Advanced Scripting   
Module Manifests

Last Updated: 4/18/2024 10:45 AM Version 1  
Document Prepared for: CYBER360 Student

# Name Click here to enter name ID Click here to enter id

# Instructions

Save a copy of this document. Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

Now lets create a module manifest to control the loading of our module.

# Requirements

PowerShell (any platform)  
MyUtilities.psm1 from the previous exercise.

# Setup

# Task 1—Create a Module Manifest

The easiest way to create a manifest is to use the **New-ModuleManifest** cmdlet. You can pass a lot of information to the cmdlet to set properties in the manifest, or you can modify the manifest yourself. Let’s use a combination of both.

## Steps

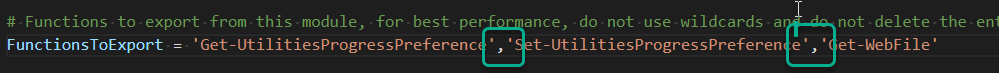
1. Create a manifest for your module. The manifest should be named the same as your module, but with a **.psd1** extension. Also you need to identify the root module; it will be your **.psm1** file.
2. Change your working directory to where your **MyUtilities.psm1** file is.
3. Create the manifest.   
   New-ModuleManifest -Path MyUtilities.psd1 -RootModule MyUtilities.psm1
4. Open the Manifest file in your editor and view the results. Notice several default values have already been added.
   1. What is the **ModuleVersion**? Click or tap here to enter text.
   2. The **GUID**? Click or tap here to enter text.
   3. The **Author**? Click or tap here to enter text. Where do you suppose that value came from? Click or tap here to enter text.
5. Notice most of the items are commented out. If you would like to set one you can uncomment it.
6. Partway down you will find a section that exports items. You must modify this section to only export your three functions. Modify the file as follows:   
   A screen shot of a computer

   Description automatically generated
7. Save the manifest file.

# Task 2—Test Your Manifest

You can perform some automatic validation on the manifest file with the **Test-ModuleManifest** cmdlet.

## Steps

1. Test the manifest.   
   **Test-ModuleManifest MyUtilities.psd1**
2. Oh oh, something is wrong. The problem is that we should have used commas rather than semicolons when we created the array list for **FunctionsToExport**.
3. Fix the manifest:   
   
4. Save the manifest and test it again. It should pass, if not fix until it does.

# Task 3—Manually Load your Module via Manifest

## Steps

1. If your module is currently loaded, unload it.
2. Load the module via manifest:   
   Import-Module MyUtilities.psd1
3. See what commands are available.
4. Test it out.

# Task 4—Install Module

## Steps

1. Create a folder named **MyUtilities** in your personal module folder. If you don’t remember where that is, look at your **$env:PSModulePath**.
2. Copy your **MyUtilites.psm1** and **MyUtilities.psd1** to that folder.
3. Start a new PowerShell instance.
4. Test. Enter:   
   **Get-UtilitiesProgressPreference**
5. It should work. If not, troubleshoot and fix.

# Wrap-up

Copy the entire contents of your Manifest file here:

Click or tap here to enter text.

# Deliverable

Upload this document with completed answers to I-Learn Canvas.