**Documentation sur la programmation PowerShell**

**L'opérateur -f est utilisé pour le formatage de l'affichage**

**{I,A:FS}.. -f nombre**

**I** Indexe des items à afficher

**A** Alignement

Si A est un nombre positif alors l'alignement sera de n caractères vers la droite

Si A est un nombre négatif alors l'alignement sera de n caractères vers la gauche

**FS** Un paramètre optionnel de formatage qui agit sur l'item en fonction de son type

**Voici la liste des paramètres FS valides**

|  |  |
| --- | --- |
| :c | Représentation monétaire |
| :d | Padded. (:dP precision=number of digits);  if needed, leading zeros are added to the beginning of the (whole) number. |
| :e | Scientific (exp) notation |
| :f | Fixed point  :f5 = fix to 5 places |
| :g | Most compact format, fixed or sci  :g5 = 5 significant digits |
| :n | Number (:nP precision=number of decimal places),  includes culture separator for thousands 1,000.00 |
| :p | Pourcentage |
| :r | Précision réversible |
| :x | Représentation en format hexadécimal |
| :hh  :mm  :ss | Affiche l'heure, les minutes et les secondes d'une date  "{0:hh}:{0:mm}:{0:ss}" |
| :HH | Affiche l'heure sous le format 24H |
| MM | Affiche le mois |
| MMMM | Affiche le nom au complet du mois |
| :dd | Affiche le jour du mois |
| :ddd | Affiche le nom du jour de la semaine |
| :dddd | Affiche le nom au complet du jour de la semaine |
| :yyyy | Affiche l'année au complet |
| # | Caractère spécial |

**Voici plusieurs exemples**

**"{0:c}" -f 5.25** -> **5,25 $**

**"{0:d5}" -f 123** -> **00123**

**"{0,10:d5}" -f 123** ->  **00123**

**# Pour créer une liste de nom avec un suffixe (ORDI01 à ORDI10)**

**1..10 | ForEach-Object {"ORDI{0:d2}" -f $PSItem}**

**ORDI01**

**ORDI02**

**………**

**ORDI09**

**ORDI10**

**"{0,15:n4}" -f 2123.24597** ->  **2 123,2460**

**"{0:x}" -f 255** -> **ff**

**# Convertir le caractère 'A' en valeur hexadécimale**

**'0x' + "{0:x}" -f [int][char]'A' -**->**0x41**

**"{0:###-###-###}" -f 123456789** -> **123-456-789**

**# Alignement de texte à gauche et à droite**

**"{0,-10}|{1,10}" -f "Power","Shell"** -> **Power | Shell**

**# Affichage de texte et de nombre**

**"{1,10:d5}{0,10}{2,10:x}" -f "Salut",64,255** -> **00064 Salut ff**

**# Affiche l'heure, les minutes et les secondes de la date**

**# note: n'indique pas AM ou PM**

**"{0:hh}:{0:mm}:{0:ss}" -f (Get-Date)** -> **02:44:49**

**# Affiche l'heure (format 24H), les minutes et les secondes de la date**

**"{0:HH}:{0:mm}:{0:ss}" -f (Get-Date)** -> **14:45:57**

**# Affiche le jour du mois de la date**

**"{0:dd}" -f (Get-Date)** -> **11**

**# Affiche le nom du jour de la semaine de la date**

**"{0:ddd}" -f (Get-Date)** -> **jeu.**

**# Affiche le nom au complet du jour de la semaine de la date**

**"{0:dddd}" -f (Get-Date)** -> **Jeudi**

**# Affiche l'année de la date**

**"{0:yyyy}" -f (Get-Date)** -> **2021**

**# Affiche le mois de la date**

**"{0:MM}" -f (Get-Date)** -> **02**

**# Affiche le nom au complet du mois de la date**

**PS D:\\_OUTILS> "{0:MMMM}" -f (Get-Date)** -> **février**

**Utilisation de la classe "System.Math" du ".NET Framework"**

Voici plusieurs méthodes de la classe **system.math** du ".NET Framework".

[system.math]::Abs(n)

[system.math]::Equals(m,n)

[system.math]::Exp(n) # retourne la valeur de la constante "e" à la puissance "n"

[system.math]::Ceiling(n)

[system.math]::Floor(n)

[system.math]::Max(m,n)

[system.math]::Min(m,n)

[system.math]::Pow(m,n) # retourne la valeur du nombre "m" à la puissance "n"

[system.math]::Round(n)

[system.math]::Sqrt(n)

[system.math]::Truncate(n)

Voici deux propriétés de la classe **system.math** du ".NET Framework".

