Registre Windows

Powershell traite les entrées de registre au même titre que les fichiers et peuvent donc être récupérés, créés et modifiés avec les mêmes commandes que les fichiers. Plutôt que de viser une lettre de lecteur et un répertoire tel que C:\, la "ruche" ou le registre sera visé avec un identifiant comme HKLM:\

Accès aux ruches du registre Windows

Par défaut, seulement HKCU (*HKEY_CURRENT_USER*) et HKLM (*HKEY_LOCAL_MACHINE*) sont accessibles. Puisque le registre est traité au même titre qu'un lecteur, il est possible d'observer ce qui est accessible dans une session ou un script powershell ainsi que de gagner accès à d'autres ruches en manipulant les PSDrives.

Get-PSDrive

Pour observer la liste des ruches de registres accessibles, il est possible d'utiliser la commande "Get-PSDrive" et de filtrer les résultats par "Provider" (type d'objet à manipuler).

```
PS C:\WINDOWS\system32> Get-PSDrive | Where-Object Provider -like *Registry*

Name           Used (GB)     Free (GB) Provider
Root                                            CurrentLocation
----           ---------     --------- --------      ----
                                                ---------------
HKCU                                  Registry      HKEY_CURRENT_USER
HKLM                                  Registry      HKEY_LOCAL_MACHINE
```

New-PSDrive

Il est possible de gagner accès aux autres ruches telles que HKCR (*HKEY_CLASSES_ROOT*) et HKU (*HKEY_USERS*) en les ajoutant comme PSDrive.

```
PS C:\WINDOWS\system32> New-PSDrive -PSProvider Registry -Name HKU -Root HKEY_USERS

Name           Used (GB)     Free (GB) Provider
Root                                            CurrentLocation
----           ---------     --------- --------      ----
                                                ---------------
HKU                                   Registry      HKEY_USERS
```

```
PS C:\WINDOWS\system32> New-PSDrive -PSProvider Registry -Name HKCR -Root HKEY_CLASSES_ROOT

Name           Used (GB)     Free (GB) Provider
Root                                             CurrentLocation
----           ---------     --------- --------      ----
                                                 ---------------
HKCR                                  Registry      HKEY_CLASSES_ROOT

PS C:\WINDOWS\system32> Get-PSDrive | Where-Object Provider -like *Registry*

Name           Used (GB)     Free (GB) Provider
Root                                             CurrentLocation
----           ---------     --------- --------      ----
                                                 ---------------
HKCR                                  Registry      HKEY_CLASSES_ROOT
HKCU                                  Registry      HKEY_CURRENT_USER
HKLM                                  Registry      HKEY_LOCAL_MACHINE
HKU                                   Registry      HKEY_USERS
```

Récupération d'information du registre

Get-ItemProperty

La commande Get-ItemProperty est utilisée pour récupérer l'information sur une valeur unique dans une clé. L'exemple suivant afficherait l'état du paramètre masquant les extensions de fichier connus dans l'explorateur Windows.

```
PS C:\Users\alexa> Get-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name "HideFileExt"

HideFileExt  : 0
PSPath       :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersio
n\Explorer
               \Advanced
PSParentPath :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersio
n\Explorer
PSChildName  : Advanced
```

```
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry
```

Sa valeur étant 0, les extensions de fichiers connues sont donc affichées dans l'explorateur de fichiers.

Elle peut être utilisée sur une clé sans mentionner de valeur pour afficher toutes les valeurs assignées à une clé. L'exemple suivant afficherait l'ensemble des applications qui démarreront à la connexion de l'utilisateur concerné.

```
PS C:\Users\alexa> Get-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Run" | Select-Object -Property *


Discord                 : "C:\Users\alexa\AppData\Local\Discord\Update.exe" --processStart
Discord.exe
Steam                   : "C:\Program Files (x86)\Steam\steam.exe" -silent
electron.app.Base Camp™  : C:\Program Files (x86)\Mountain Base Camp\Base Camp.exe --process-
start-args --hidden
com.squirrel.Teams.Teams : C:\Users\alexa\AppData\Local\Microsoft\Teams\Update.exe --
processStart "Teams.exe"
                           --process-start-args "--system-initiated"
EADM                    : "C:\Program Files\Electronic Arts\EA Desktop\EA
Desktop\EALauncher.exe" -silent
EpicGamesLauncher       : "C:\Program Files (x86)\Epic
Games\Launcher\Portal\Binaries\Win64\EpicGamesLauncher.exe"
                           -silent -launchcontext=boot
HASS.Agent              : C:\Users\alexa\AppData\Roaming\LAB02
Research\HASS.Agent\HASS.Agent.exe
PSPath                  :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVers
                           ion\Run
PSParentPath            :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVers
                           ion
PSChildName             : Run
PSDrive                 : HKCU
PSProvider              : Microsoft.PowerShell.Core\Registry
```

