Advanced Scripting   
Authoring Modules

Last Updated: 11/11/2021 10:41 AM Version 1  
Document Prepared for: CIT361 Student

# Name Zach Lucas ID 895649438

# 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 you will create a module and add a few utility functions.

# Requirements

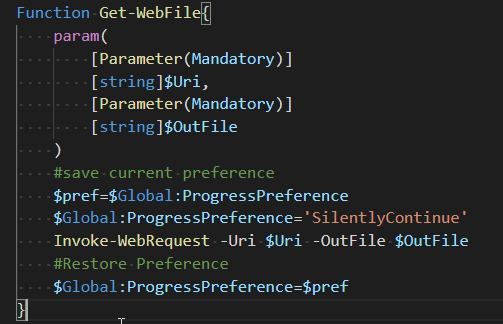
# Setup

# Task 1—Create a Module

You will create a module file that has a utility to download files from the internet faster. You may have noticed that PowerShell’s Invoke-Webrequest cmdlet is fairly slow downloading large files. As it turns out this is because of the progress bar that is show. You will add a function that will turn off the progress bar before it downloads the file to speed things up, then it will return the progress bar to its previous state. The $ProgressPreference variable controls if the progress bar is shown or not.

## Steps

1. Create a file named MyUtilities.psm1
2. Add the following code



1. Save your file and give it a test. To test the module you can’t just run it from vscode. You need to import the module to the session to try it out. Use the Import-Module command   
   Import-Module MyUtilities.psm1
2. View the commands in your module  
   Get-Command -Module MyUtilities
3. Try it out. It should download without the progress bar.

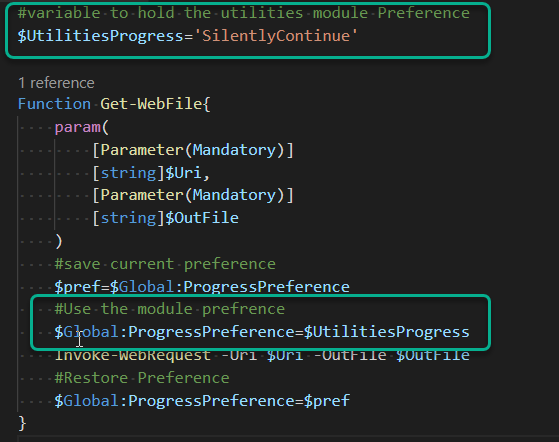
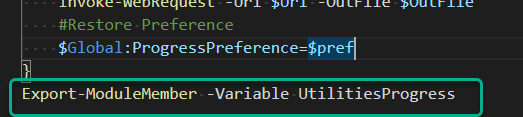
Get-WebFile cf.esage.com/psfiles.zip junk.zip

1. If you get an error fix you code and try again. To try again you must:
   1. Save your code
   2. Load your module again. Since your module is already loaded you need to either
      1. Unload your module
      2. Reload your module
   3. Or
      1. Or if you can use the -force option for Import-Module

# Task 2—Add a Script Level Variable

Maybe you want to add a method for the user to set a preference for the progress bar in the Utilities module. To do that you will need a variable to store the preference in.

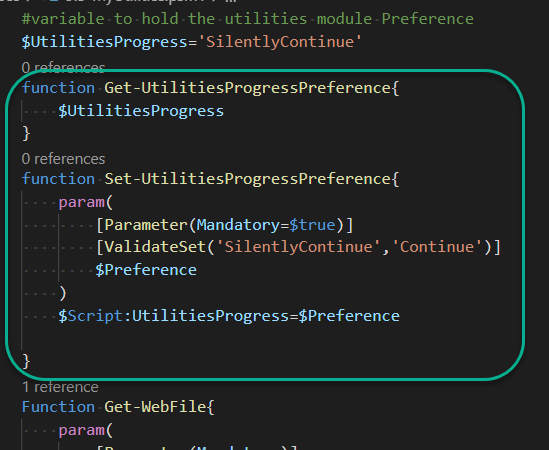
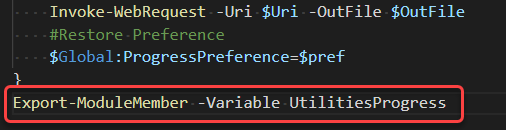
## Steps

1. Make the following changes to your module file.  
   
2. Save the file, reload the module.
3. What happens if you enter  
   $UtilitiesProcess in the shell?
4. Script level variables are not exposed by default. If you want to expose a script level variable you can tell the module to do so with the Export-ModuleMember cmdlet. Add the following line of code at the bottom of the module  
   
5. Save and re-import the module
6. Enter  
   $UtilitiesProgress
   1. What was returned? SilentlyContinue
7. Get a list of the commands in the module. Is Get-WebFile listed?
8. Once you use the Export-ModuleMemeber cmdlet all members are hidden except those explicitly exported.

# Task 3—Provide Commands to Change the Module Preference

Allowing direct access to the $UtilitiesProgress variable is probably not a good idea, for one thing the user could easily enter an invalid option. In this task you will add code to modify the $UtilitiesProgress Variable and View the $UtilitiesProgress value.

## Steps

1. Add the functions Get-UtilitiesProgressPreference and function Set-UtilitiesProgressPreference  
   
2. Remove the Export-ModuleMember line from the module.  
   
3. Save and re-import.
4. Test the new module using the new commands to set the module preferences.

# Wrap-up

Enter your module code here

*#variable to hold the utilities module preference  
$UtilitiesProgress='SilentlyContinue'  
  
function get-UtilitiesProgressPreference {  
 $UtilitiesProgress  
}  
  
function Set-UtilitiesProgressPreference {  
 param (  
 [parameter(Mandatory=$true)]  
 [ValidateSet('SilentlyContinue','Continue')]  
 $Preference  
 )  
 $Script:UtilitiesProgress=$Preference  
}  
  
function Get-WebFile {  
 param (  
 [parameter(Mandatory)]  
 [string]$Uri,  
 [parameter(Mandatory)]  
 [string]$Outfile  
 )  
 #save current preference  
 $pref=$Global:ProgressPreference  
 #Use the module preference  
 $Global:ProgressPreference=$UtilitiesProgress  
 #use the module preference  
 $Global:ProgressPreference='SilentlyContinue'  
 Invoke-WebRequest -Uri $Uri -OutFile $Outfile  
 #retore preference  
 $Global:ProgressPreference=$pref  
}  
#Export-ModuleMember -Variable UtilitiesProgress*

# Deliverable

Upload this document with completed answers to i-learn.