

# User Management mit PowerShell

Office 365

## Inhaltsverzeichnis

|           |  |          |
|-----------|--|----------|
| <b>1.</b> | <b>Benötigte Module .....</b>                          | <b>3</b> |
| <b>2.</b> | <b>Verbindung mit Office365 herstellen.....</b>        | <b>3</b> |
| 2.1       | PowerShell Scripts ausführen .....                     | 3        |
| <b>3.</b> | <b>Benutzer erfassen .....</b>                         | <b>4</b> |
| 3.1       | Ein Benutzer erfassen.....                             | 4        |
| 3.2       | Mehrere Benutzer erfassen .....                        | 5        |
| <b>4.</b> | <b>Benutzer zu Sicherheitsgruppen hinzufügen .....</b> | <b>6</b> |
| 4.1       | Ein Benutzer zur Sicherheitsgruppe hinzufügen .....    | 7        |
| 4.2       | Mehrere Benutzer zu Sicherheitsgruppen hinzufügen..... | 7        |

## 1. Benötigte Module

### Microsoft Online Services Sign-in Assistant

- [Microsoft Online Services Sign-In Assistant – 32 bit version](#)
- [Microsoft Online Services Sign-In Assistant – 64 bit version](#)

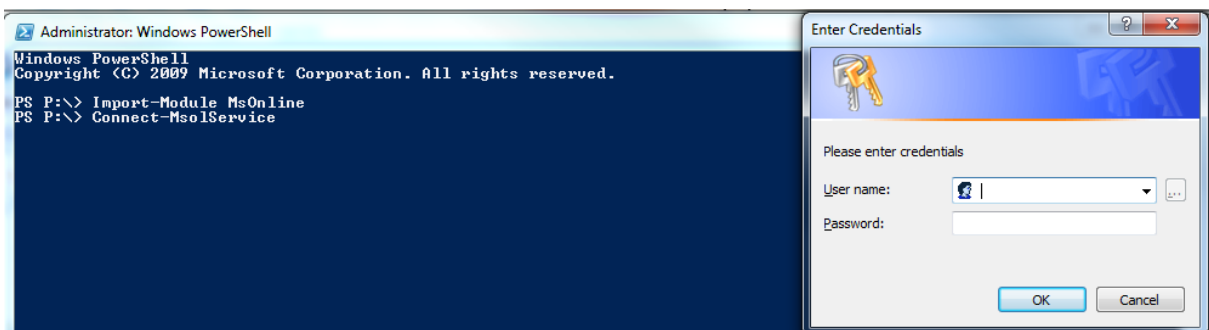
### Install Microsoft Online Services module for windows PowerShell

- [Microsoft Online Services Module for Windows PowerShell \(32-bit version\)](#)
- [Microsoft Online Services Module for Windows PowerShell \(64-bit version\)](#)

## 2. Verbindung mit Office365 herstellen

Um mit dem Onlinedienst von Office 365 zu verbinden, benötigen wir folgende Schritte:

```
„ Import-Module MsOnline  
„ Connect-MsolService
```



### 2.1 PowerShell Scripts ausführen

Damit PowerShell Scripts ausgeführt werden können, muss folgende Ausführung getätigt werden. Danach den Ordner wie angegeben aufrufen und dann das entsprechende Script ausführen.

```
„ Set-ExecutionPolicy Unrestricted  
Cd C:\temp\  
. \AddUserToGroup.ps1
```

```

Administrator: Windows PowerShell
PS P:\> Set-ExecutionPolicy Unrestricted

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic. Do you want to change the execution policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
PS P:\> cd C:\temp
PS C:\temp> .\AddUserToGroup.ps1_
  
```

## 3. Benutzer erfassen

### 3.1 Ein Benutzer erfassen

Um einen einzelnen User zu erfassen benötigen wir mehrere Schritte. Als erstes müssen wir das lizenzierte Paket wählen. Dies brauchen wir später beim hinzufügen des Benutzers.

```
„ Get-MsolAccountSku
```

