NOTES DE COURS

Les modules

Automne 2019

Table des matières

1 Gestion des modules

Les commandes et les fonctions sont groupées par module selon la fonctionnalité. Ainsi les commandes relatives à Active Directory sont groupées dans un module **ActiveDirectory**.

1.1 Lister les commandes d'un module

Lister les commandes du module DnsClient :

```
PS C:\> Get-Command -Module DnsClient

CommandType     Name                                ModuleName
-----------     ----                                ----------
Function        Add-DnsClientNrptRule               DnsClient
Function        Clear-DnsClientCache                DnsClient
Function        Get-DnsClient                       DnsClient
Function        Get-DnsClientCache                  DnsClient
Function        Get-DnsClientGlobalSetting          DnsClient
Function        Get-DnsClientNrptGlobal             DnsClient
Function        Get-DnsClientNrptPolicy             DnsClient
Function        Get-DnsClientNrptRule               DnsClient
Function        Get-DnsClientServerAddress          DnsClient
Function        Register-DnsClient                  DnsClient
Function        Remove-DnsClientNrptRule            DnsClient
Function        Set-DnsClient                       DnsClient
Function        Set-DnsClientGlobalSetting          DnsClient
Function        Set-DnsClientNrptGlobal             DnsClient
Function        Set-DnsClientNrptRule               DnsClient
Function        Set-DnsClientServerAddress          DnsClient
Cmdlet          Resolve-DnsName                     DnsClient

PS C:\>
```

1.2 Lister les modules importés

La commande **Get-Module** permet de lister les modules de la session courante.

```
PS C:\> Get-Module

ModuleType Version    Name
---------- -------    ----
Manifest   1.0.0.0    DnsClient
Script     1.0.0.0    ISE
Manifest   3.1.0.0    Microsoft.PowerShell.Management
Manifest   3.0.0.0    Microsoft.PowerShell.Security
Manifest   3.1.0.0    Microsoft.PowerShell.Utility
Manifest   3.0.0.0    Microsoft.WSMan.Management
```

Afficher les modules disponibles :

```
PS C:\> Get-Module -ListAvailable

ModuleType Version Name
---------- ------- ----
  Manifest 1.0.0.0 AppBackgroundTask
  Manifest 2.0.0.0 AppLocker
  Manifest 2.0.0.0 Appx
    Script 1.0.0.0 AssignedAccess
  Manifest 1.0.0.0 BitLocker
  Manifest 1.0.0.0 BitsTransfer
  Manifest 1.0.0.0 BranchCache
  Manifest 1.0.0.0 CimCmdlets
  Manifest 1.0     Defender
  Manifest 1.0.0.0 DirectAccessClientComponents
    Script 3.0     Dism
  Manifest 1.0.0.0 DnsClient
  Manifest 2.0.0.0 International
  Manifest 1.0.0.0 iSCSI
    Script 1.0.0.0 ISE
  Manifest 1.0.0.0 Kds
  Manifest 3.0.0.0 Microsoft.PowerShell.Diagnostics
  Manifest 3.0.0.0 Microsoft.PowerShell.Host
  Manifest 3.1.0.0 Microsoft.PowerShell.Management
  Manifest 3.0.0.0 Microsoft.PowerShell.Security
  Manifest 3.1.0.0 Microsoft.PowerShell.Utility
  Manifest 3.0.0.0 Microsoft.WSMan.Management
  Manifest 1.0     MMAgent
  Manifest 1.0.0.0 MsDtc
  Manifest 2.0.0.0 NetAdapter
```

```
  Manifest 1.0.0.0 NetConnection
  Manifest 1.0.0.0 NetEventPacketCapture
  Manifest 2.0.0.0 NetLbfo
  Manifest 1.0.0.0 NetNat
  Manifest 2.0.0.0 NetQos
  Manifest 2.0.0.0 NetSecurity
  Manifest 1.0.0.0 NetSwitchTeam
  Manifest 1.0.0.0 NetTCPIP
  Manifest 1.0.0.0 NetworkConnectivityStatus
  Manifest 1.0.0.0 NetworkTransition
  Manifest 1.0.0.0 PcsvDevice
  Manifest 1.0.0.0 PKI
  Manifest 1.1     PrintManagement
  Manifest 1.0     PSDesiredStateConfiguration
    Script 1.0.0.0 PSDiagnostics
    Binary 1.1.0.0 PSScheduledJob
  Manifest 2.0.0.0 PSWorkflow
  Manifest 1.0.0.0 PSWorkflowUtility
  Manifest 1.0.0.0 ScheduledTasks
  Manifest 2.0.0.0 SecureBoot
  Manifest 2.0.0.0 SmbShare
  Manifest 2.0.0.0 SmbWitness
  Manifest 1.0.0.0 StartScreen
  Manifest 2.0.0.0 Storage
  Manifest 2.0.0.0 TLS
  Manifest 1.0.0.0 TroubleshootingPack
  Manifest 2.0.0.0 TrustedPlatformModule
  Manifest 2.0.0.0 VpnClient
  Manifest 1.0.0.0 Wdac
  Manifest 1.0.0.0 WindowsDeveloperLicense
    Script 1.0     WindowsErrorReporting
  Manifest 1.0.0.0 WindowsSearch

PS C:\>
```

1.3 Importer un module

A partir de la version 3, les modules sont importés automatiquement. Cependant, il est possible d'importer un module manuellement.

