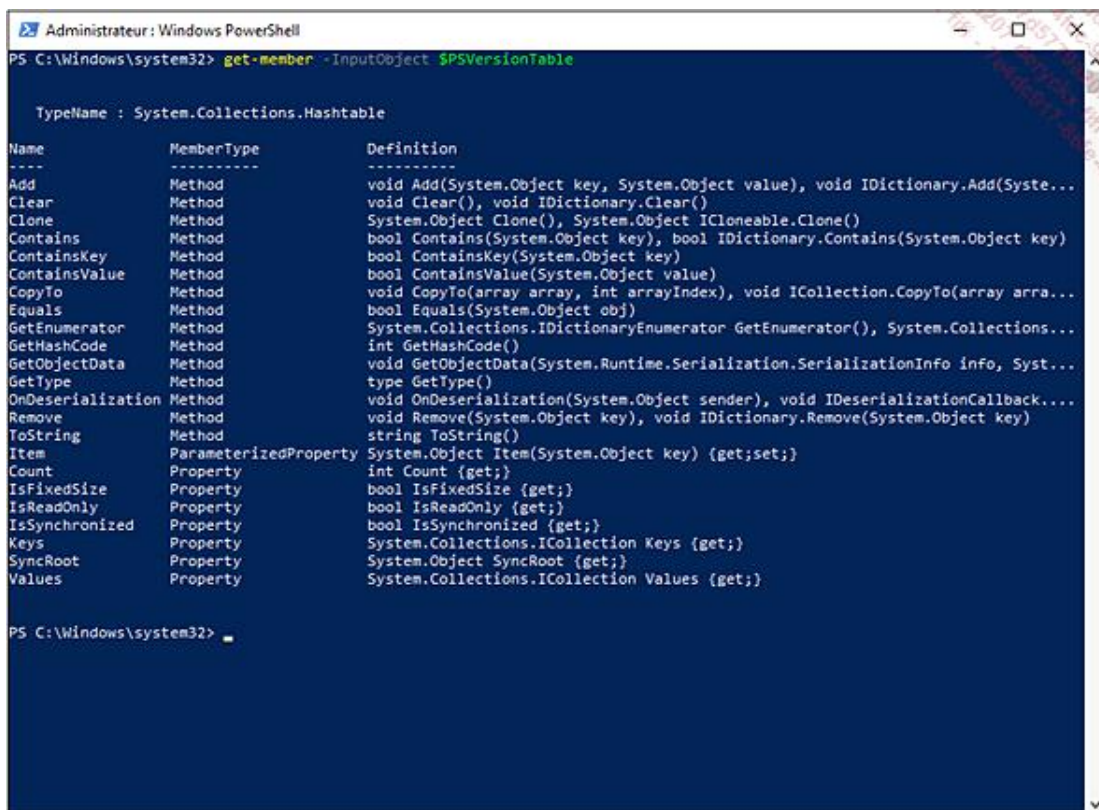


La cmdlet `get-command` permet de retourner la liste des cmdlets standards. Vous ne pourrez pas retenir l'ensemble des cmdlets disponibles pour l'environnement d'exécution PowerShell, celles-ci étant trop nombreuses. Par contre, vous devez pouvoir les retrouver et déterminer la syntaxe principale de celles-ci.

Les cmdlets importantes à connaître sont donc les suivantes :

- `get-command` : cette commande liste l'ensemble des cmdlets disponibles dans l'environnement d'exécution de PowerShell. La commande `get-command -noun process` permet de lister les cmdlets contenant la chaîne "process", cmdlets relatives à la gestion des processus.
- `get-help` : cette commande permet d'obtenir des informations détaillées sur l'utilisation des cmdlets, comme le nom et la description de la syntaxe de la cmdlet. La commande `get-help ls` permet de lister les informations détaillées sur la cmdlet qui fait référence à l'alias "ls".
- `get-member` : cette commande permet d'obtenir des informations sur les objets et listes d'objets des cmdlets. La liste des commandes suivantes permet de répertorier les objets attachés à un objet du système d'exploitation, ici le répertoire Windows, et d'afficher le contenu d'un objet de type propriété.

Exemple de l'utilisation de la commande `get-member` :



```
Administrateur: Windows PowerShell
PS C:\Windows\system32> get-member -InputObject $PSVersionTable

TypeName : System.Collections.Hashtable

Name      MemberType Definition
-----
Add        Method      void Add(System.Object key, System.Object value), void IDictionary.Add(Syste...
Clear      Method      void Clear(), void IDictionary.Clear()
Clone      Method      System.Object Clone(), System.Object ICloneable.Clone()
Contains   Method      bool Contains(System.Object key), bool IDictionary.Contains(System.Object key)
ContainsKey Method      bool ContainsKey(System.Object key)
ContainsValue Method      bool ContainsValue(System.Object value)
CopyTo     Method      void CopyTo(array array, int arrayIndex), void ICollection.CopyTo(array arra...
Equals     Method      bool Equals(System.Object obj)
GetEnumerator Method      System.Collections.IDictionaryEnumerator GetEnumerator(), System.Collections...
GetHashCode Method      int GetHashCode()
GetObjectData Method      void GetObjectData(System.Runtime.Serialization.SerializationInfo info, Syst...
GetType    Method      type GetType()
OnDeserialization Method      void OnDeserialization(System.Object sender), void IDeserializationCallback....
Remove     Method      void Remove(System.Object key), void IDictionary.Remove(System.Object key)
ToString   Method      string ToString()
Item       ParameterizedProperty System.Object Item(System.Object key) {get;set;}
Count      Property    int Count {get;}
IsFixedSize Property    bool IsFixedSize {get;}
IsReadOnly Property    bool IsReadOnly {get;}
IsSynchronized Property    bool IsSynchronized {get;}
Keys       Property    System.Collections.ICollection Keys {get;}
SyncRoot   Property    System.Object SyncRoot {get;}
Values     Property    System.Collections.ICollection Values {get;}
```

3. Windows Remote Management (WinRM)

Ce service implémente le protocole WS-Management utilisé par PowerShell pour l'administration distante des systèmes Windows dans un mode de connexion client-serveur. Le protocole WS-Management est un Web Service standard de type SOAP. Le service WinRM agit comme un listener (service d'écoute) côté serveur. L'outil de ligne de commande Winrs.exe permet de lancer des commandes distantes côté client réceptionnées par le service WinRM. PowerShell utilise le service WinRM pour l'exécution de scripts distants.

Avant de pouvoir lancer des commandes distantes et utiliser le service WinRM, vous devez configurer ce service.