### License

Copyright (c) Microsoft Corporation.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED \*AS IS\*, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

##### MECMOfflineUpdateGUI.ps1 Script (Service Connection Tool GUI)

To make updates easier I have prepared a PowerShell script that has a GUI to perform the updates. As with the Service Connection Tool, this needs to be run as Administrator.

The GUI has three tabs. The two tabs that have a red background are run on the Site System Server with Service Connection Point role installed. The green tab is run on the internet connected computer.

When you first run the script, it looks for the Service Connection Tool, if it cannot find it, it will ask you to locate the folder. First it looks to see if the $env:SMS\_LOG\_PATH\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool\ServiceConnectionTool.exe exists, if it does not it looks to see if the “$PSScriptRoot\ServiceConnectionTool\ ServiceConnectionTool.exe” file exists ($PSScriptRoot is the location the script is run from). If it finds the ServiceConnectionTool.exe file it will use that folder otherwise it will prompt for it.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Select the folder that contains the ServiceConnectionTool.exe (NOTE: you need the version that is for your environment. It is best to use the one on the Site Server in the cd.latest\SMSSETUP\Tools\ServiceConnectionTool folder.

###### Export Usage Data Cab

Note: the PowerShell script needs to be run as Administrator

The first tab is the Export Usage Data Cab. This needs to be run on the Site System Server with the Service Connection Point installed on it. Next pick the location to save the package. The tool will automatically copy the ServiceConnectionTool folder as well as the script you are running to it and it will create the UsageData subfolder with the UsageData.cab in it.

Graphical user interface, text, email, website

Description automatically generated

If you look at the PowerShell out put you will see what was copied.

Graphical user interface, text, application, chat or text message

Description automatically generated

If you look in the folder that you selected, you will see what is now in there.

Graphical user interface, text, application, email

Description automatically generated

If you look in the UsageData folder you will see 2 files:

1. The <ServerName>-UsageData.cab Which is the file that is uploaded to Microsoft
2. The <ServerName>-UsageData.csv which is a text file output of what is in the cab file.

The cab file only has a list of file and the version of those files. This helps Microsoft cloud services make sure you get the updates you need.

Graphical user interface, text, application, email

Description automatically generated

###### Download Updates

Move the folder you selected to save everything in, in the example it was the Updates folder. This needs to be moved to a location with as short of as possible full path name. You need it to be less than 40 characters. I usually will put the folder in C:\Temp so it will be C:\Temp\Updates.

Again the script you need to run was put in the folder you copied over, so in this case it is in C:\Temp\Updates folder. ***Remember to run it As Administrator***.

If you copied the entire folder over it should have included the ServiceConnectionTool folder and will not prompt you for the location when you run the tool. It should also detect the UsageData folder. By default it will select download only updates and hot fixes for versions that are newer than the current version of the site you are updating .

Text

Description automatically generated

When you click “Download Updates”, I command window will come up that asks you to wait. Once it is done downloading it will disappear.  
Graphical user interface, text

Description automatically generated

You can also follow the .\ServiceConnectionTool\ServiceConnectionTool.log log file.

###### Send Updates to CM

The last step is getting it into Configuration Manager. Once it is downloading, copy the folder back to the Site System with the Service Connection Point role installed on it. Then run the Script again as Administrator.

Graphical user interface, text

Description automatically generated

After you click “Upload Updates” a Black Command screen will appear telling you it is importing data is in progress. Please wait.

Graphical user interface, text

Description automatically generated

The updates usually appear in the console in just a couple minutes, but some times it can take 30 minutes.