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS P:\> Import-Module MsOnline
PS P:\> Connect-MsolService
PS P:\>
PS P:\>
PS P:\> Get-MsolAccountSku

AccountSkuId                                     ActiveUnits   WarningUnits   ConsumedUnits
-----
IOZAG338:ENTERPRISEPACK_B_PILOT                   25             0                5
  
```

Alle Unterlizenzen des Pakets werden mit folgendem Befehl ausgelesen:

```
„ Get-MsolAccountSku | Where-Object {$_.SkuPartNumber -eq 'ENTERPRISEPACK_B_PILOT'} |
ForEach-Object {$_.ServiceStatus}
```

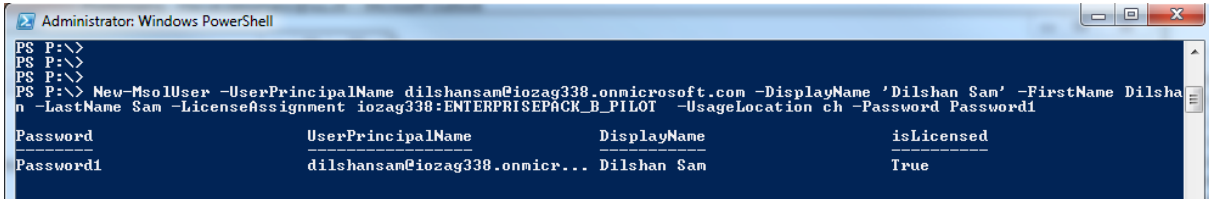
```

Administrator: Windows PowerShell
PS P:\>
PS P:\> Get-MsolAccountSku | Where-Object {$_.SkuPartNumber -eq 'ENTERPRISEPACK_B_PILOT'} | ForEach-Object {$_.ServiceStatus}

ServicePlan                                     ProvisioningStatus
-----
RMS_S_ENTERPRISE_B_PILOT                        Success
OFFICE_PRO_PLUS_SUBSCRIPTION_B_PILOT           Success
LYNC_S_ENTERPRISE_B_PILOT                      Success
SHAREPOINTWAC_B_PILOT                         Success
SHAREPOINT_S_ENTERPRISE_B_PILOT               Success
EXCHANGE_S_ENTERPRISE_B_PILOT                  Success
  
```

Jetzt kann ein einzelner User angelegt werden. Wir sehen auch gleich, dass er lizenziert wurde.

```
„ New-MsolUser -UserPrincipalName dilshansam@iozag338.onmicrosoft.com -DisplayName 'Dilshan Sam' -FirstName Dilshan -LastName Sam -LicenseAssignment
iozag338:ENTERPRISEPACK_B_PILOT -UsageLocation ch -Password Password1
```



```

Administrator: Windows PowerShell
PS P:\>
PS P:\>
PS P:\>
PS P:\> New-MsolUser -UserPrincipalName dilshansan@iozag338.onmicrosoft.com -DisplayName 'Dilshan Sam' -FirstName Dilshan -LastName Sam -LicenseAssignment iozag338:ENTERPRISEPACK_B_PILOT -UsageLocation ch -Password Password1

```

| Password  | UserPrincipalName             | DisplayName | isLicensed |
|-----------|-------------------------------|-------------|------------|
| Password1 | dilshansan@iozag338.onmicr... | Dilshan Sam | True       |

## 3.2 Mehrere Benutzer erfassen

Dass mehrere Benutzer gleichzeitig (Bulk add) erfasst werden können, benötigen wir eine Liste mit den gewünschten Usern. Es kann auf dem Office365 Admin Center unter „Bulk add“ eine leere Vorlage heruntergeladen werden. Diese ist jedoch für PowerShell nicht geeignet, da sie Abstände beinhaltet.

Deshalb gibt es eine Vorlageliste der IOZ AG, „**Template\_Userlist.txt**“



Template\_Userlist.txt

Diese Datei einfach in ein CSV-File umwandeln, bevor man es benutzt.

Die Datei soll nun mit Notepad geöffnet und mit "Save As" und der "Encoding: Unicode" Option gespeichert werden. Wir benötigen diese Option damit die Umlaute korrekt auf Office365 importiert werden. Wichtig ist, dass keine Umlaute im UserPrincipalName eingegeben werden, ansonsten wird der Import fehlschlagen.

Im Ps1 Script gibt es noch eine Option der zu deaktivierenden Lizenzen. Auf Office365 Preview funktioniert dieser leider noch nicht.

