

Managing Office 365 in PowerShell

Remote Office

Microsoft offers its Office programs as a service in the cloud. To integrate the Office 365 service into your own infrastructure, PowerShell is a natural choice. *By Thomas Joos*

Companies are increasingly opting for cloud services like those offered by Microsoft with Office 365. These services are usually managed via a web interface. However, extensions for PowerShell offer administrators the ability to retrieve data and adjust settings.

To manage Office 365 in PowerShell, you must download and install two extensions. Next, load the normal PowerShell and connect this session with your Office 365 subscription. You first need to download and install the Microsoft Online Services Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell. The downloads can be found online [1] [2] [3]. Make sure that you download the correct version for your setup – that is, 32-bit or 64-bit. You also need at least version 3.5.1 of the .NET Framework in place on your computer.

Problems that occur during the setup and installation most likely relate to incompatibilities between the Online Services Sign-in Assistant, the Windows Azure Active Directory Module for Windows PowerShell, and your Office 365 subscription. Always install the latest versions of these applications from the sites I just mentioned. You can also complete the setup in your Office 365 portal's Software section [4]. If errors occur, uninstall the individual modules in Windows using `appwiz.cpl` and then install the latest versions again.

If you still cannot log into Office 365, you most likely have an issue with the Microsoft Online Services Sign-in Assistant. In this case, either install the current beta version or use the current official version from the site [5]. Also make sure that you install the correct language. You cannot run both versions in parallel on a com-

puter. You must always uninstall the other version (`appwiz.cpl`) before you install the successor.

When you launch PowerShell for the first time as an administrator or with a connection to Office 365 or Windows Azure, you also need to allow script execution. To do this, enter the command `Set-ExecutionPolicy RemoteSigned` on.

Connecting to the Cloud

To import the commandlets for managing Office 365, enter the `Import-Module MSOnline` command in PowerShell. You can use this command and later also the Office 365 commandlets in PowerShell ISE, of course. To be able to access your Office 365 subscription with the commandlets, you must first log into Office 365 and authenticate. The `Connect-MSOLService` commandlet does

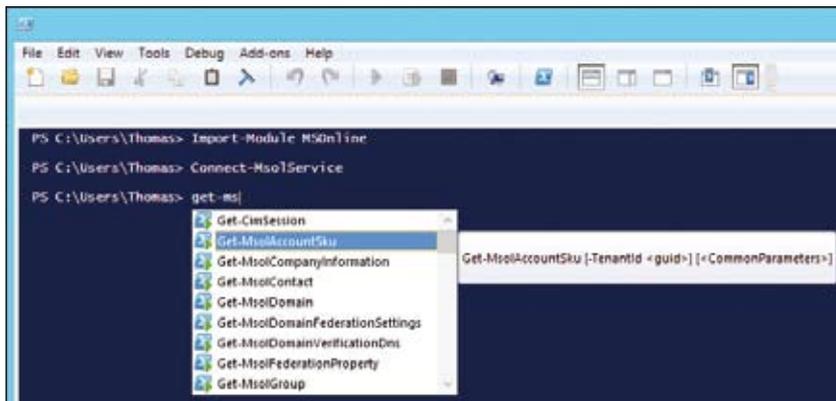


Figure 1: PowerShell displays help when you enter commands.

this for you. In the login window that appears, enter the address and password of an Office 365 administrator. You will not get any feedback about the successful connection. That commandlet displays an error if the connection fails.

One advantage of using PowerShell ISE is the built-in help. If you enter a commandlet, the program displays the possible commands and options (Figure 1). Most commands for managing Office 365 in the PowerShell contain the `msol` string. If you want to view a list of existing commandlets for managing Office 365, just type `get-command *msol*`.

You can save the credentials for Office 365 in a variable. To do this, type `$aut = Get-Credential`, for example. In the dialog that appears, enter the credentials you want to store in the variable. The `Connect-MsolService -Credential $aut` command then opens the connection again.

Overview

With PowerShell, you can access important information significantly faster than with the graphical user interface. `Get-MsolDomain` shows the domains that you use in your Office 365 subscription. You can also see the status of the domain, that is, whether it is verified by Office 365 and managed.

`Get-MsolDomainVerificationDns` queries data for the domain. If everything is OK, you will not see any feedback. If the domain is not connected to your Office 365 subscription, an error mes-

sage appears. You can also remove domains via PowerShell and specify the default domain. To delete domains, you must first change the default domain. The following example illustrates the changes:

```
Set-MsolDomain -Name >
    contoso.onmicrosoft.com -IsDefault
```

Once you have changed the default domain, you can delete unnecessary domains:

```
Remove-MsolDomain >
    -DomainNamecontoso.com
```

You should never delete the default domain, `domain.onmicrosoft.com`, which is required in part by internal services. The same applies to `mail.domain.onmicrosoft.com`, which is used for example by the Exchange Online Protection antivirus in Office 365. This also applies to Hybrid Deployments with Exchange 2010/2013.