Get-ChildItem

La commande Get-ChildItem permet de récupérer tous les éléments contenus dans une clé de registre et peut être exécutée récursivement pour aussi afficher les sous-clés. L'exemple suivant

afficherait toutes les clés contenant les paramètres associés au logiciel 7-Zip ainsi que les valeurs associées à ces clés.

```
PS C:\Users\alexa> Get-ChildItem -Path "HKCU:\Software\7-Zip" -Recurse


    Hive: HKEY_CURRENT_USER\Software\7-Zip

Name                           Property
----                           --------
FM                             FolderShortcuts      : {}
                               FolderHistory        : {73, 0, 58, 0...}
                               PanelPath0           : I:\Documents\420-153-AT -
Conception\A2024\TP2_A24\
                               FlatViewArc0         : 0
                               PanelPath1           :
                               FlatViewArc1         : 0
                               ListMode             : 771
                               Position             : {162, 1, 0, 0...}
                               Panels               : {1, 0, 0, 0...}
                               ShowDots             : 1
                               ShowRealFileIcons    : 1
                               FullRow              : 1
                               ShowGrid             : 0
                               SingleClick          : 0
                               AlternativeSelection : 0
                               ShowSystemMenu       : 1
                               CopyHistory          : {70, 0, 58, 0...}


    Hive: HKEY_CURRENT_USER\Software\7-Zip\FM

Name                           Property
----                           --------
Columns                        RootFolder     : {1, 0, 0, 0...}
                               7-Zip.zip      : {1, 0, 0, 0...}
                               FSFolder       : {1, 0, 0, 0...}
                               7-Zip.Rar5     : {1, 0, 0, 0...}
                               7-Zip.Rar      : {1, 0, 0, 0...}
                               7-Zip.gzip     : {1, 0, 0, 0...}
                               7-Zip.Compound : {1, 0, 0, 0...}
                               7-Zip.7z       : {1, 0, 0, 0...}
```

```
                                7-Zip.tar      : {1, 0, 0, 0...}
                                7-Zip.Nsis     : {1, 0, 0, 0...}


    Hive: HKEY_CURRENT_USER\Software\7-Zip

Name                           Property
----                           --------
Options                        MenuIcons    : 1
                               CascadedMenu : 1
                               ContextMenu  : 2147487782
```

Création et définition de valeurs dans le registre

New-ItemProperty

La commande Set-ItemProperty permet de définir une valeur associée à une clé. La commande suivante activerait l'affichage avancé à l'écran de connexion d'un utilisateur (chaque étape plutôt que seulement "Veuillez patienter").

```
New-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"VerboseStatus" -Value 1
```

New-Item

La commande New-Item permet la création de nouvelles clés. Accompagné du paramètre -Force, la commande créera l'ensemble de la hiérarchie jusqu'à l'item créé mais si la clé était déjà existante, elle sera vidée. Il est donc généralement préférable de s'assurer de son existance avant de tenter sa création. Ceci peut facilement être réalisé à l'aide d'un "try catch" comme suit. Dans cet exemple, _Lab n'existe pas.

```
PS C:\Users\alexa> try {
>>     Get-Item -Path "HKCU:\_Lab" -ErrorAction Stop
>> }
>> catch {
>>     New-Item -Path "HKCU:\_Lab" -Force
>> }



    Hive: HKEY_CURRENT_USER
```

```
Name                           Property
----                           --------
_Lab
```

Dans cet exemple, _Lab existe et contient la valeur "Soleil123" nommée "Password". L'item n'est par conséquent pas remplacé.

```
PS C:\Users\alexa> try {
>>     Get-Item -Path "HKCU:\_Lab" -ErrorAction Stop
>> }
>> catch {
>>     New-Item -Path "HKCU:\_Lab" -Force
>> }



    Hive: HKEY_CURRENT_USER



Name                           Property
----                           --------
_Lab                           Password : Soleil123
```

Set-ItemProperty

La commande Set-ItemProperty permet de modifier une valeur assignée à une clé. Il est possible d'exécuter ceci conjointement à une vérification de l'existance de la clé pour s'assurer de la réussite de l'exécution de la commande comme suit. Dans cet exemple, le bouton de Windows Copilot sera retirée de la barre des tâches de l'utilisateur concerné.

```
$registryPath = "HKCU:\Software\Policies\Microsoft\Windows\WindowsCopilot" 
$registryValueName = "ShowCopilotButton"
$registryValueData = 0
try {
    Get-Item -Path $registryPath -ErrorAction Stop
}
catch {
    New-Item -Path $registryPath -Force
}
Set-ItemProperty -Path $registryPath -Name $registryValueName -Value $registryValueData
```

Revision #14
Created 2025-01-19 20:10:10 UTC by Alexandre Arsenault-Jetté

Updated 2025-02-08 17:46:38 UTC by Alexandre Arsenault-Jetté