

# PowerShell and Office 365

---

Presentation created for [Simplex-IT](#)

Developed by Sarah Dutkiewicz

# Agenda

---

- Prerequisites
- Signing On
- Company Information
- Subscriptions
- Users
- Groups & Roles

*This presentation is covering cmdlets specific to the Office 365 core management. We will not be covering any cmdlets related to Exchange Online, SharePoint Online, or Lync Online*

# Prerequisites

---

- Minimum Operating System: Windows 7 or Windows Server 2008 R2
- Minimum Software Requirements: PowerShell and .NET Framework 3.5.1
- Microsoft Online Services Sign-In Assistant (IDCRL7):
  - [32-bit version](#)
  - [64-bit version](#)
- Microsoft Online Services Module for Windows PowerShell / Windows Azure Active Directory Module for Windows PowerShell
  - [32-bit version](#)
  - [64-bit version](#)

# The Office 365 PowerShell Module

---

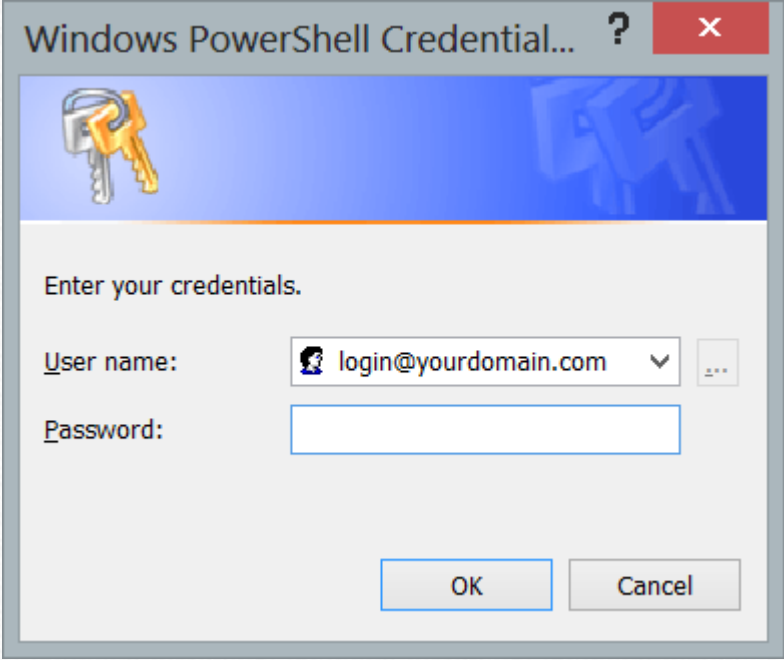
- MSOnline

*Remember to import this into your PowerShell session with the **Import-Module** command if you aren't running PowerShell from the Windows Azure Active Directory Module for Windows PowerShell shortcut..*

# Connecting to Office 365

---

Connect-MsolService



The screenshot shows a Windows PowerShell Credential dialog box. The title bar reads "Windows PowerShell Credential..." with a question mark and a close button. The dialog features a blue header with a key icon. Below the header, it says "Enter your credentials." There are two input fields: "User name:" with a dropdown menu showing "login@yourdomain.com" and a small icon to its right, and "Password:" with an empty text box. At the bottom right, there are "OK" and "Cancel" buttons.