Licenses

To manage Office 365, you first need to store some important information about your subscription, the stored users, and configured settings. The most important point in managing

Office 365 is certainly licensing. You can manage the licenses in the web interface in Licenses administration section.

You can delegate the management of licenses in larger environments, for example, to accounts or controlling. However, this is only possible in larger editions of Office 365. Just click on Settings in the Licenses section and select User Management Administrator. There, select the user that you want to manage the licenses. This user can then also use PowerShell to view and manage licenses. With PowerShell, you can, for example, display clear-cut tables, which is not possible in the graphical interface. To display all roles available in your Office 365 subscription and their functions in PowerShell, type `Get-MsolRole`.

Remember that the User Management Administrator user role has advanced rights for user management features, such as password resets. `Get-MsolUser` displays information on your users. If you use the `|ft` option, like with other commandlets, you will see a formatted table, while `|fl` displays a formatted list. You can follow the `|ft` or `|fl` option with the columns that you want Powershell to display. For example, to display the licensed user, use `get-msoluser |ft UserPrincipalName, Displayname, *lic*` (Figure 2). The `-AutoSize` option lets you adjust the table to match the window size. In PowerShell, you can also retrieve the licensing model and the Office 365 Edition that your company uses. To do this, use the `Get-MsolAccountSku` commandlet. The `Get-MsolSubscription` commandlet also displays more information about your subscription. Contact information, technical emails, and other data can be retrieved using `Get-MsolCompanyInformation`. To manage the licenses of a specific user, use:

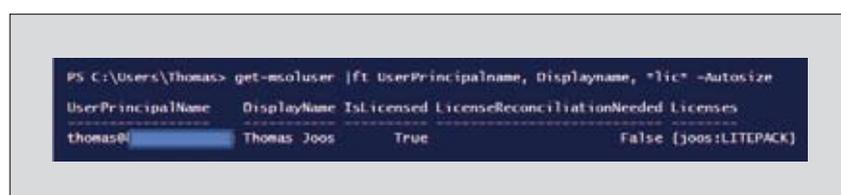


Figure 2: In PowerShell, you can view the licenses used in your Office 365 subscription.

```
get-msoluser -userprincipalname"UPN" | ?
  select licenses, islicensed| fl
```

Of course, you can also assign licenses directly:

```
set-msoluserlicense ?
  -userprincipalname"UPN" ?
  -addlicenses "licensed name"
```

For a detailed list of the options of Set-MsolUserLicense see the TechNet library [6]. Information on batch changing of licenses is also available [7].

Managing Users

In addition to the commandlets already mentioned, other commands let you create, manage, and delete users in Office 365. The most important commandlets in this area include:

- **New-MsolUser** – Create a new user. For example, to create a new user with a short command, use `New-MsolUser -userPrincipalName "email address" -DisplayName "display name"`. The commandlet generates the password and displays it.
- **Remove-MsolUser** – Delete the user and return the assigned licenses. After that, the user can still sign in for 30 days.
- **Restore-MsolUser** – The user's original state is restored. This works up to 30 days after deletion. However, you can permanently delete mailboxes to prevent them from being restored.
- **Set-MsolUser** – Update a user and change their settings.
- **Set-MsolUserPassword** – Change the password.

To view all of the user management commandlets, type `get-command *msoluser*` (Figure 3). The commands available for group administration are shown by typing `get-command *msolgroup*`.

For example, if you want to specify that a user password never expires, use `Set-MsolUser -UserPrincipalName Username -PasswordNeverExpires $True`. Without this option, the cor-

responding user must change their password every 90 days. `$False` lets you reverse this option. It is also possible to activate complex passwords for a user, which is what the `-StrongPasswordRequired` option does:

```
Set-MsolUser ?
  -UserPrincipalNameBenutzername-?
  StrongPasswordRequired $True
```

`$False` disables this function. As mentioned earlier on, a user resides for 30 days in the Recycle Bin when you delete them and can be restored from the Recycle Bin using `Restore-MsolUser`. The following command displays deleted mailboxes that can be restored:

```
Get-MsolUser -ReturnDeletedUsers | ?
  fl UserPrincipalName, ObjectID
```

If you want to finally delete a user, use their `ObjectID`:

```
Get-MsolUser -Remove-MsolUser ?
  -ObjectID GUID ?
  -RemoveFromRecycleBin -Force
```

Additionally, you can use this command:

```
Get-MsolUser -ReturnDeletedUsers | ?
  Remove-MsolUser ?
  -RemoveFromRecycleBin-Force
```

to erase individual users and also remove all users from the Recycle Bin.

