

# Gérer les comptes d'utilisateurs Office 365

## Ressources

- <https://docs.microsoft.com/fr-fr/microsoft-365/enterprise/view-user-accounts-with-microsoft-365-powershell>
- <https://docs.microsoft.com/fr-fr/microsoft-365/enterprise/manage-user-accounts-and-licenses-with-microsoft-365-powershell?view=o365-worldwide>

## Se connecter au locataire Microsoft 365

```
PS > Connect-AzureAD
```

## Lister les comptes

- Lister tous les comptes

```
PS > Get-AzureADUser
```

| ObjectId                             | DisplayName                                  | UserPrincipalName   |
|--------------------------------------|--|---|
| 2d5dee3e-0665-4675-bf29-18b8a79f4651 | 31b93a88.educ-valadon-limoges.fr@fr.teams.ms | 31b93a88.educ-valadon-limoges.fr_fr.teams.ms#EXT#@educ-valadon-limoges.fr |
| 7e58156b-9145-46de-97fd-e84d1fcef12c | Killian ABADA                                | abada.killian@educ-valadon-limoges.fr                                     |
| bee1c80b-7d28-447a-a493-8024c7d7a7e1 | Chaïma ABBOU                                 | abbou.chaima@educ-valadon-limoges.fr                                      |

- Lister intégralement tous les comptes `PS > Get-AzureADUser -Top 10000`
- Lister un compte spécifique `PS /home/administrateur> Get-AzureADUser -ObjectID techer.charles@educ-valadon-limoges.fr`

```
ObjectId DisplayName UserPrincipalName UserType ----
```

```
c7f2e846-6fa0-4ca7-9cba-a23f7f829292 Charles TECHER techer.charles@educ-valadon-limoges.fr Member Member
```

- Lister toutes les propriétés d'un compte `> PS /home/administrateur> Get-AzureADUser -ObjectID techer.charles@educ-valadon-limoges.fr | select * ExtensionProperty : {[odata.metadata, https://graph.windows.net/1ae69c2d-61b2-4de4-84c7-2a4dd0de7330/$metadata#directoryObjects/@Element], [odata.type, Microsoft.DirectoryServices.User], [createdDateTime, 9/19/2019 8:28:29 AM], [employeeid, ...] DeletionTimestamp : ObjectId : c7f2e846-6fa0-4ca7-9cba-a23f7f829292 ObjectType : User AccountEnabled : True AgeGroup : AssignedLicenses : {class AssignedLicense { DisabledPlans: System.Collections.Generic.List1[System.String] SkuId: 12b8c807-2e20-48fc-b453-542b6ee9d171 } , class AssignedLicense { DisabledPlans: System.Collections.Generic.List1[System.String] Skuld: f30db892-07e9-47e9-837c-80727f46fd3d } , class AssignedLicense { DisabledPlans: System.Collections.Generic.List1[System.String] SkuId: 94763226-9b3c-4e75-a931-5c89701abe66 } , class AssignedLicense { DisabledPlans: System.Collections.Generic.List1[System.String] Skuld: a403ebcc-fae0-4ca2-8c8c-7a907fd6c235 } } AssignedPlans : {class AssignedPlan { AssignedTimestamp: 4/8/2021 12:06:30 PM CapabilityStatus: Enabled Service: CRM ServicePlanId: 40b010bb-0b69-4654-ac5e-ba161433f4b4 } , class AssignedPlan { AssignedTimestamp: 4/8/2021 12:06:30 PM CapabilityStatus: Enabled Service: ProjectProgramsAndPortfolios ServicePlanId: a55dfd10-0864-46d9-a3cd-da5991a3e0e2 } , class AssignedPlan { AssignedTimestamp: 10/18/2019 1:03:58 PM CapabilityStatus: Enabled Service: OfficeForms ServicePlanId: 9b5de886-f035-4ff2-b3d8-c9127bea3620 } , class AssignedPlan { AssignedTimestamp: 10/18/2019 1:03:58 PM CapabilityStatus: Enabled Service: SharePoint ServicePlanId: e03c7e47-402c-463c-ab25-949079bedb21 } ...} City : CompanyName : Lycée Suzanne Valadon ConsentProvidedForMinor : Country : CreationType : Department : Professeurs / Eco-Info DirSyncEnabled : True DisplayName : Charles TECHER FacsimileTelephoneNumber : GivenName : Charles IsCompromised : ImmutableId : dktyPZ6Ptk+Q7+JrtXvCLA== JobTitle : Professeurs LastDirSyncTime : 8/30/2022 9:39:22 AM LegalAgeGroupClassification : Mail : techer.charles@educ-valadon-limoges.fr MailNickName : techer.charles Mobile : OnPremisesSecurityIdentifier : S-1-5-21-3235202287-4148145215-1993724183-3292 OtherMails : {} PasswordPolicies : PasswordProfile : class PasswordProfile { Password: ForceChangePasswordNextLogin: True EnforceChangePasswordPolicy: False } PhysicalDeliveryOfficeName : Eco-Info PostalCode : PreferredLanguage : ProvisionedPlans : {class ProvisionedPlan { CapabilityStatus: Enabled ProvisioningStatus: Success Service: SharePoint } , class ProvisionedPlan { CapabilityStatus: Enabled ProvisioningStatus: Success Service: CRM } , class ProvisionedPlan { CapabilityStatus: Enabled ProvisioningStatus: Success Service: SharePoint } , class ProvisionedPlan { CapabilityStatus: Enabled ProvisioningStatus: Success Service:`

```
MicrosoftCommunicationsOnline } ...} ProvisioningErrors : {} ProxyAddresses :  
{smtp:techer.charles@educvaladonlimogesfr.onmicrosoft.com, SMTP:techer.charles@educ-valadon-limoges.fr}  
RefreshTokensValidFromDateTime : 9/19/2019 8:28:29 AM ShowInAddressList : SignInNames : {} SipProxyAddress :  
techer.charles@educ-valadon-limoges.fr State : StreetAddress : Surname : TECHER TelephoneNumber : UsageLocation : FR  
UserPrincipalName : techer.charles@educ-valadon-limoges.fr UserState : UserStateChangedOn : UserType : Member </code>
```

