

Microsoft.NET.Native.Framework.2.2_2.2.27912.0_x64__8wekyb3d8bbwe.appx
 Microsoft.NET.Native.Runtime.2.2_2.2.28604.0_x64__8wekyb3d8bbwe.appx
 Microsoft.Services.Store.Engagement_10.0.19011.0_x64__8wekyb3d8bbwe.appx
 Microsoft.UI.Xaml.2.3_2.32002.13001.0_x64__8wekyb3d8bbwe.appx
 Il est possible d'écouter les logs nécessaires via [Oxide](#) [lib](#) [8wekyb3d8bbwe](#) [Microsoft Store](#).

Dans cet exemple j'ai fait un package pour installer Microsoft Company Portal. Voir mes autres scripts PowerShell, le premier pour installer et le second pour désinstaller.

Voici un schéma de l'arborescence du dossier avec un sous dossier Dependencies

```
# Install Microsoft.WhiteBoard Appx
# Information :
#           Name           : Microsoft.Whiteboard
#           Publisher       : CN=Microsoft Corporation, O=Microsoft
#                           Corporation, L=Redmond, S=Washington, C=US
#           Architecture    : X64
#           ResourceId      :
#           Version         : 20.10420.5102.0
#           PackageFullName :
Microsoft.Whiteboard_20.10420.5102.0_x64__8wekyb3d8bbwe
#           InstallLocation : C:\Program
Files\WindowsApps\Microsoft.Whiteboard_20.10420.5102.0_x64__8wekyb3d8bbwe
#           IsFramework    : False
#           PackageFamilyName : Microsoft.Whiteboard_8wekyb3d8bbwe
#           PublisherId     : 8wekyb3d8bbwe
#           IsResourcePackage : False
#           IsBundle        : False
#           IsDevelopmentMode : False
#           NonRemovable    : False
#           Dependencies    :
{Microsoft.VCLibs.140.00_14.0.27810.0_x64__8wekyb3d8bbwe,
#
Microsoft.NET.Native.Framework.2.2_2.2.27912.0_x64__8wekyb3d8bbwe,
#
Microsoft.NET.Native.Runtime.2.2_2.2.28604.0_x64__8wekyb3d8bbwe}
#           IsPartiallyStaged : False
#           SignatureKind    : Store
#           Status           : 0k

#===== Variable for use Root folder of PS1 =====
If ($psISE)
{
    $currentScriptDirectory = Split-Path -Parent -Path
$psISE.CurrentFile.FullPath
}
If (!$currentScriptDirectory)
{
    If ($MyInvocation.MyCommand.CommandType -eq "ExternalScript")
    {
        $currentScriptDirectory = Split-Path -Parent -Path
$MyInvocation.MyCommand.Definition
    }
}
```

```

    }
    Else
    {
        $currentScriptDirectory = Split-Path -Parent -Path
([Environment]::GetCommandLineArgs()[0])
    }
}

# Function Write-Log

Function Write-Log {
    [CmdletBinding()]
    Param(
        [Parameter(Mandatory=$False)]
        [ValidateSet("INFO","WARN","ERROR","FATAL","DEBUG")]
        [String]
        $Level = "INFO",

        [Parameter(Mandatory=$True)]
        [string]
        $Message,

        [Parameter(Mandatory=$False)]
        [string]
        $logfile
    )

    $Stamp = (Get-Date).ToString("yyyy/MM/dd HH:mm:ss")
    $Line = "$Stamp $Level $Message"
    If($logfile) {
        Add-Content $logfile -Value $Line
    }
    Else {
        Write-Output $Line
    }
}

#=====
# Log Path
$Manufacturer = 'Microsoft'
$ManufacturerApplicationName = 'Whiteboard'
$Version = '20.10420.5102.0'
$logpath = 'C:\Windows\Packages\Logs\' + $Manufacturer + '_' +
$ManufacturerApplicationName + '_' + $Version + '_INSTALL.log'

If ((Test-path -Path 'C:\Windows\Packages\Logs\' -eq $false){
New-item -Path 'C:\Windows\Packages\Logs\' -ItemType Directory -ErrorAction
SilentlyContinue | Out-Null
}
Write-Log -Level INFO -Message "Install $Manufacturer
$ManufacturerApplicationName $Version" -logfile $logpath