Graphical user interface, text, application, email

Description automatically generated

|  |
| --- |
| MECMOfflineUpdateGUI.ps1 |
| # -------------------------------------------------------------------------------------------------  # <copyright file="MECMOfflineUpdateGUI.ps1" Company="Justin Mnatsakanyan-Barbalace,Sr CE, Microsoft, 2022">  # All rights reserved.  # </copyright>  #  # Description: This script prepares the host machine for various Azure Migrate Scenarios.  # Version: 1.0.3  # Requirements: Run as Administrator  #  <#  LEGAL DISCLAIMER  This Sample Code is provided for the purpose of illustration only and is not  intended to be used in a production environment. THIS SAMPLE CODE AND ANY  RELATED INFORMATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER  EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF  MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. We grant You a  nonexclusive, royalty-free right to use and modify the Sample Code and to  reproduce and distribute the object code form of the Sample Code, provided  that You agree: (i) to not use Our name, logo, or trademarks to market Your  software product in which the Sample Code is embedded; (ii) to include a valid  copyright notice on Your software product in which the Sample Code is embedded;  and (iii) to indemnify, hold harmless, and defend Us and Our suppliers from and  against any claims or lawsuits, including attorneys’ fees, that arise or result  from the use or distribution of the Sample Code.    This script is provided "AS IS" with no warranties, and confers no rights. Use  of included script samples are subject to the terms specified  at http://www.microsoft.com/info/cpyright.htm.  #>  $ScriptFullPath=$myInvocation.InvocationName  $ScriptName=$myInvocation.MyCommand  Add-Type -Name Window -Namespace Console -MemberDefinition '  [DllImport("Kernel32.dll")]  public static extern IntPtr GetConsoleWindow();  [DllImport("user32.dll")]  public static extern bool ShowWindow(IntPtr hWnd, Int32 nCmdShow);'  [Console.Window]::ShowWindow([Console.Window]::GetConsoleWindow(), 0)  #-------------------------------------------------------------#  #----Initial Declarations-------------------------------------#  #-------------------------------------------------------------#  Add-Type -AssemblyName PresentationCore, PresentationFramework, System.Windows.Forms  $Xaml = @"  <Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" Width="700" Height="450">  <Grid>  <TabControl Margin="15,15,15,15">  <TabItem Name="ExTab" Header="Export Usage Data Cab">  <Grid Background="#ff0000">    <Label Name="ExDescription" HorizontalAlignment="Left" VerticalAlignment="Top" Content="This utility should be run on the server with the Service Connection Point role." Margin="10,4,0,0" Foreground="#ffffff" FontWeight="Bold" Width="637" Height="25" FontStyle="Italic"/>    <TextBox Name="ExSCTLocation" HorizontalAlignment="Left" VerticalAlignment="Top" Height="22" Width="314" Margin="231,30,0,0" FontWeight="Medium"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="ServiceConnectionTool.exe Location" Margin="10,30,0,0" Foreground="#ffffff" FontWeight="Medium" Width="214" Height="27"/>  <Button Name="ExSCTBrowse" Content="Browse" HorizontalAlignment="Left" VerticalAlignment="Top" Width="76" Margin="555,30,0,0" Height="23"/>  <Label Name="ExDescription1" HorizontalAlignment="Left" VerticalAlignment="Top" Content="This utility will out put several files that need to be moved to a internet connected device. Please select a folder " Margin="10,75,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="ExDescription2" HorizontalAlignment="Left" VerticalAlignment="Top" Content="to out put to And make sure the ServiceConnectionTool.exe is the current version in the CD.Latest folder. By default " Margin="10,95,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="ExDescription3" HorizontalAlignment="Left" VerticalAlignment="Top" Content="the