

## Chapitre 3

# Systeme et fonctionnalités

### 1. Introduction

Après avoir passé en revue l'ensemble des cmdlets de base et les différents lecteurs Windows PowerShell, entrons dans le vif du sujet : l'administration Windows des postes de travail.

Vous allez voir à travers ce chapitre comment accomplir des tâches administratives qui touchent directement le cœur du système Windows : activation et désactivation de fonctionnalités Windows, gestion des processus, des services, etc.

### 2. Les fonctionnalités Windows

Depuis Windows 7, il est possible d'effectuer des actions d'administration dans l'invite de commandes avec l'exécutable Dism.exe. Il est possible d'installer des mises à jour, des packs de langues, des fonctionnalités Windows, et bien plus encore. Outre sous forme d'exécutable, Dism est également présent sous la forme d'un module dans Windows PowerShell. Cela signifie que les actions possibles avec Dism.exe ont été retranscrites sous la forme de plusieurs cmdlets.

Il est possible de les consulter grâce à la commande ci-après :

```
PS C:\Windows\system32> Get-Command -Module Dism
```

CommandType	Name	Version	Source
-----	----	-----	-----
Alias	Add-AppProvisionedPackage	3.0	Dism
Alias	Add-ProvisionedAppPackage	3.0	Dism
Alias	Add-ProvisionedAppSharedPackageContainer	3.0	Dism
Alias	Add-ProvisionedAppxPackage	3.0	Dism
Alias	Apply-WindowsUnattend	3.0	Dism
Alias	Get-AppProvisionedPackage	3.0	Dism
Alias	Get-ProvisionedAppPackage	3.0	Dism
Alias	Get-ProvisionedAppSharedPackageContainer	3.0	Dism
Alias	Get-ProvisionedAppxPackage	3.0	Dism
Alias	Optimize-AppProvisionedPackages	3.0	Dism
Alias	Optimize-ProvisionedAppPackages	3.0	Dism
Alias	Optimize-ProvisionedAppxPackages	3.0	Dism
Alias	Remove-AppProvisionedPackage	3.0	Dism
Alias	Remove-ProvisionedAppPackage	3.0	Dism
Alias	Remove-ProvisionedAppSharedPackageContainer	3.0	Dism
Alias	Remove-ProvisionedAppxPackage	3.0	Dism
Alias	Set-AppPackageProvisionedDataFile	3.0	Dism
Alias	Set-ProvisionedAppPackageDataFile	3.0	Dism
Alias	Set-ProvisionedAppxDataFile	3.0	Dism
Cmdlet	Add-AppProvisionedSharedPackageContainer	3.0	Dism
Cmdlet	Add-AppxProvisionedPackage	3.0	Dism
Cmdlet	Add-WindowsCapability	3.0	Dism
Cmdlet	Add-WindowsDriver	3.0	Dism
Cmdlet	Add-WindowsImage	3.0	Dism
Cmdlet	Add-WindowsPackage	3.0	Dism
Cmdlet	Clear-WindowsCorruptMountPoint	3.0	Dism
Cmdlet	Disable-WindowsOptionalFeature	3.0	Dism
Cmdlet	Dismount-WindowsImage	3.0	Dism
Cmdlet	Enable-WindowsOptionalFeature	3.0	Dism
Cmdlet	Expand-WindowsCustomDataImage	3.0	Dism
Cmdlet	Expand-WindowsImage	3.0	Dism
Cmdlet	Export-WindowsCapabilitySource	3.0	Dism
Cmdlet	Export-WindowsDriver	3.0	Dism
Cmdlet	Export-WindowsImage	3.0	Dism
Cmdlet	Get-AppProvisionedSharedPackageContainer	3.0	Dism
Cmdlet	Get-AppxProvisionedPackage	3.0	Dism
Cmdlet	Get-NonRemovableAppsPolicy	3.0	Dism
Cmdlet	Get-WIMBootEntry	3.0	Dism
Cmdlet	Get-WindowsCapability	3.0	Dism
Cmdlet	Get-WindowsDriver	3.0	Dism
Cmdlet	Get-WindowsEdition	3.0	Dism
Cmdlet	Get-WindowsImage	3.0	Dism
Cmdlet	Get-WindowsImageContent	3.0	Dism
Cmdlet	Get-WindowsOptionalFeature	3.0	Dism
Cmdlet	Get-WindowsPackage	3.0	Dism
Cmdlet	Get-WindowsReservedStorageState	3.0	Dism
Cmdlet	Mount-WindowsImage	3.0	Dism
Cmdlet	New-WindowsCustomImage	3.0	Dism
Cmdlet	New-WindowsImage	3.0	Dism
Cmdlet	Optimize-AppxProvisionedPackages	3.0	Dism
Cmdlet	Optimize-WindowsImage	3.0	Dism
Cmdlet	Remove-AppProvisionedSharedPackageContainer	3.0	Dism
Cmdlet	Remove-AppxProvisionedPackage	3.0	Dism
Cmdlet	Remove-WindowsCapability	3.0	Dism