## Afficher l'état de synchronisation du compte

Les comptes d'utilisateur ont deux sources :

- Windows Server Active Directory (AD), qui sont des comptes qui se synchronisent entre AD local et le cloud.
- Azure Active Directory comptes (Azure AD), qui sont créés directement dans le cloud.

La commande suivante permet de rechercher les comptes qui se synchronisent à partir d'AD local en demandant à PowerShell d'obtenir la valeur True pour tous les utilisateurs dont l'attribut **DirSyncEnabled** est défini.

```
PS > Get-AzureADUser | Where {$_.DirSyncEnabled -eq $true}
```

La commande suivante permet de rechercher des comptes cloud uniquement en indiquant à PowerShell d'obtenir tous les utilisateurs dont l'attribut **DirSyncEnabled** a la valeur **False** ou non défini (**Null**) :

- Un compte qui n'a jamais été synchronisé à partir d'AD local a DirSyncEnabled défini sur Null.
- Un compte qui a été synchronisé initialement à partir d'AD local mais qui n'est plus synchronisé a la valeur False pour DirSyncEnabled.

```
PS > Get-AzureADUser | Where {$_.DirSyncEnabled -ne $true}
```

## Afficher les comptes en fonction d'une propriété commune

L'applet de commande **Where** en combinaison avec l'applet de commande **Get-AzureADUser** en utilisant le caractère « pipe » (« | »), qui indique à Azure Active Directory PowerShell de prendre les résultats d'une commande et de l'envoyer à la commande suivante permet d'afficher uniquement les comptes d'utilisateur dont l'emplacement d'utilisation n'est pas spécifié :

```
PS > Get-AzureADUser | Where {$_.UsageLocation -eq $Null}
```

La syntaxe de l'applet de commande **Where** est **Where {\$\_. [nom de la propriété du compte d'utilisateur] [opérateur de comparaison] [valeur] }**.

[opérateur de comparaison] est :

- -eq for equals,
- -ne for not equals,
- -lt for inférieur à,
- -gt for great than, and others.
- -Like avec els joker \* et ?

[value] est généralement une chaîne (une séquence de lettres, de nombres et d'autres caractères), une valeur numérique ou \$Null pour les caractères non spécifiés.

Pour plus d'informations, consultez Where :

<https://docs.microsoft.com/fr-fr/powershell/module/microsoft.powershell.core/where-object>

## Afficher les groupes en fonction d'une propriété commune

```
PS > Get-AzureADGroup -Top 1000 | Where {$_.Description -Like "*koxo*"}
```

## Lister les membres d'un groupe

```
PS > Get-AzureADGroupMember -ObjectId (Get-AzureADGroup -Top 1000 | where {$_.Description -eq "[koxoadm]
```

```
b2 sio2"}).ObjectId | select UserPrincipalName, GivenName, SurName
```

From:

/ - **Les cours du BTS SIO**

Permanent link:

</doku.php/reseau/cloud/azure/cloudshell/azureoffice365compte?rev=1662488244>

Last update: **2022/09/06 20:17**