```

```

Write-Log -Level INFO -Message 'Install Appx for All Users' -logfile
$logpath

# Install
Try{
    $DependenciesPath = Join-Path -Path $currentScriptDirectory -ChildPath
Dependencies -Resolve
    Add-AppxProvisionedPackage -Online -PackagePath $(Join-Path -Path
$currentScriptDirectory -ChildPath
"Microsoft.Whiteboard_20.10420.5102.0_x64__8wekyb3d8bbwe.appx") -SkipLicense
-LogPath $(Join-Path -Path $currentScriptDirectory -ChildPath "Temp.log") -
DependencyPackagePath $(Join-Path -Path $DependenciesPath -ChildPath
Microsoft.NET.Native.Framework.2.2_2.2.27912.0_x64__8wekyb3d8bbwe.Appx),$(Jo
in-Path -Path $DependenciesPath -ChildPath
Microsoft.NET.Native.Runtime.2.2_2.2.28604.0_x64__8wekyb3d8bbwe.Appx),$(Join
-Path -Path $DependenciesPath -ChildPath
Microsoft.VCLibs.140.00_14.0.27810.0_x64__8wekyb3d8bbwe.Appx) | Out-Null
    Get-Content $(Join-Path -Path $currentScriptDirectory -ChildPath
"Temp.log") | ForEach-Object {Add-Content $logpath -Value $_}
    Remove-Item -Path $(Join-Path -Path $currentScriptDirectory -ChildPath
"Temp.log") -Force
    Write-Log -Level INFO -Message "Appx is Provisioned" -logfile $logpath
}
Catch
{
    $ErrorMessage = $_.Exception.Message
    Write-Log -Level ERROR -Message $ErrorMessage -logfile $logpath
}

```

Désinstallation d'un APPX

```

# Uninstall Microsoft.WhiteBoard Appx
#===== Variable for use Root folder of PS1 =====
If ($psISE)
{
    $currentScriptDirectory = Split-Path -Parent -Path
$psISE.CurrentFile.FullPath
}
If (!$currentScriptDirectory)
{
    If ($MyInvocation.MyCommand.CommandType -eq "ExternalScript")
    {
        $currentScriptDirectory = Split-Path -Parent -Path
$MyInvocation.MyCommand.Definition
    }
    Else
    {
        $currentScriptDirectory = Split-Path -Parent -Path
([Environment]::GetCommandLineArgs()[0])
    }
}

```

```
}  
}  
  
# Function Write-Log  
  
Function Write-Log {  
    [CmdletBinding()]  
    Param(  
        [Parameter(Mandatory=$False)]  
        [ValidateSet("INFO","WARN","ERROR","FATAL","DEBUG")]  
        [String]  
        $Level = "INFO",  
  
        [Parameter(Mandatory=$True)]  
        [string]  
        $Message,  
  
        [Parameter(Mandatory=$False)]  
        [string]  
        $logfile  
    )  
  
    $Stamp = (Get-Date).ToString("yyyy/MM/dd HH:mm:ss")  
    $Line = "$Stamp $Level $Message"  
    If($logfile) {  
        Add-Content $logfile -Value $Line  
    }  
    Else {  
        Write-Output $Line  
    }  
}  
  
# Log Path  
$Manufacturer = 'Microsoft'  
$ManufacturerApplicationName = 'Whiteboard'  
$Version = '20.10420.5102.0'  
$logpath = 'C:\Windows\Packages\Logs\' + $Manufacturer + '_' +  
$ManufacturerApplicationName + '_' + $Version + '_UNINSTALL.log'  
  
If ((Test-path -Path 'C:\Windows\Packages\Logs\') -eq $false){  
    New-item -Path 'C:\Windows\Packages\Logs\' -ItemType Directory -ErrorAction  
    SilentlyContinue | Out-Null  
}  
Write-Log -Level INFO -Message "Uninstall $Manufacturer  
$ManufacturerApplicationName $Version" -logfile $logpath  
Get-AppxProvisionedPackage -Online | Where-Object Displayname -ceq  
"Microsoft.Whiteboard" | Remove-AppxProvisionedPackage -Online -LogPath  
$(Join-Path -Path $currentScriptDirectory -ChildPath "Temp.log") | Out-Null  
Get-Content $(Join-Path -Path $currentScriptDirectory -ChildPath "Temp.log")  
| ForEach-Object {Add-Content $logpath -Value $_}  
Remove-Item -Path $(Join-Path -Path $currentScriptDirectory -ChildPath
```

"Temp.log") -Force

```
Write-Log -Level INFO -Message "Remove Microsoft.Whiteboard already install  
on all users" -logfile $logpath  
Get-AppxPackage -AllUsers | Where-Object Name -ceq "Microsoft.Whiteboard" |  
Remove-AppxPackage -AllUsers
```

From:

<http://poste2travail.free.fr/dokuwiki/> - **Poste2Travail**

Permanent link:

<http://poste2travail.free.fr/dokuwiki/doku.php?id=application:appx>



Last update: **2020/08/10 23:07**