Cmdlet	Remove-WindowsDriver	3.0	Dism
Cmdlet	Remove-WindowsImage	3.0	Dism
Cmdlet	Remove-WindowsPackage	3.0	Dism
Cmdlet	Repair-WindowsImage	3.0	Dism
Cmdlet	Save-SoftwareInventory	3.0	Dism
Cmdlet	Save-WindowsImage	3.0	Dism
Cmdlet	Set-AppXProvisionedDataFile	3.0	Dism
Cmdlet	Set-NonRemovableAppsPolicy	3.0	Dism
Cmdlet	Set-WindowsEdition	3.0	Dism
Cmdlet	Set-WindowsProductKey	3.0	Dism
Cmdlet	Set-WindowsReservedStorageState	3.0	Dism
Cmdlet	Split-WindowsImage	3.0	Dism
Cmdlet	Start-OSUninstall	3.0	Dism
Cmdlet	Update-WIMBootEntry	3.0	Dism
Cmdlet	Use-WindowsUnattend	3.0	Dism

Nous allons étudier quelques cmdlets dans ce qui suit, à commencer par **Enable-WindowsOptionalFeature** et **Disable-WindowsOptionalFeature**. Certaines cmdlets parlent d'elles-mêmes, et nous vous invitons à les découvrir grâce à la documentation fournie par chacune d'elles.

Pour rappel, la documentation des cmdlets peut être consultée en utilisant **Get-Help**, puis en spécifiant la cmdlet concernée. En ajoutant le paramètre **-Full**, la documentation complète s'affiche, avec généralement des exemples.

```
PS C:\Windows\system32> Get-Help Enable-WindowsOptionalFeature -Full
```

Pour installer une fonctionnalité Windows directement en ligne de commande PowerShell, utilisez la cmdlet **Enable-WindowsOptionalFeature**. Mais avant d'utiliser cette commande, il faut cependant connaître le nom des fonctionnalités que vous souhaitez installer sur le poste de travail. La liste des fonctionnalités peut être consultée via la cmdlet **Get-WindowsOptionalFeature** :

```
PS C:\Windows\system32> Get-WindowsOptionalFeature -Online |
Format-Table

FeatureName                                     State
-----
Windows-Defender-Default-Definitions          Disabled
Printing-PrintToPDFServices-Features          Enabled
Printing-XPSServices-Features                Disabled
TelnetClient                                  Disabled
TFTP                                           Disabled
TIFFIFilter                                   Disabled
LegacyComponents                              Disabled
DirectPlay                                    Disabled
MSRDC-Infrastructure                          Enabled
Windows-Identity-Foundation                  Disabled
```

MicrosoftWindowsPowerShellV2Root	Enabled
MicrosoftWindowsPowerShellV2	Enabled
SimpleTCP	Disabled
NetFx4-AdvSrvs	Enabled
NetFx4Extended-ASPNET45	Disabled
WCF-Services45	Enabled
WCF-HTTP-Activation45	Disabled
WCF-TCP-Activation45	Disabled
WCF-Pipe-Activation45	Disabled
WCF-MSMQ-Activation45	Disabled
WCF-TCP-PortSharing45	Enabled
IIS-WebServerRole	Enabled
IIS-WebServer	Enabled
[...]	