[system.math]::E # constante E = 2.71828182845905

[system.math]::PI # constante PI = 3.14159265358979

**Utilisation de la classe "System.Environment" du ".NET Framework"**

# NOTE: c'est la commande la plus rapide pour récupérer le nom d'un ordinateur

[system.environment]::MachineName

[system.environment]::UserDomainName

[system.environment]::UserName

[system.environment]::OSVersion

**Utilisation de la classe "System.Windows.MessageBox" du ".NET Framework"**

Voici comment afficher un message dans un **MessageBox** en utilisant la méthode **Show**.

NOTE: La méthode Show est surchargée.

**Voici la syntaxe d’une surcharge de la méthode Show**

**Show (string messageBoxText,**

**string caption)**

**Exemple**

**[System.Windows.MessageBox]::Show('Le message.','Le titre')**

**Voici la syntaxe d’une surcharge de la méthode Show**

**Show (string messageBoxText,**

**string caption,**

**System.Windows.MessageBoxButton button,**

**System.Windows.MessageBoxImage icon)**

**Exemple**

**$rep = [System.Windows.MessageBox]::Show('Le message','Le titre','YesNoCancel','Error')**

**switch ($rep)**

**{**

**'Yes' { Write-Host 'Oui' }**

**'No' { Write-Host 'Non' }**

**'Cancel' { Write-Host 'Annuler' }**

**}**

**Voici la liste des valeurs possibles pour l'objet "System.Windows.MessageBoxButton"**

**[System.Enum]::GetNames([System.Windows.MessageBoxButton])**

**OK**

**OKCancel**

**YesNoCancel**

**YesNo**

**Voici la liste des valeurs possibles pour l'objet "System.Windows.MessageBoxImage"**

**[System.Enum]::GetNames([System.Windows.MessageBoxImage])**

**None**

**Hand**

**Error**

**Stop**

**Question**

**Exclamation**

**Warning**

**Asterisk**

**Information**

**Utilisation de la classe "Microsoft.VisualBasic.Interaction" du ".NET Framework"**

Voici comment afficher un message dans un **MsgBox** qui est toujours au premier plan.

Add-Type -AssemblyName Microsoft.VisualBasic

$rep = [Microsoft.VisualBasic.Interaction]::MsgBox('Message','YesNoCancel,SystemModal,Information','Titre')

$rep

switch ($rep)

{

'Yes' { Write-Host 'Oui' }

'No' { Write-Host 'Non' }

'Cancel' { Write-Host 'Annuler' }

}

**"Microsoft.VisualBasic.MsgBoxStyle" a plusieurs valeurs qu'il est possible de combiner par une virgule**

[System.Enum]::GetNames([Microsoft.VisualBasic.MsgBoxStyle])

ApplicationModal

DefaultButton1

OkOnly

OkCancel

AbortRetryIgnore

**YesNoCancel**

YesNo

RetryCancel

Critical

Question

Exclamation

**Information**

DefaultButton2

DefaultButton3

**SystemModal**

MsgBoxHelp

MsgBoxSetForeground

MsgBoxRight

MsgBoxRtlReading

**Utilisation de la classe "System.Convert" du ".NET Framework"**

**ToInt32(String, Int32) Convertit la représentation d’une chaîne de caractères d’un nombre dans une base spécifiée en un entier 32 bits signé.**

[system.convert]::ToInt32('10000000',2) # Conversion d’un nombre binaire en entier

[system.convert]::ToInt32('10',8) # Conversion d’un nombre octal en entier

[system.convert]::ToInt32('FF',16) # Conversion d’un nombre hexadécimal en entier

**ToInt64(String, Int32) Convertit la représentation d’une chaîne de caractères d’un nombre dans une base spécifiée en un entier 64 bits signé.**

[system.convert]::ToInt64('FFFFFFFF',16) # Conversion d’un grand nombre hexadécimal en entier

$a = [system.convert]::ToString(1024,2) # Conversion d’un nombre entier en binaire (chaîne de caractères)

$b = $a.PadLeft(32,'0') # Affichage du nombre binaire sur 32 colonnes (32 bits)

