

PowerShell – `Get-Package` : lister les logiciels installés

`Get-Package` est une commande PowerShell qui permet d'afficher les logiciels installés sur un système Windows. Elle est utilisée par les techniciens pour faire un inventaire, diagnostiquer un poste, ou préparer une maintenance (mise à jour, désinstallation, migration).

À quoi sert Get-Package ?

`Get-Package` permet de :

- lister les logiciels installés
- identifier **comment** un logiciel a été installé
- savoir **quel gestionnaire** (provider) gère ce logiciel
- automatiser des audits logiciels

C'est une commande **en lecture seule** : elle n'installe ni ne supprime rien.

Commande de base

```
1 | Get-Package
```

Cette commande retourne, entre autres :

- `Name`
- `Version`
- `ProviderName`
- `Source`
- `Status`

Exemple courant

```
1 | Get-Package | Select-Object Name, Version, ProviderName
```

Comprendre `ProviderName` (élément clé)

Qu'est-ce qu'un provider ?

Un **provider** indique **quel système d'installation** est responsable du logiciel. Windows peut installer des logiciels de différentes manières, et **Get-Package** les distingue grâce à **ProviderName**.

👉 **ProviderName** répond à la question : « **Par quel mécanisme ce logiciel a-t-il été installé ?** »

Principaux ProviderName et leur signification

msi

- Logiciels installés via **Windows Installer (.msi)**
- Cas le plus classique en entreprise

Exemples :

- Microsoft Office (version MSI)
- 7-Zip
- VLC (version MSI)

👉 Ces logiciels sont généralement **désinstallables proprement**.

msu

- Mises à jour Windows (**.msu**)
- Correctifs système

Exemples :

- KB5021234
- mises à jour cumulatives

👉 Souvent **liés au système**, pas à des logiciels utilisateurs.

Programs

- Logiciels détectés via le **registre Windows**
- Souvent installés via :
 - **.exe**
 - installateurs personnalisés
 - anciens programmes

Exemples :

- Google Chrome
- Adobe Reader
- logiciels propriétaires

👉 Très fréquent, mais **moins standardisé** que MSI.

NuGet

- Paquets destinés au développement (.NET)
- Utilisés surtout par les développeurs

👉 Peu utilisé sur des postes utilisateurs classiques.

PowerShellGet

- Modules PowerShell installés via :

```
1 | Install-Module
```

👉 Ne concerne pas les logiciels Windows classiques.

Filtrer par provider (exemples utiles)

Afficher uniquement les logiciels MSI

```
1 | Get-Package | Where-Object { $_.ProviderName -eq "msi" }
```

Lister uniquement les programmes classiques

```
1 | Get-Package | Where-Object { $_.ProviderName -eq "Programs" }
```

Cas pratique technicien

Objectif

Identifier les logiciels installés manuellement par l'utilisateur.

```
1 | Get-Package |  
2 | Where-Object { $_.ProviderName -eq "Programs" } |  
3 | Select-Object Name, Version
```

Limites importantes à connaître

- **Get-Package ne liste pas toujours tout**
- Certains logiciels portables (sans installation) n'apparaissent pas
- Les résultats peuvent varier selon :

- les droits
 - la version de Windows
 - le provider installé
-

Résumé

ProviderName	Signification	Usage typique
msi	Windows Installer	Logiciels standards
msu	Windows Update	Correctifs système
Programs	Registre Windows	Logiciels classiques
NuGet	Paquets .NET	Développement
PowerShellGet	Modules PowerShell	Administration

Conclusion

`Get-Package` est un outil essentiel pour l'**inventaire logiciel** et le **diagnostic système**. La colonne `ProviderName` permet de comprendre l'**origine des installations**, ce qui est crucial pour :

- la maintenance
 - la désinstallation
 - la standardisation des postes
-