Advanced Scripting   
Script - Manipulating Text Data

Last Updated: 6/30/2021 5:51 PM Version 1  
Document Prepared for: CIT361 Student

# Overview

A common task in PowerShell is to manipulate textual data and either create new data or reports from the data. In this lab you will write a script that contains two functions. The first function when run will get the machines ethernet addresses, look up the vendors name. The vendors name is in a text file. The second function will load a list of song data and output a formatted report that organizes the songs by album then by song.

# Requirements:

The required data files are in data directory of the psfiles download we have been using. You can get a copy here <http://cf.esage.com/psfiles.zip>

1. Create a script named Week8.ps1
2. Include the following author comment block

# <# Program Name : <program> Date: <datecreated> Author: <yourname> Corse: CIT361 I, <yourname>, affirm that I wrote this script as original work completed by me. #>

1. Function 1
   1. Create a function Named Get-MACVendor that returns the Vendors Full Name.
   2. The script takes one optional argument $MACAddress and one Mandatory argument $DatabasePath
      1. The DatabasePath accepts filename of the database to use. Don’t assume a directory location.
      2. Throw an error if the database file can’t be found.
   3. If a MAC address is specified return the Vendor of that MAC address
   4. If the MAC address is not supplied discover the MAC addresses of the network adapters on the computer and look up the Vendor of each address. Return Each Vendor Name.
      1. You can get the MAC address on windows With **Get-NetAdapter**. On Linux you might use the **ip address** command look for the ether section.
   5. The script must work on Windows and Linux or OSX
   6. The MAC address database is in the psfiles/data/MACDatabase.txt
2. Function 2
   1. Create A function named Format-Songs that retrieves and formats the song data from psfiles/data/RushSongs.txt as follows
   2. The script takes one mandatory and one optional arguments
      1. $DatabasePath (madatory) is the file that contains the song information
      2. $Path is the file to save the output to, if missing just return report to output stream.
      3. Throw an error if the database can’t be found or the file can’t be saved.
   3. Group songs by album then sort alphabetically by title.
   4. (Stretch goal) List albums by year newest to oldest in the report.
   5. Example Output

R40 Live (2015)

The Story so Far

Clockwork Angels Tour (2013)

Drumbastica

Here It Is!

Peke's Repose

The Percussor

Clockwork Angels (2012)

BU2B  
…

# Hints

* Use PowerShell interactively to learn how to manipulate the data before writing the script.
* You’re not coding in Java or C use the strengths of PowerShell. If you code looks like you would do it in Java or C you are doing it wrong.
* The vendor portion of a MAC address is the first 3 bytes of the address, so if the address is 11:22:33:44:55:66 the vendor portion is 11:22:33

### Get-MACVendor

* Import-csv may not be the easiest way to work with the MAC database
* Notice that Windows usually uses - in MAC addresses where Linux uses : The database uses :
* $PSVersionTable, $IsLinux

### Format-Songs

* Consider adding a property to the object returned by group-object to add the year to sort by.

# Deliverable:

Upload your script to I-Learn

# Grading Rubric:

|  |  |
| --- | --- |
| Element | Points |
| Comment block | 1, -10(if missing) |
| MAC Vendor Lookup | 3 |
| Discovered Addresses on Windows and Linux | 3 |
| Song Report | 3 |
| Error Handling | 1 |
| Extra Credit, albums sorted by year | 1 |
| Total | 10 |