## Dot source the required App Deploy Toolkit Functions

[string]$moduleAppDeployToolkitMain = ".\AppDeployToolkitMain.ps1"

If (-not (Test-Path -LiteralPath $moduleAppDeployToolkitMain -PathType 'Leaf')) { Throw "Module does not exist at the specified location [$moduleAppDeployToolkitMain]." }

. $moduleAppDeployToolkitMain

## Create a Windows Installer object

[\_\_comobject]$Installer = New-Object -ComObject 'WindowsInstaller.Installer' -ErrorAction 'Stop'

## Set DisplayLevel to no UI (silent)

Set-ObjectProperty -InputObject $Installer -PropertyName 'UILevel' -Value 2

## Open the MSI database in read only mode

[string]$MsiPath = "C:\Temp\putty-64bit-0.78-installer.msi"

[int32]$msiOpenDatabaseModeReadOnly = 0

Write-Log -Message "Open the MSI database [$MsiPath] in read only mode." -Source ${CmdletName}

[\_\_comobject]$MsiPathDatabase = Invoke-ObjectMethod -InputObject $Installer -MethodName 'OpenDatabase' -ArgumentList @($MsiPath, $msiOpenDatabaseModeReadOnly)

## Apply tranform on database

[string]$ApplyTransformPath = "C:\Temp\putty-64bit-0.78-installer.mst"

[int32]$msiSuppressApplyTransformErrors = 63

$null = Invoke-ObjectMethod -InputObject $MsiPathDatabase -MethodName 'ApplyTransform' -ArgumentList @($ApplyTransformPath, $msiSuppressApplyTransformErrors)

## Start Windows Installer Session

[int32]$msiOpenPackageOption = 0

[\_\_comobject]$InstallerSession = Invoke-ObjectMethod -InputObject $Installer -MethodName 'OpenPackage' -ArgumentList @($MsiPathDatabase, $msiOpenPackageOption)

## Do required standard actions for FileCosting

$null = Invoke-ObjectMethod -InputObject $InstallerSession -MethodName 'DoAction' -ArgumentList @('CostInitialize', 0)

$null = Invoke-ObjectMethod -InputObject $InstallerSession -MethodName 'DoAction' -ArgumentList @('FileCost', 0)

$null = Invoke-ObjectMethod -InputObject $InstallerSession -MethodName 'DoAction' -ArgumentList @('CostFinalize', 0)

$null = Invoke-ObjectMethod -InputObject $InstallerSession -MethodName 'DoAction' -ArgumentList @('InstallValidate', 0)

## Get VerifyDiskSpace property

# Returns true if enough disk space exists, and false if the disk is full.

[bool]$FreeDiskSpace = Get-ObjectProperty -InputObject $InstallerSession -PropertyName 'VerifyDiskSpace'

## Release Session, Database and Installer Object

$null = [Runtime.Interopservices.Marshal]::ReleaseComObject($InstallerSession)

$null = [Runtime.Interopservices.Marshal]::ReleaseComObject($MsiPathDatabase)

$null = [Runtime.Interopservices.Marshal]::ReleaseComObject($Installer)