### 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.

### CM Update Reset Tool Gui

The problem with the Update Reset Tool is there is no Gui. To address that I have made a simple PowerShell Gui for it. Technically, This should be able to run from any computer as long as it has the MECM console installed on it and it can gain remote access to the registry of the Site Server, however the fail-safe place to run it is on the Site Server.

Graphical user interface, text, email

Description automatically generated

When you run this script, it attempts to detect all required settings, but if it can’t, you will need to manually enter them. If it does detect them, make sure they are the correct settings, especially if you have more than 1 Configuration Manager environment or you have an environment with a Central Administration Site (CAS).

If you are using a Named instance, you will need to make sure the Instance Name is in the SQL Instance field, but if you are using the default instance, you can leave it blank.

|  |
| --- |
| CMUpdateResetToolGui.ps1 |
| #-------------------------------------------------------------#  #----Initial Declarations-------------------------------------#  #-------------------------------------------------------------#  Add-Type -AssemblyName PresentationCore, PresentationFramework  $Xaml = @"  <Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" Width="800" Height="400" Background="#00ced1">  <Grid>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="Site Code" Margin="16.5,13,0,0"/>  <TextBox HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120" TextWrapping="Wrap" Margin="85,12,0,0" Name="SiteCodeTxt"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="SMS Provider Server" Margin="240.5,13,0,0"/>  <TextBox HorizontalAlignment="Left" VerticalAlignment="Top" Height="30" Width="282" TextWrapping="Wrap" Margin="374,12,0,0" Name="SMSProviderServerName"/>  <DataGrid HorizontalAlignment="Left" IsReadOnly="True" VerticalAlignment="Top" Background="#ffffff" Width="712" Height="122" Margin="44,137,0,0" Name="UpdateListData">  <DataGrid.Columns>  </DataGrid.Columns>  </DataGrid>  <Button Content="Refresh Updates" HorizontalAlignment="Left" VerticalAlignment="Top" Width="120" Margin="76,295,0,0" Name="RefreshBtn" Height="19"/>  <Button Content="Reset Selected" HorizontalAlignment="Left" VerticalAlignment="Top" Width="120" Margin="266,295,0,0" Name="Remove"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="SQL Server" Margin="10,54,0,0"/>  <TextBox HorizontalAlignment="Left" VerticalAlignment="Top" Height="28" Width="210" TextWrapping="Wrap" Margin="82,53,0,0" Name="SqlServerTxt"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="Sql Instance" Margin="295,54,0,0"/>  <TextBox HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120" TextWrapping="Wrap" Margin="370,53,0,0" Name="SqlInstanceTxt"/>  <Label HorizontalAlignment="Left" VerticalAlignment="Top" Content="SQL Database" Margin="502,54,0,0"/>  <TextBox HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120" TextWrapping="Wrap" Margin="600,53,0,0" Name="SQLDBTxt"/>  </Grid>  </Window>  "@  $Window = [Windows.Markup.XamlReader]::Parse($Xaml)  [xml]$xml = $Xaml  $xml.SelectNodes("//\*[@Name]") | ForEach-Object { Set-Variable -Name $\_.Name -Value $Window.FindName($\_.Name) }  ##################  # Events  ####################  #$LastUsedMP=(Get-ItemProperty HKLM:\SOFTWARE\Microsoft\CCM\LocationServices ).EventLastUsedMP  $Provider=$null  $Instance=$null  $SiteServer=(Get-ItemProperty HKLM:\SOFTWARE\WOW6432Node\Microsoft\ConfigMgr10\AdminUI\Connection).server  $Reg = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey('LocalMachine', $SiteServer)  $RegKey= $Reg.OpenSubKey("SOFTWARE\\Microsoft\\SMS\\Setup")  $Provider = $RegKey.GetValue("Provider Location")  if($Provider -ne $null){  $SMSProviderServerName.Text=$Provider  }  $RegKey= $Reg.OpenSubKey("SOFTWARE\\Microsoft\\SMS\\SQL Server\\Site System SQL Account")  $SQLServer=$RegKey.GetValue("Server")  $SqlServerTxt.Text=$SQLServer  $SQLDB=$RegKey.GetValue("Database Name")  $SQLDBTxt.Text=$SQLDB  $Instance=$RegKey.GetValue("Instanace")  $SqlInstanceTxt.Text=$instance  $RegKey= $Reg.OpenSubKey("SOFTWARE\\Microsoft\\SMS\\DP")  $SiteCodeTxt.Text=$RegKey.GetValue("SiteCode")  $StateCodes=(262145,262146,327679,65537,65538,131071,131073,131074,131075,196607)  $RefreshBtn.Add\_Click({  Fresh  })  $Remove.Add\_Click({  if($SqlInstanceTxt.Text -ne $null -and $SqlInstanceTxt.Text -ne ""){  $instance = "-i $($SqlInstanceTxt.Text) "  }  $SiteServer=(get-CMSite).ServerName  $Roles=Get-CMSiteRole -SiteCode $($SiteCodeTxt.Text)  #$DBServer=($Roles | Where-Object {$\_.RoleName -eq "SMS SQL Server"} | Where-Object {$\_.RoleName -eq "SMS SQL Server"} ).NetworkOSPath.replace("\","")  foreach($SelectedUpdate in $UpdateListData.SelectedItems){  $PackageGuid=$SelectedUpdate.PackageGuid  Start-Process "\\$($SiteServer)\SMS\_$($SiteCodeTxt.Text)\cd.latest\SMSSETUP\TOOLS\CMUpdateReset\CMUpdateReset.exe" -ArgumentList "-S $($SqlServerTxt.Text) -D $($SQLDBTxt.Text) $instance -P $PackageGuid"  }  Fresh  })  Function Fresh{  Write-Host $SiteCodeTxt.Text  # Site configuration  $SiteCode = $SiteCodeTxt.Text # Site code  #$ProviderMachineName = $SMSProviderServerName.Text # SMS Provider machine name  # Customizations  $initParams = @{}  #$initParams.Add("Verbose", $true) # Uncomment this line to enable verbose logging  #$initParams.Add("ErrorAction", "Stop") # Uncomment this line to stop the script on any errors  # Do not change anything below this line  # Import the ConfigurationManager.psd1 module  if($Provider -eq $null -and $SMSProviderServerName.Text -ne $null){  $Provider=$SMSProviderServerName.Text  if($Provider -like "\*,\*"){$Provider=$Provider.Split(",")}  }  if((Get-Module ConfigurationManager) -eq $null) {  if(Test-Path "$($ENV:SMS\_ADMIN\_UI\_PATH)\..\ConfigurationManager.psd1"){  Import-Module "$($ENV:SMS\_ADMIN\_UI\_PATH)\..\ConfigurationManager.psd1" @initParams  }else{  [System.Windows.MessageBox]::Show('Iam unable to locate the "Configuration Manager" PowerShell Module. The MECM console may not be installed but is required.')  $Console=$false  }  }  # Connect to the site's drive if it is not already present  if((Get-PSDrive -Name $SiteCode -PSProvider CMSite -ErrorAction SilentlyContinue) -eq $null) {  if($Provider.count -gt 1){  try{  New-PSDrive -Name $SiteCode -PSProvider CMSite -Root $Provider[0] @initParams  }catch{  New-PSDrive -Name $SiteCode -PSProvider CMSite -Root $Provider[1] @initParams  }  }else{  New-PSDrive -Name $SiteCode -PSProvider CMSite -Root $Provider @initParams  }  }  # Set the current location to be the site code.  Set-Location "$($SiteCode):\" @initParams  Write-Host $($SiteCode)  $CMUpdates=Get-CMSiteUpdate -Fast | Select Name, PackageGuid, State, MoreInfoLink | Where-Object {$\_.State -in $StateCodes}  $UpdateSet=New-Object System.Collections.ArrayList  Foreach($Update in $CMUpdates){  $Row = "" | Select Name, PackageGuid, State, MoreInfoLink  $Row.name=$Update.Name  $Row.PackageGuid=$Update.PackageGuid  switch ($Update.State){  262145 {$Row.State="DOWNLOAD\_IN\_PROGRESS"}  262146 {$Row.State="DOWNLOAD\_SUCCESS"}  327679 {$Row.State="CONTENT\_REPLICATING"}  65538 {$Row.State="CONTENT\_REPLICATION\_SUCCESS"}  131071 {$Row.State="CONTENT\_REPLICATION\_FAILED"}  131073 {$Row.State="PREREQ\_IN\_PROGRESS"}  131074 {$Row.State="PREREQ\_WARNING"}  196607 {$Row.State="PREREQ\_ERROR"}  }    $Row.MoreInfoLink=$Update.MoreInfoLink  $UpdateSet.add($Row)  }  $UpdateListData.ItemsSource=$UpdateSet  }  Fresh  ###########################  $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()  <#  State Codes we care about:  DOWNLOAD\_IN\_PROGRESS = 262145  DOWNLOAD\_SUCCESS = 262146  DOWNLOAD\_FAILED = 327679  CONTENT\_REPLICATING = 65537  CONTENT\_REPLICATION\_SUCCESS = 65538  CONTENT\_REPLICATION\_FAILED = 131071  PREREQ\_IN\_PROGRESS = 131073  PREREQ\_SUCCESS = 131074  PREREQ\_WARNING = 131075  PREREQ\_ERROR = 196607  Full List of Update State Codes  DOWNLOAD\_IN\_PROGRESS = 262145  DOWNLOAD\_SUCCESS = 262146  DOWNLOAD\_FAILED = 327679  APPLICABILITY\_CHECKING = 327681  APPLICABILITY\_SUCCESS = 327682  APPLICABILITY\_HIDE = 393213  APPLICABILITY\_NA = 393214  APPLICABILITY\_FAILED = 393215  CONTENT\_REPLICATING = 65537  CONTENT\_REPLICATION\_SUCCESS = 65538  CONTENT\_REPLICATION\_FAILED = 131071  PREREQ\_IN\_PROGRESS = 131073  PREREQ\_SUCCESS = 131074  PREREQ\_WARNING = 131075  PREREQ\_ERROR = 196607  INSTALL\_IN\_PROGRESS = 196609  INSTALL\_WAITING\_SERVICE\_WINDOW = 196610  INSTALL\_WAITING\_PARENT = 196611  INSTALL\_SUCCESS = 196612  INSTALL\_PENDING\_REBOOT = 196613  INSTALL\_FAILED = 262143  INSTALL\_CMU\_VALIDATING = 196614  INSTALL\_CMU\_STOPPED = 196615  INSTALL\_CMU\_INSTALLFILES = 196616  INSTALL\_CMU\_STARTED = 196617  INSTALL\_CMU\_SUCCESS = 196618  INSTALL\_WAITING\_CMU = 196619  INSTALL\_CMU\_FAILED = 262142  INSTALL\_INSTALLFILES = 196620  INSTALL\_UPGRADESITECTRLIMAGE = 196621  INSTALL\_CONFIGURESERVICEBROKER = 196622  INSTALL\_INSTALLSYSTEM = 196623  INSTALL\_CONSOLE = 196624  INSTALL\_INSTALLBASESERVICES = 196625  INSTALL\_UPDATE\_SITES = 196626  INSTALL\_SSB\_ACTIVATION\_ON = 196627  INSTALL\_UPGRADEDATABASE = 196628  INSTALL\_UPDATEADMINCONSOLE = 196629  #> |

Code Snippet 9 - CMUpdateResetToolGui.ps1