Connecting to the Exchange Management Shell

Office 365 accounts and a local Exchange server are used in parallel in many environments. It thus makes sense to make the commands from the Exchange Management Shell available in PowerShell in addition to Office 365 commandlets. To do this, first store the Office 365 login credentials in a variable:

```
$cred = Get-Credential
```

Open a connection to Office 365 and the Exchange Management Shell:

```
$Session = New-PSSession ?
  -ConfigurationName ?
  Microsoft.Exchange-ConnectionUri ?
  https://ps.outlook.com/powershell ?
  -Credential $cred-Authentication Basic ?
  -AllowRedirection
```

Then, import these data with `import-PSSession $Session`.

Exchange Online Protection

Microsoft offers the cloud-virus scanner Exchange Online Protection (EOP) for local Exchange installations and for Office 365. This tool is the successor of Forefront Online Protection for Exchange (FOPE). In Office 365, Exchange Online Protection is automatically enabled, which means that incoming and outgoing emails are scanned for viruses and spam. If you also have local Exchange servers, you can send the emails from your local servers to Office 365, where they are scanned for viruses and then forwarded. This approach also works in both directions with Exchange Server 2013 and with the virus scanner integrated in Exchange 2013.

The following command ensures that Exchange Server 2013 will scan email messages after Exchange Online Protection emails has scanned them:

```
Set-MailwareFilteringServer ?
  -ForceRescan $true
```

To check the value, enter:

```
Get-MailwareFilteringServer | ?
  Format-List Name, ForceRescan
```

```
PS C:\Users\Thomas> get-command *msoluser*
CommandType      Name
-----
Cmdlet            Get-MsoUser
Cmdlet            Get-MsolUser
Cmdlet            Get-MsolUserByStrongAuthentication
Cmdlet            Get-MsolUserByStrongAuthentication
Cmdlet            Get-MsolUserRole
Cmdlet            Get-MsolUserRole
Cmdlet            New-MsolUser
Cmdlet            Remove-MsolUser
Cmdlet            Remove-MsolUser
Cmdlet            Restore-MsolUser
Cmdlet            Restore-MsolUser
Cmdlet            Set-MsolUser
Cmdlet            Set-MsolUser
Cmdlet            Set-MsolUserLicense
Cmdlet            Set-MsolUserLicense
Cmdlet            Set-MsolUserPassword
Cmdlet            Set-MsolUserPassword
Cmdlet            Set-MsolUserPrincipalName
Cmdlet            Set-MsolUserPrincipalName
```

Figure 3: PowerShell has a number of commands for managing users in Office 365.

This redirection between Exchange and Office 365 is controlled by the MX record in the domain. The wizard for connecting Office 365 with a local Exchange installation adds an additional accepted domain to the Exchange organization for hybrid messaging. This domain is added as a secondary proxy domain. By default, this is the `domain.mail.onmicrosoft.com` domain. You can view the accepted domain with:

```
Get-AcceptedDomain | FL
DomainName, IsCoexistenceDomain
```

In the Exchange Management Shell, you can view the settings by typing `Get-Hybrid Configuration`.

Besides managing the Office 365 services in the PowerShell, you can also manage other cloud services from Microsoft, such as Windows Azure. The installation file for managing Windows Azure in PowerShell can be found online [8]. After the install, just import the commandlets by typing `Import-Module Azure`. Before you can manage Azure with PowerShell, you must modify your PC so that it connects to Windows Azure. To begin, enter `Get-AzurePublishSet-`

`tingsFile` and then log into Windows Azure. Download the file, which the wizard then displays. Enter the `Import-AzurePublishSettingsFile Publishsettings-File` command followed by `Get-AzureSubscription`.

Microsoft offers the Windows Azure Backup service, which lets you backup data from servers in the cloud. In PowerShell, you can view the available commandlets for Windows Azure Backup with `get-command *ob*` (Figure 4). Alternatively, you can use the `get-command -module MSOnlineBackup` command. More information about Azure Backup is available in a previous article [9].