location this script is run from is where the package is created If tis is not where you want it created select " Margin="10,115,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="ExDescription4" HorizontalAlignment="Left" VerticalAlignment="Top" Content="a new location to create the package. " Margin="10,135,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <TextBox Name="ExPackageLocation" HorizontalAlignment="Left" VerticalAlignment="Top" Height="22" Width="314" Margin="231,175,0,0" FontWeight="Medium"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="Location to save the transfer package" Margin="10,175,0,0" Foreground="#ffffff" FontWeight="Medium" Width="214" Height="27"/>  <Button Name="ExPackageBrowse" Content="Browse" HorizontalAlignment="Left" VerticalAlignment="Top" Width="76" Margin="555,175,0,0" Height="23"/>  <Button Content="Export Usage" HorizontalAlignment="Left" VerticalAlignment="Top" Width="75" Margin="545,295,0,0" Name="ExportUsageData"/>  </Grid>  </TabItem>  <TabItem Name="DLTab" Header="Download Updates">  <Grid Background="#008000">  <Label Name="DLDescription" HorizontalAlignment="Left" VerticalAlignment="Top" Content="This utility should be run on the server with the Service Connection Point role." Margin="10,4,0,0" Foreground="#ffffff" FontWeight="Bold" Width="637" Height="25" FontStyle="Italic"/>    <TextBox Name="DLSCTLocation" HorizontalAlignment="Left" VerticalAlignment="Top" Height="22" Width="314" Margin="231,30,0,0" FontWeight="Medium"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="ServiceConnectionTool.exe Location" Margin="10,30,0,0" Foreground="#ffffff" FontWeight="Medium" Width="214" Height="27"/>  <Button Name="DLSCTBrowse" Content="Browse" HorizontalAlignment="Left" VerticalAlignment="Top" Width="76" Margin="555,30,0,0" Height="23"/>  <Label Name="DLDescription1" HorizontalAlignment="Left" VerticalAlignment="Top" Content="The computer this is downloaded from needs to be able to connect to the internet." Margin="10,75,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="DLDescription2" HorizontalAlignment="Left" VerticalAlignment="Top" Content="" Margin="10,95,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="DLDescription3" HorizontalAlignment="Left" VerticalAlignment="Top" Content="" Margin="10,115,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="DLDescription4" HorizontalAlignment="Left" VerticalAlignment="Top" Content="If the location of the Usagedata.cab folder is not automatically selected, please select it. " Margin="10,135,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <TextBox Name="DLPackageLocation" HorizontalAlignment="Left" VerticalAlignment="Top" Height="22" Width="314" Margin="231,175,0,0" FontWeight="Medium"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="Root location where files will be saved" Margin="10,175,0,0" Foreground="#ffffff" FontWeight="Medium" Width="230" Height="27"/>  <Button Name="DLPackageBrowse" Content="Browse" HorizontalAlignment="Left" VerticalAlignment="Top" Width="76" Margin="555,175,0,0" Height="23"/>  <Grid HorizontalAlignment="Left" VerticalAlignment="Top" Width="600" Height="164" Margin="10,200,0,0">  <RadioButton Name="DownloadAll" GroupName="DownloadType" Foreground="#ffffff" HorizontalAlignment="Left" VerticalAlignment="Top" Content="Download everything, including updates and hotfixes." Margin="16,10,0,0" Width="370" Height="38"/>  <RadioButton Name="DownloadHotfix" GroupName="DownloadType" Foreground="#ffffff" HorizontalAlignment="Left" VerticalAlignment="Top" Content="Only download all hotfixes" Margin="16,40,0,0" Width="370" />  <RadioButton Name="DownloadSiteVersion" GroupName="DownloadType" Foreground="#ffffff" IsChecked="True" HorizontalAlignment="Left" VerticalAlignment="Top" Content="Only download updates and hotfixes that have a