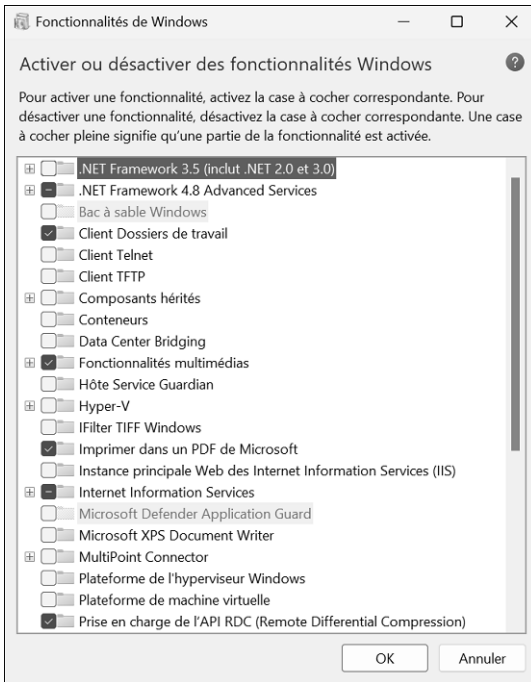
#### Remarque

Le paramètre **-Online** désigne le système Windows actuellement installé et démarré sur le poste de travail. En effet, *Dism* peut également être utilisé pour la gestion des fichiers WIM et des disques virtuels pour des scénarios de déploiement Windows.

La liste complète des fonctionnalités est présente en Annexes.

Le résultat de la commande **Get-WindowsOptionalFeature** retourne alors l'ensemble des fonctionnalités Windows, et indique l'état pour chacune d'entre elles : activée (Enabled), désactivée (Disabled) ou encore désactivée sans disposer des sources d'installation (DisableWithPayloadRemoved).

La liste des fonctionnalités peut être consultée via l'interface graphique de Windows à l'emplacement suivant : **Panneau de configuration, Programmes, Programmes et fonctionnalités, Activer ou désactiver des fonctionnalités Windows.**



Liste des fonctionnalités disponibles sur Windows 11

## 2.1 Activer une fonctionnalité

L'activation d'une fonctionnalité se fait facilement en spécifiant le nom de celle-ci avec le paramètre **-FeatureName** de la cmdlet **Enable-WindowsOptionalFeature**. Voici un aperçu des paramètres disponibles :

Paramètre	Description
<b>-FeatureName</b> <String[]>	Nom de la fonctionnalité à activer.
<b>-NoRestart</b>	Supprime la demande de redémarrage.
<b>-Online</b>	Indique que l'action est effectuée sur le système d'exploitation démarré sur le poste local.

Exemple : activation d'une fonctionnalité

La fonctionnalité IIS-WebServerRole représente l'installation du serveur web IIS.

```
PS C:\Windows\system32> Enable-WindowsOptionalFeature -Online
-FeatureName IIS-WebServerRole
```

```
Path          :
Online        : True
Restart Needed : False
```

Vous verrez, en haut de l'écran Windows PowerShell, l'avancement de l'installation de la fonctionnalité :

```
Enable-WindowsOptionalFeature: IIS-WebServerRole
Running
[ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo ]
```

Une fois l'installation terminée, si un redémarrage est nécessaire, Windows PowerShell vous le demande. Pour supprimer cette demande de redémarrage, il faut mentionner le paramètre **-NoRestart**. Un compte rendu vous indique si un redémarrage de Windows est nécessaire ou non (Restart Needed).

Il est possible d'activer plusieurs fonctionnalités en une seule ligne de commande, en mentionnant les noms des fonctionnalités séparés par une virgule.