Conclusions

Microsoft's PowerShell provides a powerful shell with many options for managing local servers. In addition to the standard operating systems by Microsoft, all current server applications, such as Exchange, SQL, SharePoint, and Lync, support PowerShell-based administration. If your company additionally uses cloud services like Windows Azure and Office 365, it makes sense to manage them in PowerShell, too. The advantages of this approach are obvious: Administrators

can manage local servers and cloud services in the same session. Additionally, many options and features in PowerShell are much faster and easier to handle than in web interfaces. If you have some experience with scripting, you can also compose scripts for comprehensive management of your services. These scripts can also perform changes in local and cloud services in parallel. Because Microsoft provides the extensions for PowerShell free of charge, administrators should seriously look into their options. With a little training, you can use PowerShell to quickly and easily manage many services – including ones that run in the cloud. ■

Info

- [1] Microsoft Online Services Sign-In Assistant: [<http://search.microsoft.com/en-gb/DownloadResults.aspx?q=microsoft%20online%20services%20sign-in%20assistant>]
- [2] Azure AD module (32-bit): [<http://go.microsoft.com/fwlink/p/?linkid=236298>]
- [3] Azure AD module (64-bit): [<http://go.microsoft.com/fwlink/p/?linkid=236297>]
- [4] Office 365 login: [<https://portal.microsoftonline.com/OLS/MySoftware.aspx>]
- [5] Microsoft Online Services Sign-In Assistant for IT Professionals RTW: [<http://www.microsoft.com/en-gb/download/details.aspx?id=28177>]
- [6] Set-MsolUserLicense: [<http://technet.microsoft.com/en-us/library/dn194094.aspx>]
- [7] Changing Office 365 licenses with PowerShell: [<http://technet.microsoft.com/en-us/library/dn530771.aspx>]
- [8] Azure downloads: [<https://www.windowsazure.com/en-us/manage/downloads/>]
- [9] "Windows Azure Backup" by Thomas Joos, ADMIN, January, 2014: [<http://www.admin-magazin.de/Das-Heft/2014/01/Windows-Azure-Backup-nutzen>] (in German)

The Author

Thomas Joos is a freelance IT consultant and has been working in IT for over 20 years. In addition to his projects, he also writes hands on books and papers on Windows and other Micro-

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator_C0H10SD> get-command -module MSOnlineBackup

CommandType      Name                                     ModuleName
-----
Cmdlet            add-OBFileSpec                          MSOnlineBackup
Cmdlet            Get-OBAllRecoveryPoints                 MSOnlineBackup
Cmdlet            Get-OBAlternateBackupServer             MSOnlineBackup
Cmdlet            Get-OBCertificateListFromLocalStore     MSOnlineBackup
Cmdlet            Get-OBFileSpec                          MSOnlineBackup
Cmdlet            Get-OBJob                               MSOnlineBackup
Cmdlet            Get-OBMachineSetting                   MSOnlineBackup
Cmdlet            Get-OBMachineUsage                     MSOnlineBackup
Cmdlet            Get-OBNotification                     MSOnlineBackup
Cmdlet            Get-OBPolicy                            MSOnlineBackup
Cmdlet            Get-OBPolicyState                      MSOnlineBackup
Cmdlet            Get-OBRecoverableItem                  MSOnlineBackup
Cmdlet            Get-OBRecoverableSource                 MSOnlineBackup
Cmdlet            Get-OBRecoveryService                   MSOnlineBackup
Cmdlet            Get-OBRetentionPolicy                  MSOnlineBackup
Cmdlet            Get-OBSchedule                          MSOnlineBackup
Cmdlet            New-OBFileSpec                          MSOnlineBackup
Cmdlet            New-OBPagingContext                     MSOnlineBackup
Cmdlet            New-OBPolicy                            MSOnlineBackup
Cmdlet            New-OBRecoverableItem                  MSOnlineBackup
Cmdlet            New-OBRecoveryOption                    MSOnlineBackup
Cmdlet            New-OBRetentionPolicy                  MSOnlineBackup
Cmdlet            New-OBSchedule                          MSOnlineBackup
Cmdlet            Remove-OBFileSpec                       MSOnlineBackup
Cmdlet            Remove-OBPolicy                         MSOnlineBackup
Cmdlet            Rename-OBVolume                         MSOnlineBackup
Cmdlet            Set-OBMachineSetting                   MSOnlineBackup
Cmdlet            Set-OBPolicy                            MSOnlineBackup
Cmdlet            Set-OBPolicyState                      MSOnlineBackup
Cmdlet            Set-OBRetentionPolicy                  MSOnlineBackup
Cmdlet            Set-OBSchedule                          MSOnlineBackup
Cmdlet            Start-OBBackup                          MSOnlineBackup
Cmdlet            Start-OBRecovery                        MSOnlineBackup
Cmdlet            Start-OBRegistration                    MSOnlineBackup
Cmdlet            Stop-OBJob                              MSOnlineBackup
```

Figure 4: Displaying the available commands for Microsoft Online Backup in the PowerShell.