```

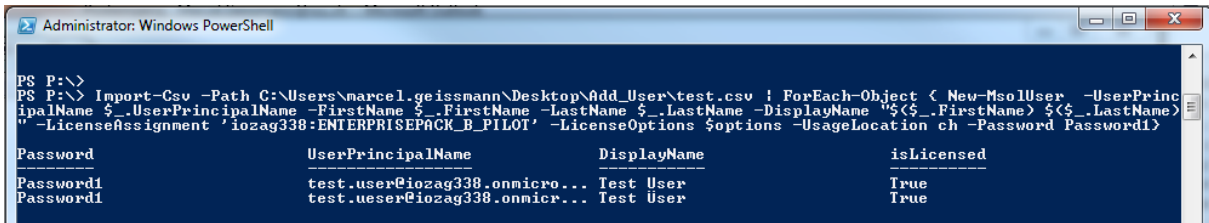
„ $options = New-MsolLicenseOptions -AccountSkuId 'iozag338:ENTERPRISEPACK_B_PILOT' -DisabledPlans RMS_S_ENTERPRISE_B_PILOT

```

```

„ Import-Csv -Path C:\Users\marcel.geissmann\Desktop\Add_User\test.csv | ForEach-Object {
New-MsolUser -UserPrincipalName $_.UserPrincipalName -FirstName $_.FirstName -LastName
$_.LastName -DisplayName "$($_.FirstName) $($.LastName)" -LicenseAssignment 'i-
ozag338:ENTERPRISEPACK_B_PILOT' -LicenseOptions $options -UsageLocation ch -Password
Password1}

```



```

Administrator: Windows PowerShell
PS P:\>
PS P:\> Import-Csv -Path C:\Users\marcel.geissmann\Desktop\Add_User\test.csv | ForEach-Object { New-MsolUser -UserPrincipalName $_.UserPrincipalName -FirstName $_.FirstName -LastName $_.LastName -DisplayName "$($_.FirstName) $($.LastName)" -LicenseAssignment 'iozag338:ENTERPRISEPACK_B_PILOT' -LicenseOptions $options -UsageLocation ch -Password Password1}

```

| Password  | UserPrincipalName             | DisplayName | isLicensed |
|-----------|-------------------------------|-------------|------------|
| Password1 | test.user@iozag338.onmicro... | Test User   | True       |
| Password1 | test.ueser@iozag338.onmicr... | Test User   | True       |

Der gesamte Ablauf gibt es per Script „**AddUser.ps1**“ unter:



AddUser.txt

## 4. Benutzer zu Sicherheitsgruppen hinzufügen

Um einen User oder mehrere zu einer Sicherheitsgruppe hinzuzufügen, müssen wir zuerst die gewünschte Gruppen ID wählen und uns die ObjectID merken. Ebenfalls brauchen wir die „UserPrincipalName“.

„ Get-MsolGroup