# List Company Information

---

`Get-MsolCompanyInformation`

# Update Company Contact Information

---

`Set-MsolCompanyContactInformation`

*Set your technical and marketing contacts with this cmdlet.*

# Directory Synchronization

---

`Set-MsolDirSyncEnabled`

*Not sure what your company is currently set at? Use the following command:*

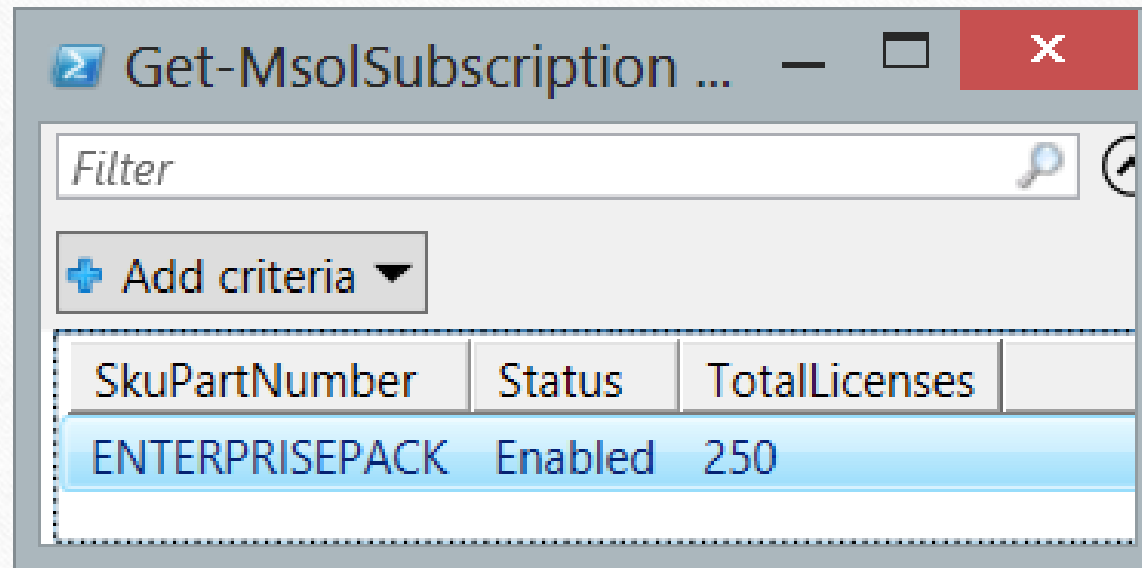
**`Get-MsolCompanyInformation | Select DirectorySynchronizationEnabled`**



# Subscription Information

---

Get-MsolSubscription

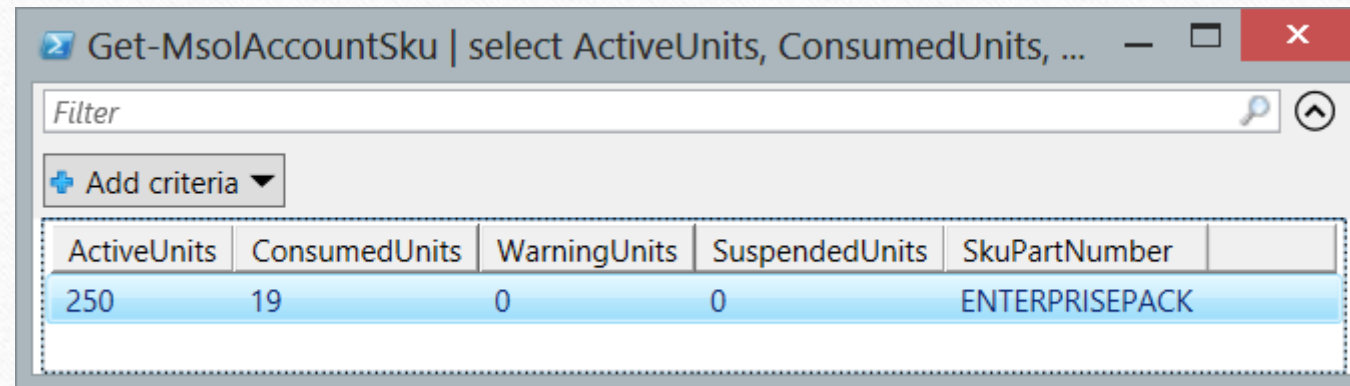


SkuPartNumber	Status	TotalLicenses
ENTERPRISEPACK	Enabled	250

# Account SKUs

---

## Get-MsolAccountSku



ActiveUnits	ConsumedUnits	WarningUnits	SuspendedUnits	SkuPartNumber
250	19	0	0	ENTERPRISEPACK

# List Users

---

## Get-MsolUser

- Use the **EnabledFilter** parameter to show a particular type (*All, EnabledOnly, DisabledOnly*)
- Use the **MaxResults** parameter to limit how many results are shown.
- Use the **UserPrincipalName** parameter to search by UPN.
- Use the **SearchString** parameter to search the display name or email address fields.

# List Users

---

```
PS C:\windows\system32> Get-MsolUser | where DisplayName -Like '*Jones#'
```

UserPrincipalName	DisplayName	isLicensed
sjones@simplex-it.com	Samuel Jones	False
sajones@simplex-it.com	Sarah Jones	False

# Create a New User

---

```
New-MsolUser -UserPrincipalName tom.jones@simplex-it.com -DisplayName  
"Tom Jones"
```

Password	UserPrincipalName	DisplayName	isLicensed
----- Qof06412	----- tom.jones@simplex-i...	----- Tom Jones	----- False

# Update User Properties

---

```
Set-MsolUser -UserPrincipalName yourname@yourdomain.com  
-UsageLocation US
```

The **UsageLocation** parameter should be the 2-character ISO country code for the user.