newer version than the version of your site" Margin="16,70,0,0" Width="600" />  </Grid>  <Button Name="DownloadUpdates" Content="Download Updates" HorizontalAlignment="Left" VerticalAlignment="Top" Width="110" Margin="530,310,0,0" />  </Grid>  </TabItem>  <TabItem Name="ULTab" Header="Send Updates To CM">  <Grid Background="#b22222">  <Label Name="ULDescription" HorizontalAlignment="Left" VerticalAlignment="Top" Content="This utility should be run on the server with the Service Connection Point role." Margin="10,4,0,0" Foreground="#ffffff" FontWeight="Bold" Width="637" Height="25" FontStyle="Italic"/>    <TextBox Name="ULSCTLocation" HorizontalAlignment="Left" VerticalAlignment="Top" Height="22" Width="314" Margin="231,30,0,0" FontWeight="Medium"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="ServiceConnectionTool.exe Location" Margin="10,30,0,0" Foreground="#ffffff" FontWeight="Medium" Width="214" Height="27"/>  <Button Name="ULSCTBrowse" Content="Browse" HorizontalAlignment="Left" VerticalAlignment="Top" Width="76" Margin="555,30,0,0" Height="23"/>  <Label Name="ULDescription1" HorizontalAlignment="Left" VerticalAlignment="Top" Content="The UpdatePack needs to be run on the computer with the Service Connection Point MECM role installed on it." Margin="10,75,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="ULDescription2" HorizontalAlignment="Left" VerticalAlignment="Top" Content="" Margin="10,95,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="ULDescription3" HorizontalAlignment="Left" VerticalAlignment="Top" Content="" Margin="10,115,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <Label Name="ULDescription4" HorizontalAlignment="Left" VerticalAlignment="Top" Content="If the location of the UpdatePack folder is not automatically selected, please select it. " Margin="10,135,0,0" Foreground="#ffffff" FontWeight="Medium" Width="637" Height="24"/>  <TextBox Name="ULPackageLocation" HorizontalAlignment="Left" VerticalAlignment="Top" Height="22" Width="314" Margin="231,175,0,0" FontWeight="Medium"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="Location of the UpdatePacks folder" Margin="10,175,0,0" Foreground="#ffffff" FontWeight="Medium" Width="214" Height="27"/>  <Button Name="ULPackageBrowse" Content="Browse" HorizontalAlignment="Left" VerticalAlignment="Top" Width="76" Margin="555,175,0,0" Height="23"/>  <Button Name="UploadUpdates" Content="Upload Updates" HorizontalAlignment="Left" VerticalAlignment="Top" Width="110" Margin="530,310,0,0" />  </Grid>  </TabItem>  </TabControl>  </Grid></Window>  "@  #-------------------------------------------------------------#  #----Control Event Handlers-----------------------------------#  #-------------------------------------------------------------#  #region Logic  #Functions  #get-childitem $PSScriptRoot -Recurse | where{$\_.Name -eq "ServiceConnectionTool.exe"}  Function Read-FolderBrowserDialog([string]$Message, [string]$InitialDirectory, [switch]$NoNewFolderButton)  {  $AssemblyFullName = 'System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089'  $Assembly = [System.Reflection.Assembly]::Load($AssemblyFullName)  $OpenFileDialog = [System.Windows.Forms.OpenFileDialog]::new()  $OpenFileDialog.AddExtension = $false  $OpenFileDialog.CheckFileExists = $false  $OpenFileDialog.DereferenceLinks = $true  $OpenFileDialog.Filter = "Folders|`n"  $OpenFileDialog.Multiselect = $false  if($Message){$OpenFileDialog.Title =$Message}else{$OpenFileDialog.Title = "Select folder"}  $OpenFileDialog.InitialDirectory=$InitialDirectory  $OpenFileDialogType = $OpenFileDialog.GetType()  $FileDialogInterfaceType = $Assembly.GetType('System.Windows.Forms.FileDialogNative+IFileDialog')  $IFileDialog = $OpenFileDialogType.GetMethod('CreateVistaDialog',@('NonPublic','Public','Static','Instance')).Invoke($OpenFileDialog,$null)  $OpenFileDialogType.GetMethod('OnBeforeVistaDialog',@('NonPublic','Public','Static','Instance')).Invoke($OpenFileDialog,$IFileDialog)  [uint32]$PickFoldersOption = $Assembly.GetType('System.Windows.Forms.FileDialogNative+FOS').GetField('FOS\_PICKFOLDERS').GetValue($null)  $FolderOptions = $OpenFileDialogType.GetMethod('get\_Options',@('NonPublic','Public','Static','Instance')).Invoke($OpenFileDialog,$null) -bor $PickFoldersOption  $FileDialogInterfaceType.GetMethod('SetOptions',@('NonPublic','Public','Static','Instance')).Invoke($IFileDialog,$FolderOptions)  $VistaDialogEvent = [System.Activator]::CreateInstance($AssemblyFullName,'System.Windows.Forms.FileDialog+VistaDialogEvents',$false,0,$null,$OpenFileDialog,$null,$null).Unwrap()  [uint32]$AdviceCookie = 0  $AdvisoryParameters = @($VistaDialogEvent,$AdviceCookie)  $AdviseResult = $FileDialogInterfaceType.GetMethod('Advise',@('NonPublic','Public','Static','Instance')).Invoke($IFileDialog,$AdvisoryParameters)  $AdviceCookie = $AdvisoryParameters[1]  $Result = $FileDialogInterfaceType.GetMethod('Show',@('NonPublic','Public','Static','Instance')).Invoke($IFileDialog,[System.IntPtr]::Zero)  $FileDialogInterfaceType.GetMethod('Unadvise',@('NonPublic','Public','Static','Instance')).Invoke($IFileDialog,$AdviceCookie)  if ($Result -eq [System.Windows.Forms.DialogResult]::OK) {  $FileDialogInterfaceType.GetMethod('GetResult',@('NonPublic','Public','Static','Instance')).Invoke($IFileDialog,$null)    }  [string]$output=$OpenFileDialog.FileName  write-host $output.trim()  if ($output.trim() -ne ""){ return $output.trim()}  }  #endregion  #-------------------------------------------------------------#  #----Script Execution-----------------------------------------#  #-------------------------------------------------------------#  $Window = [Windows.Markup.XamlReader]::Parse($Xaml)  [xml]$xml = $Xaml  $xml.SelectNodes("//\*[@Name]") | ForEach-Object { Set-Variable -Name $\_.Name -Value $Window.FindName($\_.Name) }  ################  #get SCT  ################  Try{  $SCTPath=Get-ItemPropertyValue -Path HKLM:\SOFTWARE\Microsoft\CCM\_OfflineUpdateTool -Name SCTPath -ErrorAction SilentlyContinue -Verbose  }catch{  $SCTPath=$null    }  if($SCTPath -ne $null){  $ExSCTLocation.Text=$SCTPath.trim()  }elseif(test-path $env:SMS\_LOG\_PATH\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool\ServiceConnectionTool.exe){  $ExSCTLocation.Text="$($env:SMS\_LOG\_PATH)\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool"  }  if(Test-Path "$PSScriptRoot\ServiceConnectionTool\ServiceConnectionTool.exe"){  $ULSCTLocation.Text="$PSScriptRoot\ServiceConnectionTool"  }  $ExTab.IsFocused  ##################################  #Export Usage Data (Ex)  ##################################  <#  if(test-path $env:SMS\_LOG\_PATH\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool\ServiceConnectionTool.exe){  $ExSCTLocation.Text="$($env:SMS\_LOG\_PATH)\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool"  $ExSCTLocation.Text=$ExSCTLocation.Text.trim()  $DLSCTLocation.Text=$ExSCTLocation.Text.trim()  $ULSCTLocation.Text=$ExSCTLocation.Text.trim()  }elseif(test-path $PSScriptRoot\ServiceConnectionTool\ServiceConnectionTool.exe){  $ExSCTLocation.Text="$($PSScriptRoot)\ServiceConnectionTool"  $ExSCTLocation.Text=$ExSCTLocation.Text.trim()  $DLSCTLocation.Text=$ExSCTLocation.Text.trim()  $ULSCTLocation.Text=$ExSCTLocation.Text.trim()  }else{    do{  [System.Windows.MessageBox]::Show('I need the location of the Service Connection tool to continue. It is usually located at "\\<SiteServer>\SMS\_<SiteCode>\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool". Please browse to that location.')  $ExSCTLocation.Text=Read-FolderBrowserDialog -Message "I cannot locate the folder ServiceConnectionTool folder. Please browse to it."  $ExSCTLocation.Text=$ExSCTLocation.Text.trim()  $DLSCTLocation.Text=$ExSCTLocation.Text.trim()  $ULSCTLocation.Text=$ExSCTLocation.Text.trim()  test-path "$($ExSCTLocation.Text.trim())\ServiceConnectionTool.exe"    }While(!