$b

$a = [system.convert]::ToString(255,16) # Conversion d’un nombre entier en hexadécimal (chaîne de caractères)

$b = $a.PadLeft(8,'0') # Affichage du nombre hexadécimal sur 8 colonnes (32 bits)

$b.ToUpper() # Affichage en majuscule du nombre hexadécimal

**Chargement d’une classe avant l'appel d'une méthode**

Add-Type -AssemblyName System.Web

[System.Web.Security.Membership]::GeneratePassword(8,1)

* le premier paramètre de GeneratePassword spécifie la longueur du mot de passe
* le deuxième paramètre de GeneratePassword spécifie le nombre minimum de caractères spéciaux

**Get-Member permet d'obtenir la liste des propriétés et des méthodes des objects.**

# $S contient la liste des services sur un ordinateur local

**$S = Get-Service**

# Get-Member obtient le type **System.ServiceProcess.ServiceController** et la liste des membres

# contient des méthodes comme **Pause, Start, Stop** et des propriétés comme **StartType, Status**.

**$S | Get-Member**

# Get-Member obtient le type **System.Object[]** et la liste des membres contient des méthodes

# comme **Add, Remove** et des propriétés comme **Length**.

**Get-Member -InputObject $S**

**Utilisation d'objet COM**

Voici un exemple de code qui utilise l'objet COM **Wscript.Shell** pour créer un raccourci sur le Bureau.

**$cible = "C:\\_TEMP\info.txt"**

**$lien = "C:\Users\richard\Desktop\Mon\_Lien.lnk"**

**$WshShell = New-Object -ComObject Wscript.Shell**

**$raccourci = $WshShell.CreateShortcut($lien)**

**$raccourci.TargetPath = $cible**

**$raccourci.Save()**

Le fichier **COM\_Object\_Excel.ps1** qui est sur LÉA montre comment utiliser l'objet COM **Excel.Application** pour insérer des valeurs et effectuer des calculs dans Excel.

Le fichier **GUI\_map\_drive.ps1** qui est sur LÉA montre comment utiliser la classe **System.Windows.Forms** pour créer des objets: Form, Button, Label, TextBox, MaskedTextBox.

**PSDrive**

Un PSDrive a un comportement similaire à l'Explorateur de fichiers.

Un PSDrive est relié à un fournisseur.

# Pour s'assurer que le module ActiveDirectory est accessible par un fournisseur PSDrive

Import-Module ActiveDirectory

# Affiche la liste des fournisseurs

Get-PSProvider | Format-Table -AutoSize

Name Capabilities Drives

---- ------------ ------

Registry ShouldProcess, Transactions {HKLM, HKCU}

Alias ShouldProcess {Alias}

Environment ShouldProcess {Env}

FileSystem Filter, ShouldProcess, Credentials {C, R}

Function ShouldProcess {Function}

Variable ShouldProcess {Variable}

Certificate ShouldProcess {Cert}

WSMan Credentials {WSMan}

ActiveDirectory Include, Exclude, Filter, ShouldProcess, Credentials {AD}

# Affiche les lecteurs disponibles pour PSDrive

Get-PSDrive | Format-Table -AutoSize

Name Used (GB) Free (GB) Provider Root CurrentLocation

---- --------- --------- -------- ---- ---------------

AD ActiveDirectory //RootDSE/

Alias Alias

C 15,69 110,76 FileSystem C:\ Users\Administrateur.HV1

Cert Certificate \

Env Environment

Function Function

HKCU Registry HKEY\_CURRENT\_USER

HKLM Registry HKEY\_LOCAL\_MACHINE

Variable Variable

WSMan WSMan

"**AD**" est le nom du lecteur qui permet d'accéder au fournisseur "**ActiveDirectory**".

