

SUMMARY

This article discusses how to use PowerShell to configure mailboxes for users in Exchange Online in Microsoft Office 365. It contains information about how to use PowerShell to do the following:

CONTENTS

Summary.....	1
Connect to Exchange Online by using Remote PowerShell	2
Load Powershell Commands.....	2
Configure Send As permissions	2
Configure "Send on behalf" permissions	3
Configure email forwarding	3
Grant full mailbox access	4
Grant Receive As access	4
View Mailbox Permissions	4
Remove mailbox permissions	5
Set the Time Zone of a Mailbox.....	5
Produce a list of Users	5
To add an Apostrophe to a Mailbox name	5
To Create Personal Archive for everyone	6
Change Name of Archive to "Personal Archive - [display name].....	6
Changing a group of users passwords to a common password.....	6
Importing External contacts using CSV	7
Set Password to Not Expire	7

CONNECT TO EXCHANGE ONLINE BY USING REMOTE POWERSHELL

To complete any of the common Exchange tasks in Exchange Online, customers can use the self-service options that are provided by Remote PowerShell. To learn how to connect to the Exchange Online PowerShell, see the following article:

[How to use PowerShell to manage an Office 365 environment](#)

LOAD POWERSHELL COMMANDS

```
$Cred = Get-Credential
$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
https://ps.outlook.com/powershell/ -Credential $Cred -Authentication Basic -AllowRedirection
Import-PSSession $Session
Import-Module MSOnline
Connect-MsolService -Credential $Cred
Set-ExecutionPolicy unrestricted
```

CONFIGURE SEND AS PERMISSIONS

To use PowerShell to configure a mailbox so that a user other than the mailbox owner can use that mailbox to send messages, use the following command:

Add-RecipientPermission -Identity <Mailbox ID 1> -Trustee <Mailbox ID 2> -AccessRights SendAs

In this command line, <Mailbox ID 1> represents the mailbox, contact, or distribution group that you want to grant permissions to and <Mailbox ID 2> is the user who you want to grant Send As permission.

To verify that the permissions are applied to Mailbox ID 1, using the following command:

Get-RecipientPermission -Identity <Mailbox ID 1> | Select Trustee, AccessControlType, AccessRights

In the results, you should be able to confirm that <Mailbox ID 2> has been granted Send As permission.

Note These permissions can be assigned to a user mailbox or Active Directory group objects.

CONFIGURE "SEND ON BEHALF" PERMISSIONS

To use PowerShell to grant a user the ability to send mail on behalf of another user, use the following command:

Set-Mailbox -Identity <Mailbox ID 1> -GrantSendOnBehalfTo <Mailbox ID 2>

Set-Mailbox -Identity mservices -GrantSendOnBehalfTo mservicesdg

Set-Mailbox -Identity support -GrantSendOnBehalfTo supportdg

In this command line, <Mailbox ID 1> represents the mailbox that you want to grant permissions to and <Mailbox ID 2> is the mailbox of the user who you want to grant access.

To verify that the permissions are applied to Mailbox ID 1, use the following command:

Get-Mailbox -Identity <Mailbox ID 1> | Select GrantSendOnBehalfTo

In the results, you should be able to confirm that <Mailbox ID 2> has been granted Send On Behalf permission.

Note These permissions can be assigned to a user mailbox or mail-enabled Active Directory security group objects.

CONFIGURE EMAIL FORWARDING

To configure mail forwarding for a mailbox by using PowerShell, use the following command:

Set-Mailbox -Identity "<Mailbox ID>" -ForwardingSmtpAddress "<Target SMTP Address>"

In this command line, <Target SMTP Address> can be either internal or external to Exchange Online.

Additionally, IT Gens should use the **-DeliverToMailboxAndForward** parameter to save a copy of the message to the local mailbox before the message is forwarded to another SMTP address. The following is an example:

Set-Mailbox -Identity "<Mailbox ID>" -ForwardingSmtpAddress "<Target SMTP Address>" -DeliverToMailboxAndForward \$true

To verify that the permissions are applied to Mailbox ID 1, use the following command:

Get-Mailbox -Identity <Mailbox ID 1> | Select ForwardingSmtpAddress

In the results, you will see the address to which email messages for Mailbox ID 1 are being forwarded.

GRANT FULL MAILBOX ACCESS

To grant full mailbox access to a user other than the mailbox owner by using PowerShell, use the following command line:

```
Add-MailboxPermission -Identity <Mailbox ID 1> -User <Mailbox ID 2> -AccessRights FullAccess -InheritanceType All
```

```
Add-MailboxPermission -Identity alan.brown@radiatemediacom -User kevin.miller@radiatemediacom -AccessRights FullAccess -InheritanceType All
```

In this command line, <Mailbox ID 1> represents the mailbox that you want to grant permissions to and <Mailbox ID 2> is the mailbox of the user who you want to grant Full Access permission.

If an administrator wants to grant a user access to a room mailbox, the administrator may want to specify that user as the Owner. To do this, use the following command line:

```
Add-MailboxPermission -Identity <Mailbox ID 1> -Owner <Mailbox ID 2>
```

Note Only users who have Exchange mailboxes can be granted access to other mailboxes. Users who do not have mailboxes receive a permissions error when they try to access the other mailboxes.

To verify that the permissions are applied to Mailbox ID 1, use the following command:

```
Get-MailboxPermission -Identity <Mailbox ID 1> | Select User, AccessRights, Deny
```

```
Get-MailboxPermission -Identity "charterdg" | Select User, AccessRights, Deny
```

In the results, you should be able to confirm that <Mailbox ID 2> has been granted Full Access permission.