(test-path "$($ExSCTLocation.Text.trim())\ServiceConnectionTool.exe"))  }  #>  if(Test-Path "$PSScriptRoot\ServiceConnectionTool\ServiceConnectionTool.exe"){  $DLSCTLocation.Text="$PSScriptRoot\ServiceConnectionTool"  }  $ExTab.Add\_GotFocus({  if(test-path $env:SMS\_LOG\_PATH\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool\ServiceConnectionTool.exe){  $ExSCTLocation.Text="$($env:SMS\_LOG\_PATH)\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool"  $ExSCTLocation.Text=$ExSCTLocation.Text.trim()  $ULSCTLocation.Text=$ExSCTLocation.Text.trim()    }elseif($SCTPath -ne $null){  $ExSCTLocation.Text=$SCTPath.trim()  $ULSCTLocation.Text=$ExSCTLocation.Text.trim()  }elseif(test-path $PSScriptRoot\ServiceConnectionTool\ServiceConnectionTool.exe){  $ExSCTLocation.Text="$($PSScriptRoot)\ServiceConnectionTool"  $ExSCTLocation.Text=$ExSCTLocation.Text.trim()  $ULSCTLocation.Text=$ExSCTLocation.Text.trim()  }else{    do{  if(!(test-path "$($ExSCTLocation.Text.trim())\ServiceConnectionTool.exe")){  [System.Windows.MessageBox]::Show('I need the location of the Service Connection tool to continue. It is usually located at "\\<SiteServer>\SMS\_<SiteCode>\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool". Please browse to that location. The Service Connection tool needs to be the version that came with the site being updated.')  $ExSCTLocation.Text=Read-FolderBrowserDialog -Message "I cannot locate the folder ServiceConnectionTool folder. Please browse to it."  $ExSCTLocation.Text=$ExSCTLocation.Text.trim()  }      }While(!(test-path "$($ExSCTLocation.Text.trim())\ServiceConnectionTool.exe"))  }  })  $ExPackageLocation.Text=$PSScriptRoot  $ExSCTBrowse.Add\_Click({  [string]$Value=$null  $Value=Read-FolderBrowserDialog -Message "Please select a directory" -initialDirectory $ExSCTLocation.Text.trim()  if($value.trim() -ne "" -and $value -ne $null){$ExSCTLocation.Text=$value.trim()}  })  $ExPackageBrowse.Add\_Click({  [string]$Value=$null  $Value=Read-FolderBrowserDialog -Message "Please select a directory" -initialDirectory $ExPackageLocation.Text.trim()  if($value.trim() -ne "" -and $value -ne $null){  $ExPackageLocation.Text=$value.trim()  Write-Host $ExPackageLocation.Text.trim()  }  })  $ExportUsageData.Add\_Click({    if($SCPath -eq $Null){  New-Item -Path HKLM:\SOFTWARE\Microsoft\CCM\_OfflineUpdateTool -Verbose -Force  New-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\CCM\_OfflineUpdateTool -Name SCTPath -PropertyType String -Value $ExSCTLocation.Text.trim() -Verbose -Force  $SCTPath=$ExSCTLocation.Text.trim()  }elseif($SCTPath.trim() -ne $ExSCTLocation.Text.trim()){  Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\CCM\_OfflineUpdateTool -Name SCTPath -PropertyType String -Value $ExSCTLocation.Text.trim() -Verbose -Force  }    $DateSerial=Get-Date -Format "yyyy.MM.dd.HH.mm.ss"  $reg=$null  $reg=Get-ChildItem -Path Registry::HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\SMS\Components\SMS\_SERVICE\_CONNECTOR -ErrorAction SilentlyContinue -Verbose  if($reg.count -lt 1){[System.Windows.MessageBox]::Show('Export Usage Data needs to be run on the server with the MECM Service Connection Point role installed')}else{  [string]$Value=$null  if("$($ExPackageLocation.Text.trim())\ServiceConnectionTool" -ne "$($ExSCTLocation.Text.trim())"){  if(Test-Path "$($ExPackageLocation.Text.trim())\ServiceConnectionTool"){Rename-Item -Path "$($ExPackageLocation.Text.trim())\ServiceConnectionTool" -NewName "$($ExPackageLocation.Text.trim())\ServiceConnectionTool.$DateSerial"}  Copy-Item -Path $ExSCTLocation.Text.trim() -Destination "$($ExPackageLocation.Text.trim())\ServiceConnectionTool" -Recurse -Verbose  Copy-Item -Path $ScriptFullPath -Destination "$($ExPackageLocation.Text.trim())\$ScriptName" -Force -ErrorAction SilentlyContinue -Verbose  }  if(Test-Path "$($ExPackageLocation.Text.trim())\UsageData"){Rename-Item -Path "$($ExPackageLocation.Text.trim())\UsageData" -NewName "$($ExPackageLocation.Text.trim())\UsageData.