```

Administrator: Windows PowerShell

PS P:\> Get-MsolGroup

ObjectID                               DisplayName                               GroupType                               Description
-----
4f5800e1-b71e-4956-b816-8851bbc852ff  SP_intranet_geschaeftsl... Security
c000ee78-721c-466e-b31e-1742e727ad1f  sp_home_admin Security
6f6752ee-343d-48c7-ae96-b5ed00a63f68  sp_home_write Security
bf18f140-2e8b-4473-91e1-a8b1d1ca705d  sp_home_read Security
00006d2e-9595-43f7-b0d9-b0a730eb10e6  sp_management_admin Security
0b9aeefa-9412-4390-af6a-4d623a3eb6e9  sp_management_write Security
1c7aea67-f3f3-47d6-9b83-92f4960d2bbd  sp_management_read Security
495fdda0-efc0-42a6-bf86-6e5c374b2046  Management AG Security
2d0d380e-9071-409a-8196-68f20fb6aa9b  Production AG Security
6c82c26c-9d95-46cf-a2f6-0674efc4ccf  Spezial AH Security
  
```

„ Get-MsolUser

```

Scroll Administrator: Windows PowerShell

PS P:\> Get-MsolUser

UserPrincipalName                       DisplayName                               isLicensed
-----
fabian@IOZAG338.onmicrosoft.com         Fabian Häfliger                          True
heinz.suess@iozag338.onmicrosoft.com    Heinz Suess                              True
david.mehr@iozag338.onmicrosoft.com     David Mehr                                True
paul.meister@IOZAG338.onmicrosoft.com   Peter                                     True
test.user@iozag338.onmicrosoft.com      Test User                                True
hans.muster@IOZAG338.onmicrosoft.com    Hans Muster                              True
marco.loetscher@iozag338.onmicrosoft... Marco Lötischer                          False
dilshansam@iozag338.onmicrosoft.com    Dilshan Sam                              True
test.ueser@iozag338.onmicrosoft.com     Test User                                True
  
```

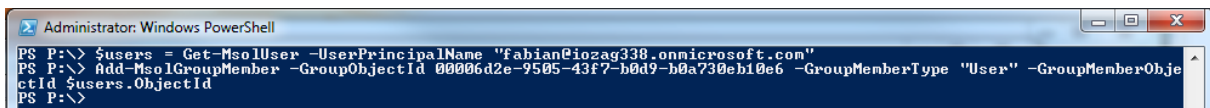
## 4.1 Ein Benutzer zur Sicherheitsgruppe hinzufügen

Mit diesem Code kann nun ein Benutzer einer Sicherheitsgruppe hinzugefügt werden. Wobei der Name nach dem -UserPrincipalName angepasst werden muss:

```
„ $users = Get-MsolUser -UserPrincipalName "fabian@iozag338.onmicrosoft.com"
```

Dann kann das eigentliche Hinzufügen vollzogen werden:

```
„ Add-MsolGroupMember -GroupId 00006d2e-9505-43f7-b0d9-b0a730eb10e6 -  
GroupMemberType "User" -GroupMemberObjectId $users.ObjectId
```



```
Administrator: Windows PowerShell  
PS P:\> $users = Get-MsolUser -UserPrincipalName "fabian@iozag338.onmicrosoft.com"  
PS P:\> Add-MsolGroupMember -GroupId 00006d2e-9505-43f7-b0d9-b0a730eb10e6 -GroupMemberType "User" -GroupMemberObje  
ctId $users.ObjectId  
PS P:\>
```

## 4.2 Mehrere Benutzer zu Sicherheitsgruppen hinzufügen

Um mehrere Benutzer gleichzeitig per Filter zu einer Sicherheitsgruppe hinzuzufügen kann folgendes eingegeben werden: Die variablen Elemente „sp\_Management\_admin“ und „Information Technology“ den Bedürfnissen anpassen.

```
„ $selectGroup = Get-MsolGroup | where-object { $_.DisplayName -eq "sp_management_admin"}  
$users = Get-MsolUser -Department "Information Technology"  
ForEach ($user in $users)  
{  
Add-MsolGroupMember -GroupId $selectGroup.ObjectId -GroupMemberType "User" -  
GroupMemberObjectId $user.ObjectId  
}
```

Der gesamte Ablauf gibt es per Script „**AddUserToGroup.ps1**“ unter:



AddUserToGroup.txt