Cas particulier : NetFx3

Pour l'installation de la fonctionnalité .NET Framework 3.5, les sources d'installation ne sont pas préchargées dans Windows. Il sera alors nécessaire de disposer d'une connexion internet pour télécharger les sources, ou alors du média d'installation de Windows et d'utiliser le paramètre **-Source** pour désigner le chemin d'accès au dossier contenant les fichiers nécessaires à l'activation de cette fonctionnalité.

```
PS C:\Windows\system32> Enable-WindowsOptionalFeature -FeatureName
NetFx3 -Source D:\sources\sxs -Online
```

Dans la ligne de commande ci-dessus, le lecteur D représente le lecteur CD ou le fichier ISO qui a préalablement été monté. Il peut également s'agir d'un lecteur disposant d'une copie des fichiers d'installation de Windows.

## 2.2 Désactiver une fonctionnalité

La désactivation d'une fonctionnalité Windows passe par la cmdlet **Disable-WindowsOptionalFeature**, avec les mêmes paramètres que précédemment, à savoir **-Online** et **-FeatureName**.

Exemple : désactivation d'une fonctionnalité

La fonctionnalité `Microsoft-Hyper-V-All` désigne le système de virtualisation Hyper-V.

```
PS C:\Windows\system32> Disable-WindowsOptionalFeature -Online
-FeatureName Microsoft-Hyper-V-All
Voulez-vous redémarrer l'ordinateur pour terminer cette opération
maintenant ?
[Y] Yes [N] No [?] Aide (la valeur par défaut est « Y ») :
```

Comme mentionné auparavant, pour passer outre les demandes de redémarrage, il faut spécifier le paramètre **-NoRestart**. Dans le cas d'un script exécuté sur un poste distant, il est important de spécifier ce paramètre, car sinon le script risque d'être bloqué jusqu'à l'attente d'une réponse.

Un message d'avertissement s'affiche si un redémarrage du poste de travail est nécessaire pour terminer l'opération.

## 3. Les applications du Microsoft Store

Avec l'arrivée de Windows 8, Microsoft a implémenté une nouvelle catégorie d'applications, plus couramment nommées apps. Mais ce nouveau type d'applications ne vient pas tout seul : Microsoft inaugure un magasin d'applications permettant alors de télécharger et d'installer de nouvelles apps. Avec Windows 10, Microsoft lance une plateforme universelle, nommée *Universal Windows Platform* (couramment nommée UWP), permettant ainsi à une seule et même app d'être exécutée aussi bien sur la version PC que mobile, ayant pourtant des architectures processeur très différentes (x86 pour le PC et ARM pour le mobile).

Incluse nativement dans Windows 10 et Windows 11, l'application Microsoft Store est le magasin d'applications permettant ainsi d'installer sur l'ordinateur de nouvelles apps. Par défaut, Windows autorise l'installation de nouvelles apps uniquement depuis le Microsoft Store. L'ensemble des applications présentes dans ce magasin sont testées, validées et signées par l'éditeur de Redmond.

Si une entreprise qui souhaite déployer une app (par exemple, une application métier qui a été développée en interne) ne peut se permettre d'attendre cette validation, il est possible d'installer des apps directement par l'intermédiaire de Windows PowerShell. Ce procédé est nommé SideLoading.

Cependant, si vous souhaitez faire du SideLoading avec Windows 10 ou Windows 11, il est nécessaire de respecter certains prérequis techniques :

- Aucune clé de licence supplémentaire n'est nécessaire, que ce soit avec une édition Professionnel ou Entreprise de Windows.
- Activer l'une des deux stratégies de groupes suivantes en fonction de votre besoin : ouvrir l'éditeur de stratégie de groupe locale (gpedit.msc) et aller dans **Configuration ordinateur - Modèles d'administration - Composants Windows \_ Déploiement de package Appx**. Activer alors la stratégie **Autorise le développement d'application du Windows Store et leur installation depuis un environnement de développement intégré** ou **Autoriser l'installation des applications approuvées**.
- Installer une app signée : l'application à installer doit être signée par une autorité de certification reconnue par les clients, ce qui signifie que le certificat ou la chaîne de certificat utilisé(e) pour signer l'app doit être présent(e) dans le magasin de certificats "Autorités de certification racines de confiance" de l'ordinateur local.