$DateSerial"}  New-Item "$($ExPackageLocation.Text.trim())\UsageData" -ItemType Directory -Force -Verbose  Start-Process "$($ExPackageLocation.Text.trim())\ServiceConnectionTool\ServiceConnectionTool.exe" -ArgumentList "-prepare -usagedatadest ""$($ExPackageLocation.Text.trim())\UsageData\$($env:ComputerName)-UsageData.cab""" -Wait -Verbose  Start-Process "$($ExPackageLocation.Text.trim())\ServiceConnectionTool\ServiceConnectionTool.exe" -ArgumentList "-export -dest ""$($ExPackageLocation.Text.trim())\UsageData\$($env:ComputerName)-UsageDate.csv""" -Wait -Verbose  }  })  ##################################  #Download Updates (DL)  ##################################  $DLTab.Add\_GotFocus({  if(test-path $PSScriptRoot\ServiceConnectionTool\ServiceConnectionTool.exe){  $DLSCTLocation.Text="$($PSScriptRoot)\ServiceConnectionTool"  $DLSCTLocation.Text=$DLSCTLocation.Text.trim()  }else{    do{  if(!(test-path "$($DLSCTLocation.Text.trim())\ServiceConnectionTool.exe")){  [System.Windows.MessageBox]::Show('I need the location of the Service Connection tool to continue. It is usually located at "\\<SiteServer>\SMS\_<SiteCode>\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool". Please browse to that location. The Service Connection tool needs to be the version that came with the site being updated.')  $DLSCTLocation.Text=Read-FolderBrowserDialog -Message "I cannot locate the folder ServiceConnectionTool folder. Please browse to it."  $DLSCTLocation.Text=$DLSCTLocation.Text.trim()  }      }While(!(test-path "$($DLSCTLocation.Text.trim())\ServiceConnectionTool.exe"))  }  })  $DLPackageLocation.Text="$PSScriptRoot"  $DLPackageBrowse.Add\_Click({  [string]$Value=$null  $Value=Read-FolderBrowserDialog -Message "Please select the location of the proper version of the Service Connection Tool." -initialDirectory $DLPackageLocation.Text.trim()  if($value.trim() -ne "" -and $value -ne $null){  $DLPackageLocation.Text=$value.trim()  Write-Host $DLPackageLocation.Text.trim()  }  })  $DownloadUpdates.Add\_Click({  if($DownloadAll.IsChecked){$DownloadType="-downloadall"}  elseif($DownloadHotfix.IsChecked){$DownloadType="-downloadhotfix"}  elseif($DownloadSiteVersion.IsChecked){$DownloadType="-downloadsiteversion"}  $DateSerial=Get-Date -Format "yyyy.MM.dd.HH.mm.ss"  if(Test-Path "$($DLPackageLocation.Text.trim())\UpdatePacks"){Rename-Item -Path "$($DLPackageLocation.Text.trim())\UpdatePacks" -NewName "$($DLPackageLocation.Text.trim())\UpdatePacks.$DateSerial"}  New-Item "$($DLPackageLocation.Text.trim())\UpdatePacks" -ItemType Directory -Force -Verbose  Start-Process "$($DLSCTLocation.Text.trim())\ServiceConnectionTool.exe" -ArgumentList "-connect $DownloadType -usagedatasrc ""$($DLPackageLocation.Text.trim())\UsageData"" -updatepackdest ""$($DLPackageLocation.Text.trim())\UpdatePacks""" -Wait -Verbose    })  ##################################  #Upload Updates (UL)  ##################################  $ULPackageLocation.Text="$PSScriptRoot\UpdatePacks"  $ULTab.Add\_GotFocus({  if(test-path "$($PSScriptRoot)\ServiceConnectionTool\ServiceConnectionTool.exe"){    $ULSCTLocation.Text="$($PSScriptRoot)\ServiceConnectionTool"  }elseif(test-path $env:SMS\_LOG\_PATH\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool\ServiceConnectionTool.exe){  $ULSCTLocation.Text="$($env:SMS\_LOG\_PATH)\..\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool"    }elseif($SCTPath -ne $null -and (Test-Path "$SCTPath\ServiceConnectionTool.exe" -Verbose)){  $ULSCTLocation.Text=$SCTPath.trim()  }else{  do{  if(!