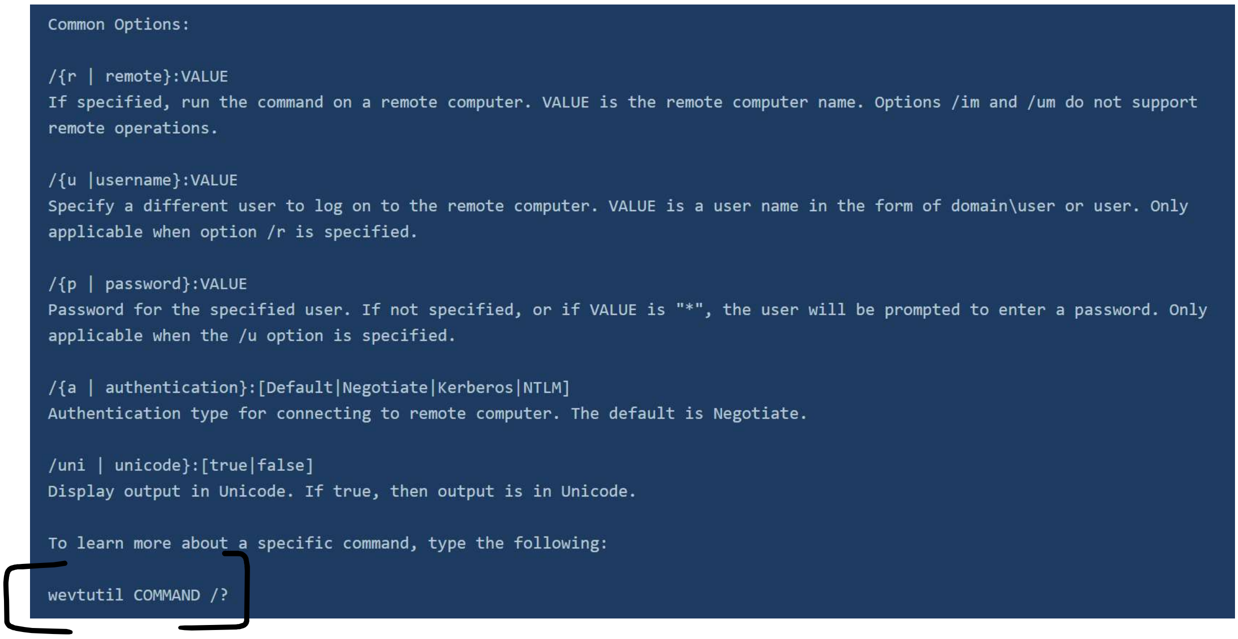
wevtutil.exe
"enables you to retrieve information about event logs and publishers. You can also use this command to install and uninstall event manifests, to run queries, and to export, archive, and clear logs."

- create scripts to query events logs via the command line/PowerShell



```
PS> C:\Users\Administrator> wevtutil.exe /?
Windows Events Commandline Utility.
Enables you to retrieve information about event logs and publishers, install and uninstall event manifests, run queries, and
export, archive and clear logs.

Usage:

You can use either the short (for example, ep /uni) or long (for example, enum-publishers /unicode) version of the command and
option names. Commands, options and option values are not case-sensitive.

Variables are noted in all upper-case.

wevtutil COMMAND [ARGUMENT [ARGUMENT] ...] [/OPTION:VALUE [/OPTION:VALUE] ...]

Commands:

el  | enum-logs             List log names.
gl  | get-log               Get log configuration information.
sl  | set-log               Modify configuration of a log.
ep  | enum-publishers       List event publishers.
gp  | get-publisher         Get publisher configuration information.
im  | install-manifest      Install event publishers and logs from manifest.
um  | uninstall-manifest    Uninstall event publishers and logs from manifest.
qe  | query-events          Query events from a log or log file.
gli | get-log-info          Get log status information.
epl | export-log            Export a log.
al  | archive-log           Archive an exported log.
cl  | clear-log             Clear a log.
```