GRANT RECEIVE AS ACCESS

To grant Receive As access, use the steps that are described in the **Grant full mailbox access** section. Granting full mailbox access in Office 365 Exchange Online covers both the same permissions that are required for Receive As access.

VIEW MAILBOX PERMISSIONS

```
get-recipientpermission
```

REMOVE MAILBOX PERMISSIONS

To remove any of the permissions that were set earlier as described in this article, use the following cmdlets:

To remove Send As settings from a mailbox, use the following command:

```
Remove-RecipientPermission -Identity <Mailbox ID 1> -AccessRights SendAs -Trustee <Mailbox ID 2>
```

To remove the Full Access configuration on a mailbox, use the following command:

```
Remove-MailboxPermission -Identity <Mailbox ID 1> -User <Mailbox ID 2> -AccessRights FullAccess
```

To remove email forwarding that is configured on a mailbox, use the following command:

```
Set-Mailbox -Identity "<Mailbox ID>" -ForwardingSmtpAddress $NULL
```

To remove Send On Behalf permission from a mailbox, use the following command:

```
Set-Mailbox -Identity <Mailbox ID 1> -GrantSendOnBehalfTo $NULL
```

SET THE TIME ZONE OF A MAILBOX

```
Set-MailboxRegionalConfiguration -Identity <mailboxID> -Language en-us -Timezone "Eastern Standard Time"
```

PRODUCE A LIST OF USERS

```
get-user | export-csv c:\users\userlist.csv
```

```
get-mailbox | Get-MailboxStatistics | fl displayname, LastLogonTime
```

```
Get-Mailbox -ResultSize Unlimited | Get-MailboxStatistics | Select-Object DisplayName, LastLogoffTime, LastLogonTime | Export-Csv -Path "C:\Users\Kevin's\Documents\mailboxreport.csv" -NoTypeInfo
```

TO ADD AN APOSTROPHE TO A MAILBOX NAME



```
Set-msolUserPrincipalName -userprincipalname Michelle.DAngelo@pciservices.com -
newuserprincipalname " Michelle.D'Angelo@pciservices.com"
```

```
Set-mailbox -userprincipalname "Michelle.D'Angelo@pciservices.com" -
newuserprincipalname " Michelle.DAngelo@pciservices.com"
```

TO CREATE PERSONAL ARCHIVE FOR EVERYONE

```
Get-Mailbox -ResultSize unlimited -Filter {(RecipientTypeDetails -eq 'UserMailbox')} | Enable-Mailbox -Archive
```

CHANGE NAME OF ARCHIVE TO "PERSONAL ARCHIVE - [DISPLAY NAME]

```
$users = Get-Mailbox -ResultSize unlimited -Filter {(RecipientTypeDetails -eq 'UserMailbox')}
ForEach ($a in $users) {$a.ArchiveName.Add("Personal Archive - $a")}
$users | %{Enable-Mailbox $_.Identity -Archive -ArchiveName $_.ArchiveName}
```

Question: How is the discovery mailbox created?

Answer: By default a discovery mailbox is created automatically. You can also create other Discovery Mailboxes via PowerShell if needed. See <http://help.outlook.com/en-us/ee424425.aspx?sl=1>

CHANGING A GROUP OF USERS PASSWORDS TO A COMMON PASSWORD

For your situation, I suggest you try to following steps to rest password for users.

1. Create a CSV file with columns UPN and NewPassword on your client computer, (for example, c:\password.csv).
2. Install Microsoft Online Services Module and connect to Office 365 referring the article below.

Install and configure the Microsoft Online Services Module for Windows PowerShell for single sign-on <http://onlinehelp.microsoft.com/en-us/office365-enterprises/ff652560.aspx>

3. Run the following command to reset the password for users listed in file c:\password.csv.

```
Import-Csv "C:\Users\Kevins SS Laptop\Desktop\Mitchell and Mitchell\NewPassword.csv"
| %o{Set-MsolUserPassword -userPrincipalName $_.upn -NewPassword $_.newpassword -
ForceChangePassword $false}
```

IMPORTING EXTERNAL CONTACTS USING CSV

CSV File:

ExternalEmailAddress	Name	FirstName	LastName
joe@thedomain.com	Joe Bloggs	Joe	Bloggs
fred@b1ogg3.com	Fred Blogga	Fred	Blogga

Powershell Command:

```
Import-Csv c:\externalcontacts.csv | %{New-MailContact -Name $_.Name -DisplayName $_.Name -ExternalEmailAddress $_.ExternalEmailAddress -FirstName $_.FirstName -LastName $_.LastName}
```

SET PASSWORD TO NOT EXPIRE

To set password never to expire for only one user please run below commands:

a. **Set-MsolUser -UserPrincipalName <Microsoft Online Services ID> - PasswordNeverExpires \$true**

b. **Get-MSOLUser -UserPrincipalName <Microsoft Online Services ID> | Select PasswordNeverExpires** (verify if it was successful)

---Example---

```
Set-MsolUser -UserPrincipalName john@contoso.onmicrosoft.com -PasswordNeverExpires $true
```

To set password never to expire for all users please run below commands:

- a. **Get-MSOLUser | Set-MsolUser -PasswordNeverExpires \$true**
- b. **Get-MSOLUser | Select UserPrincipalName, PasswordNeverExpires** (verify if it was successful)

ADD AN ALIAS TO A MAILBOX

```
Set-Mailbox -UserPrincipalName "<alias>" -EmailAddresses @("SMTP:joe@contoso.com";"smtp:employee1234@contoso.com";"sip:employee1234@contoso.com ")
```

Note In the command, replace [joe@contoso.com](#) and [employee1234@contoso.com](#) with the appropriate primary SMTP address, secondary smtp address, and SIP address.