**Déplacement au niveau du lecteur "AD"**

cd AD:

# Pour afficher les objets qui sont directement sous le lecteur "AD".

Get-ChildItem

Name ObjectClass DistinguishedName

---- ----------- -----------------

formation domainDNS DC=formation,DC=local

Configuration configuration CN=Configuration,DC=formation,DC=local

Schema dMD CN=Schema,CN=Configuration,DC=formation,DC=local

DomainDnsZones domainDNS DC=DomainDnsZones,DC=formation,DC=local

ForestDnsZones domainDNS DC=ForestDnsZones,DC=formation,DC=local

**Naviguer dans le lecteur "AD"**

# Déplacement au niveau du domaine FORMATION.LOCAL

cd "DC=formation,DC=local"

# Pour afficher les objets qui sont directement sous le lecteur "AD:\DC=formation,DC=local".

Get-ChildItem

Name ObjectClass DistinguishedName

---- ----------- -----------------

\_DATA organizationalUnit OU=\_DATA,DC=formation,DC=local

Builtin builtinDomain CN=Builtin,DC=formation,DC=local

Computers container CN=Computers,DC=formation,DC=local

Domain Controllers organizationalUnit OU=Domain Controllers,DC=formation,DC=local

ForeignSecurityPr... container CN=ForeignSecurityPrincipals,DC=formation,DC=local

Infrastructure infrastructureUpdate CN=Infrastructure,DC=formation,DC=local

Keys container CN=Keys,DC=formation,DC=local

LostAndFound lostAndFound CN=LostAndFound,DC=formation,DC=local

Managed Service A... container CN=Managed Service Accounts,DC=formation,DC=local

NTDS Quotas msDS-QuotaContainer CN=NTDS Quotas,DC=formation,DC=local

ORDINATEURS organizationalUnit OU=ORDINATEURS,DC=formation,DC=local

Program Data container CN=Program Data,DC=formation,DC=local

System container CN=System,DC=formation,DC=local

TPM Devices msTPM-Information... CN=TPM Devices,DC=formation,DC=local

**Users container CN=Users,DC=formation,DC=local**

UTILISATEURS organizationalUnit OU=UTILISATEURS,DC=formation,DC=local

# Déplacement au niveau du conteneur "Users"

cd "CN=Users"

# Pour sortir du lecteur PSDrive, il suffit de se déplacer dans le lecteur "C"

**c:**

**ADSI Edit**

adsiedit.msc ouvre la console ADSI.

Les deux fournisseurs les plus utilisés pour l'administration système et réseau sont

WINNT:// # c'est le fournisseur Windows et Windows Server

LDAP:// # c'est le fournisseur qui permet d'accéder à LDAP

# LDAP (Lightweight Directory Access Protocol)

**Syntaxe générique**

WINNT://<nom\_domaine>,<nom\_objet>,<nom\_classe>

# L'utilisateur "Richard" est un utilisateur local

$a = [ADSI]"WinNT://127.0.0.1/RICHARD,user"

# L'utilisateur "Administrateur" est un utilisateur du domaine "FORMATION"

$b = [ADSI]"WinNT://formation/Administrateur,user"

# Liste les propriétés de l'objet $b

$b | Get-Member

LDAP://<nom\_unique>

Le DistinguishedName sera le nom\_unique

LDAP://CN=Administrateur,CN=Users,DC=formation,DC=local

NOTE: L'objet recherché doit être parfaitement connu.

$c = [ADSI]"LDAP://CN=Administrateur,CN=Users,DC=formation,DC=local"

# Liste les propriétés de l'objet $c

$c | Get-Member

La liste des propriétés de "**$b | Get-Member**" est différente de "**$c | Get-Member**".

**Conflits entre le module Hyper-V et le module VMware.VimAutomation.Core**

La compagnie VMware utilise PowerShell pour gérer les machines virtuelles et son hyperviseur ESXi.

Malheureusement, dans le module VMware.VimAutomation.Core il y a plusieurs cmdlet qui portent le même nom que les cmdlet du module Hyper-V.

Pour utiliser le bon cmdlet, il faut utiliser la syntaxe suivante: **[ModuleName]\[Cmdlet\_or\_Function\_Name]**

Par exemple, **Get-VM** existe dans le module Hyper-V et le module VMware.VimAutomation.Core

**# Pour vérifier si une commande existe dans plusieurs modules,**

**# ajouter le caractère \* devant le nom de la commande.**

**(Get-Command \*Get-VM).Source**

**VMware.VimAutomation.Core**

**Hyper-V**

**# Si le cartactère \* n’est pas devant le nom de la commande,**

**# le résulat affiche seulement un module.**

**(Get-Command Get-VM).Source**

**VMware.VimAutomation.Core**

**Il est important de spécifier le nom du module pour utiliser le bon cmdlet.**

**Get-Help VMware.VimAutomation.Core\Get-VM**

**Get-Help Hyper-V\Get-VM**