La commande **Import-Module** permet d'importer le module spécifié dans la session courante.

```
PS C:\> Import-Module ActiveDirectory
```

1.4 Créer un nouveau module

Afin de pouvoir les importer dynamiquement. Les nouveaux modules doivent être placés dans l'un des répertoires définis par la variable d'environnement **PSModulePath**

```
PS C:\> $env:PSModulePath
C:\Users\admin\Documents\WindowsPowerShell\Modules;
C:\Program Files\WindowsPowerShell\Modules;
C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\
```

Voici ex exemple de création d'un nouveau module convenant plusieurs fonctions:

1) Créer le répertoire qui va contenir le nouveau module :

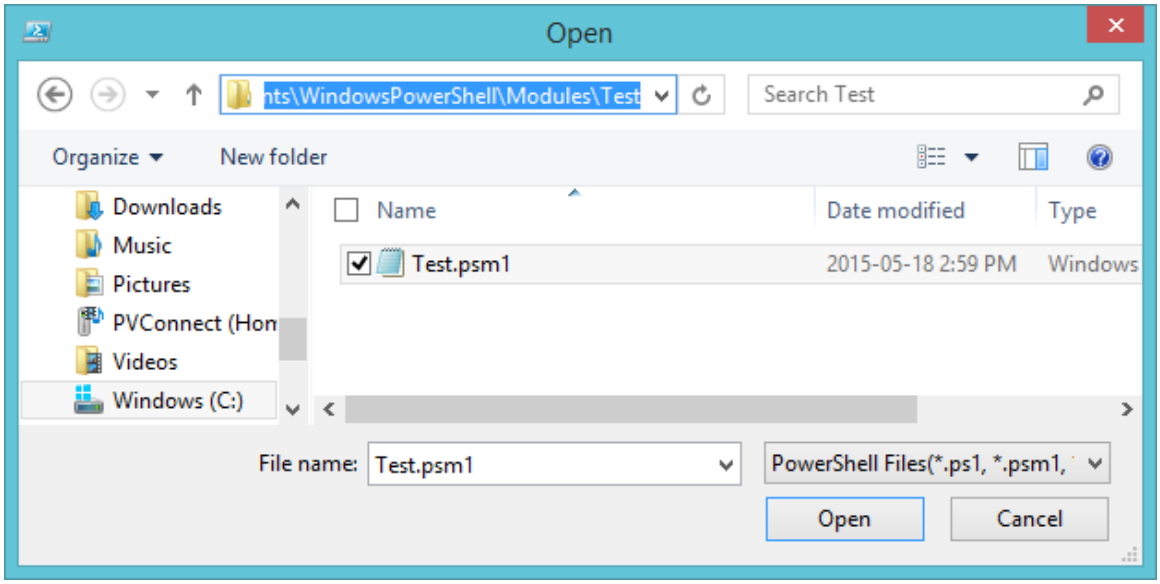
```
$rep = "$env:USERPROFILE\Documents\WindowsPowerShell\Modules"

if(!(Test-Path $rep))
{
  New-Item -Path $rep -ItemType Directory
}
```

2) Créer un répertoire nommé **Test** dans le répertoire des modules.

```
New-Item -Path $rep/Test -ItemType Directory
```

3) Créer un script PowerShell module nommé **test.psm1** dans le répertoire Test:



4) Ajouter les fonctions suivantes dans le nouveau script **test.psm1**

```
Function Get-Mois
 {
   (Get-Date).Month
 }
```

```
Function Get-Jour
 {
    $jour =  (Get-Date).DayOfWeek
    Write-Host $jour
 }
```

```
Function Get-ProchaineDate
{
  param($NombreJours)
  (Get-Date).AddDays($NombreJours)
}
```

5) Lister les fonctions du nouveau module

```
PS C:\> Get-Command -Module Test

CommandType     Name                ModuleName
-----------     ----                ----------
Function        Get-Jour            Test
Function        Get-Mois            Test
Function        Get-ProchaineDate   Test
```

```
PS C:\> Get-Module

ModuleType Version   Name    ExportedCommands
---------- -------   ----    ----------------
Script     1.0.0.0   ISE     {Get-IseSnippet,
Manifest   3.1.0.0   Microsoft.PowerShell.Management
Manifest   3.1.0.0   Microsoft.PowerShell.Utility
Script     0.0       Test    {Get-Jour, Get-Mois...
```

6) Exécuter les commandes du nouveau module

```
PS C:\> Get-Mois
7
```

```
PS C:\> Get-Jour
Tuesday
```

```
PS C:\> Get-ProchaineDate -NombreJours 5

mardi 16 juillet 2019 17:09:27
```