# Set User Licenses

---

## Set-MsolUserLicense

- *The UsageLocation field needs to be set before a license can be assigned to a user.*
- *Use this cmdlet to add or remove licenses. You can add and remove licenses in one operation using the respective **AddLicense** and **RemoveLicense** parameter. These parameters take a SkuPartNumber for their values.*

# Change User's Password

---

```
Set-MsolUserPassword -UserPrincipalName user@domain.com  
-NewPassword "Ch4ng3m3"
```



# Remove a User

---

## Remove-MsolUser

- *You can pipe multiple users to this command using the `Get-MsolUser` cmdlet, if you need to remove multiple users at a time.*
- *Use the **Force** parameter to suppress the confirmation prompt.*
- *Removing the user puts them in a recycle bin for 30 days (by default).*
- *To remove them from the recycle bin, use the **RemoveFromRecycleBin** parameter. Once a user is deleted from the recycle bin, they cannot be restored.*

# Viewing the Users in the Recycle Bin

---

```
Get-MsolUser -ReturnDeletedUsers
```

# Restoring a User

---

## Restore-MsolUser

- *You can pipe multiple users to this command using the `Get-MsolUser` cmdlet, if you need to restore multiple users at a time.*
- *Use the **AutoReconcileProxyConflicts** parameter when a restore fails due to proxy conflicts.*
- *If a previous UPN is already active for another user, you'll need to assign the restored user a new UPN. For this purpose, use the **NewUserPrincipalName**.*

# Change a User ID

---

```
Set-MsolUserPrincipalName -UserPrincipalName  
oldname@domain.com -NewUserPrincipalName newname@domain.com
```

- *Use this to change a user's principal name. For example, if you're already in PowerShell and need to change someone's UPN. Rather than open the GUI and go through the process, just use this cmdlet!*

# View the Password Policy

---

`Get-MsolPasswordPolicy`

# Change the Password Policy

---

## Set-MsolPasswordPolicy

- Use the **ValidityPeriod** parameter to specify how long a password is valid.
- Use the **NotificationDays** parameter to specify how long before a user is notified to change their password.

# Security Groups

---

- 
- Add-MsolGroupMember
- Get-MsolGroup
- Get-MsolGroupMember
- New-MsolGroup
- Remove-MsolGroup
- Remove-MsolGroupMember
- Set-MsolGroup

# List Groups

---

Get-MsolGroup



# Add a New Group

---

New-MsolGroup

# Add Members to a Security Group

---

## Add-MsolGroupMember

- *Store the group's ID in a variable, as it will help.*
- ***GroupMemberType** values include User, Contact, Group, Service Principal, and Other*

# List Group Members

---

Get-MsolGroupMember

- *This again requires a groupId.*

# Remove a Security Group Member

---

```
Remove-MsolGroupMember -GroupObjectId $groupId -  
GroupMemberObjectId $userId -GroupMemberType User
```

- *Store the user ID in a variable. (\$userId)*
- *Store the group ID in a variable (\$groupId)*

# Update a Security Group

---

`Set-MsolGroup`

- *Change display name and description*

# Remove a Security Group

---

## Remove-MsolGroup

- *This cmdlet runs a confirmation prompt. Use the **Force** to suppress the prompt.*

# Roles

---

- Add-MsolRoleMember
- Get-MsolRole
- Get-MsolRoleMember
- Remove-MsolRoleMember
- Get-MsolUserRole

# List Administrator Roles

---

`Get-MsolRole`



# Get All Users in a Role

---

`Get-MsolRoleMember`

# Get Roles for a User

---

`Get-MsolUserRole`

# Add Users to a Role

---

```
Add-MsolRoleMember -RoleName "Billing Administrator"  
-RoleMemberEmailAddress "name@domain.com"
```

# Remove Users from a Role

---

```
Remove-MsolRoleMember -RoleName "Billing Administrator"  
-RoleMemberEmailAddress "name@domain.com"
```

# Contact Information

---

- General feedback on the webinar: [Bob@Simplex-IT.com](mailto:Bob@Simplex-IT.com)
- Office 365 PowerShell questions (or even general PowerShell questions):  
[Sarah@Simplex-IT.com](mailto:Sarah@Simplex-IT.com)