(test-path "$($ULSCTLocation.Text.trim())\ServiceConnectionTool.exe")){  [System.Windows.MessageBox]::Show('I need the location of the Service Connection tool to continue. It is usually located at "\\<SiteServer>\SMS\_<SiteCode>\cd.latest\SMSSETUP\TOOLS\ServiceConnectionTool". Please browse to that location. The Service Connection tool needs to be the version that came with the site being updated.')  $ULSCTLocation.Text=Read-FolderBrowserDialog -Message "I cannot locate the folder ServiceConnectionTool folder. Please browse to it."  $ULSCTLocation.Text=$ULSCTLocation.Text.trim()  }    }While(!(test-path "$($ULSCTLocation.Text.trim())\ServiceConnectionTool.exe"))  }    if(Test-Path $($ULPackageLocation.Text.trim())){  $ULPackageLocation.Text="$PSScriptRoot\UpdatePacks"  }else{  do{  if(!(test-path "$($ULPackageLocation.Text.trim())")){  [System.Windows.MessageBox]::Show('I need the location of the UpdatePacks to continue. It is usually located at "scriptfolder\UpdatePacks". Please browse to that location. The Service Connection tool needs to be the version that came with the site being updated.')  $ULPackageLocation.Text=Read-FolderBrowserDialog -Message "I cannot locate the folder UpdatePacks folder. Please browse to it." -InitialDirectory $PSScriptRoot  $ULPackageLocation.Text=$ULPackageLocation.Text.trim()  }    }While(!(test-path "$($ULPackageLocation.Text.trim())"))  }  })  $ULPackageBrowse.Add\_Click({  [string]$Value=$null  $Value=Read-FolderBrowserDialog -Message "Please select the location of the proper version of the Service Connection Tool." -initialDirectory $ULPackageLocation.Text.trim()  if($value.trim() -ne "" -and $value -ne $null){  $ULPackageLocation.Text=$value.trim()  Write-Host $ULPackageLocation.Text.trim()  }  })  $UploadUpdates.Add\_Click({  Start-Process "$($ULSCTLocation.Text.trim())\ServiceConnectionTool.exe" -ArgumentList "-import -updatepacksrc ""$($ULPackageLocation.Text.trim())""" -Wait -Verbose    })  ######################################  $Global:SyncHash = [HashTable]::Synchronized(@{})  $SyncHash.Window = $Window  $Jobs = [System.Collections.ArrayList]::Synchronized([System.Collections.ArrayList]::new())  $initialSessionState = [initialsessionstate]::CreateDefault()  Function Start-RunspaceTask  {  [CmdletBinding()]  Param([Parameter(Mandatory=$True,Position=0)][ScriptBlock]$ScriptBlock,  [Parameter(Mandatory=$True,Position=1)][PSObject[]]$ProxyVars)    $Runspace = [RunspaceFactory]::CreateRunspace($InitialSessionState)  $Runspace.ApartmentState = 'STA'  $Runspace.ThreadOptions = 'ReuseThread'  $Runspace.Open()  ForEach($Var in $ProxyVars){$Runspace.SessionStateProxy.SetVariable($Var.Name, $Var.Variable)}  $Thread = [PowerShell]::Create('NewRunspace')  $Thread.AddScript($ScriptBlock) | Out-Null  $Thread.Runspace = $Runspace  [Void]$Jobs.Add([PSObject]@{ PowerShell = $Thread ; Runspace = $Thread.BeginInvoke() })  }  $JobCleanupScript = {  Do  {  ForEach($Job in $Jobs)  {  If($Job.Runspace.IsCompleted)  {  [Void]$Job.Powershell.EndInvoke($Job.Runspace)  $Job.PowerShell.Runspace.Close()  $Job.PowerShell.Runspace.Dispose()  $Job.Powershell.Dispose()    $Jobs.Remove($Job)  }  }  Start-Sleep -Seconds 1  }  While ($SyncHash.CleanupJobs)  }  Get-ChildItem Function: | Where-Object {$\_.name -notlike "\*:\*"} | select name -ExpandProperty name |  ForEach-Object {  $Definition = Get-Content "function:$\_" -ErrorAction Stop  $SessionStateFunction = New-Object System.Management.Automation.Runspaces.SessionStateFunctionEntry -ArgumentList "$\_", $Definition  $InitialSessionState.Commands.Add($SessionStateFunction)  }  $Window.Add\_Closed({  Write-Verbose 'Halt runspace cleanup job processing'  $SyncHash.CleanupJobs = $False  })  $SyncHash.CleanupJobs = $True  function Async($scriptBlock){ Start-RunspaceTask $scriptBlock @([PSObject]@{ Name='DataContext' ; Variable=$DataContext},[PSObject]@{Name="State"; Variable=$State},[PSObject]@{Name = "SyncHash";Variable = $SyncHash})}  Start-RunspaceTask $JobCleanupScript @([PSObject]@{ Name='Jobs' ; Variable=$Jobs })  $Window.ShowDialog() |