Common options that can be used with Windows Events Utility

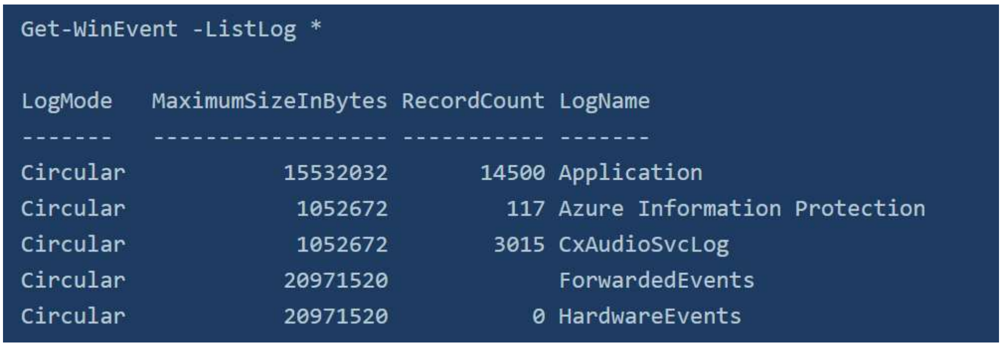


```
Common Options:

/{r | remote}:VALUE
If specified, run the command on a remote computer. VALUE is the remote computer name. Options /im and /um do not support
remote operations.

/{u |username}:VALUE
Specify a different user to log on to the remote computer. VALUE is a user name in the form of domain\user or user. Only
applicable when option /r is specified.

/{p | password}:VALUE
Password for the specified user. If not specified, or if VALUE is "*", the user will be prompted to enter a password. Only
applicable when the /u option is specified.

/{a | authentication}:[Default|Negotiate|Kerberos|NTLM]
Authentication type for connecting to remote computer. The default is Negotiate.

/uni | unicode}:[true|false]
Display output in Unicode. If true, then output is in Unicode.

To learn more about a specific command, type the following:

wevtutil COMMAND /?
```

Get-WinEvent
"gets events from event logs and event tracing log files on local and remote computers."
- PowerShell

Example 1: Get all logs from a computer

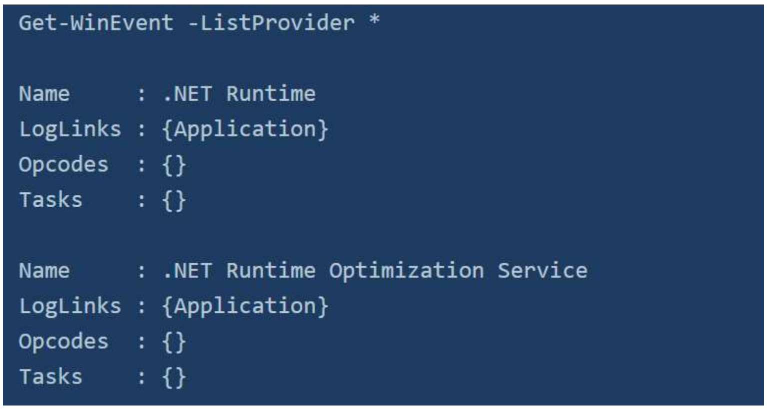
- obtain all event logs locally



```
Get-WinEvent -ListLog *

LogMode   MaximumSizeInBytes RecordCount LogName
-------   ------------------ ----------- -------
Circular            15532032       14500 Application
Circular             1052672         117 Azure Information Protection
Circular             1052672        3015 CxAudioSvcLog
Circular            20971520             ForwardedEvents
Circular            20971520           0 HardwareEvents
```

Example 2: Get event log providers and log names

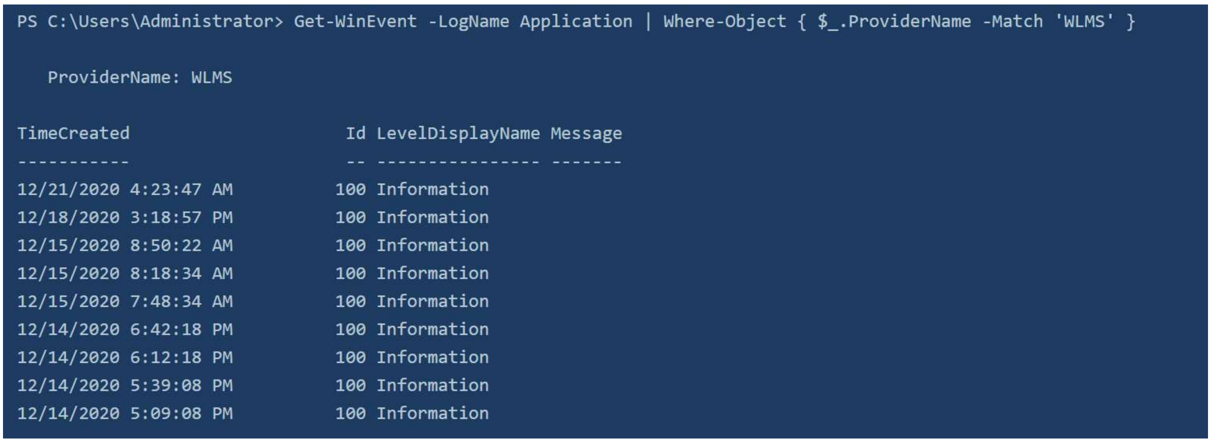
- obtain event log providers and their associated logs
- Name > provider
- LogLinks > log written to



```
Get-WinEvent -ListProvider *

Name     : .NET Runtime
LogLinks : {Application}
Opcodes  : {}
Tasks    : {}

Name     : .NET Runtime Optimization Service
LogLinks : {Application}
Opcodes  : {}
Tasks    : {}
```

Example 3: Log filtering

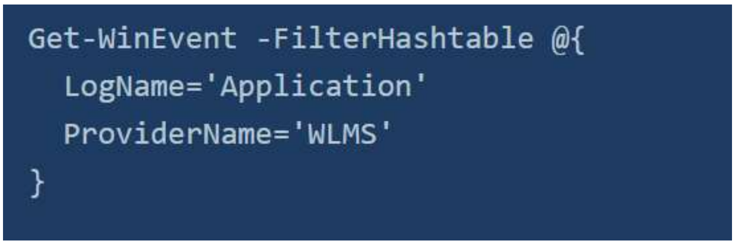
- select events from an event log
- filter logs using the **Where-Object** cmdlet



```
PS C:\Users\Administrator> Get-WinEvent -LogName Application | Where-Object { $_.ProviderName -Match 'WLMS' }

   ProviderName: WLMS

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
12/21/2020 4:23:47 AM          100 Information
12/18/2020 3:18:57 PM          100 Information
12/15/2020 8:50:22 AM          100 Information
12/15/2020 8:18:34 AM          100 Information
12/15/2020 7:48:34 AM          100 Information
12/14/2020 6:42:18 PM          100 Information
12/14/2020 6:12:18 PM          100 Information
12/14/2020 5:39:08 PM          100 Information
12/14/2020 5:09:08 PM          100 Information
```

When working with large events logs, it’s inefficient to send objects down the pipeline to Where-Object command.

- FilterHashtable parameter is recommended to filter event logs

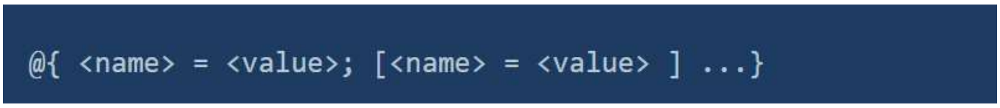


```
Get-WinEvent -FilterHashtable @{
  LogName='Application'
  ProviderName='WLMS'
}
```

Guidelines for defining a hash table are:

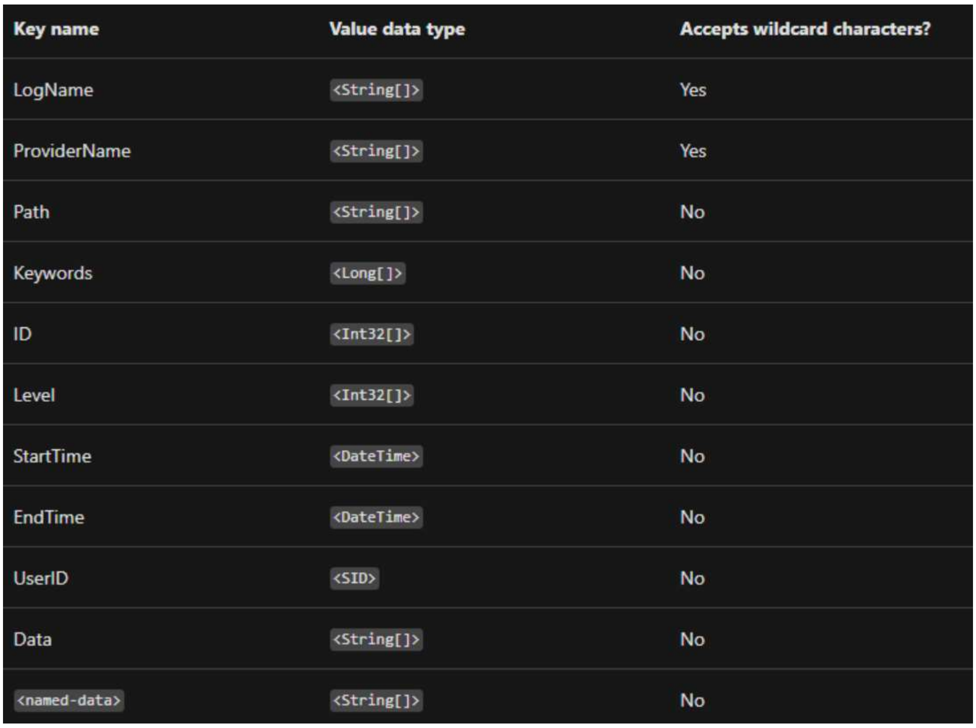
- Begin the hash table with an @ sign.
- Enclose the hash table in braces {}
- Enter one or more key-value pairs for the content of the hash table.
- Use an equal sign (=) to separate each key from its value.

Syntax:



```
@{ <name> = <value>; [<name> = <value> ] ...}
```

Table that displays the accepted key/value pairs for the Get-WinEvent FilterHashtable parameter:



Key name	**Value data type**	**Accepts wildcard characters?**
LogName	`<String[]>`	Yes
ProviderName	`<String[]>`	Yes
Path	`<String[]>`	No
Keywords	`<Long[]>`	No
ID	`<Int32[]>`	No
Level	`<Int32[]>`	No
StartTime	`<DateTime>`	No
EndTime	`<DateTime>`	No
UserID	`<SID>`	No
Data	`<String[]>`	No
`<named-data>`	`<String[]>`	No