Configuring an Exchange service account

The MyTimetable calendar push integration is able to connect to a user's calendar using *service accounts*. This page describes how to create a service account and how to grant calendar permissions to this service account. MyTimetable will then be able to access calendars without explicit consent of a user.

This page applies to both Microsoft Exchange (on-premises) and Microsoft Office 365.

Table of Contents

Office 365 / Azure AD prerequisites

The following installs are required when managing Office 365 / Azure AD through Powershell.

- The Microsoft Online Service Sign-in Assistant for IT Professionals RTW.
- The Azure AD Module for Windows PowerShell:
    - Azure Active Directory Module for Windows PowerShell (32-bit version)
    - Azure Active Directory Module for Windows PowerShell (64-bit version)

You must be a tenant admin on your Office 365 tenant to run the cmdlets.

Connecting to Office 365 using Powershell

We are going to connect to Office 365 using Powershell. For this, we have to set up a remote Powershell session. First, we need to check if we are allowed to do so:

- Check the current script execution policy

```
PS C:\> Get-ExecutionPolicy
Restricted
```

- If we are not allowed to execute remote signed scripts, we have to change the execution policy. It might be required for Powershell to be started as Administrator.

```
PS C:\> Set-ExecutionPolicy RemoteSigned
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 at http://go.microsoft.com
/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
```

Now we are able to start a remote Powershell session to Office 365:

- Connect to Office 365 using your tenant admin account and import the Powershell session:

```
PS C:\> $O365Cred = Get-Credential
PS C:\> $O365Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri https://ps.outlook.com
/powershell -Credential $O365Cred -Authentication Basic -AllowRedirection
WARNING: Your connection has been redirected to the following URI:
"https://ps.outlook.com/PowerShell-LiveID?PSVersion=4.0 "
PS C:\> Import-PSSession $O365Session -AllowClobber
WARNING: The names of some imported commands from the module 'tmp_eiaj1j0m.dcw' include unapproved verbs that
might
make them less discoverable. To find the commands with unapproved verbs, run the Import-Module command again
with the
Verbose parameter. For a list of approved verbs, type Get-Verb.
ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Script     1.0        tmp_eiaj1j0m.dcw                    {Add-AvailabilityAddressSpace, Add-
DistributionGroupMember...
```

Global steps

1. Create one or multiple service accounts, depending on the number of users using the integration.
2. Create a mail-enabled universal security group containing the created service account(s).
3. Delegate calendar permissions to the security group for all users using the integration.

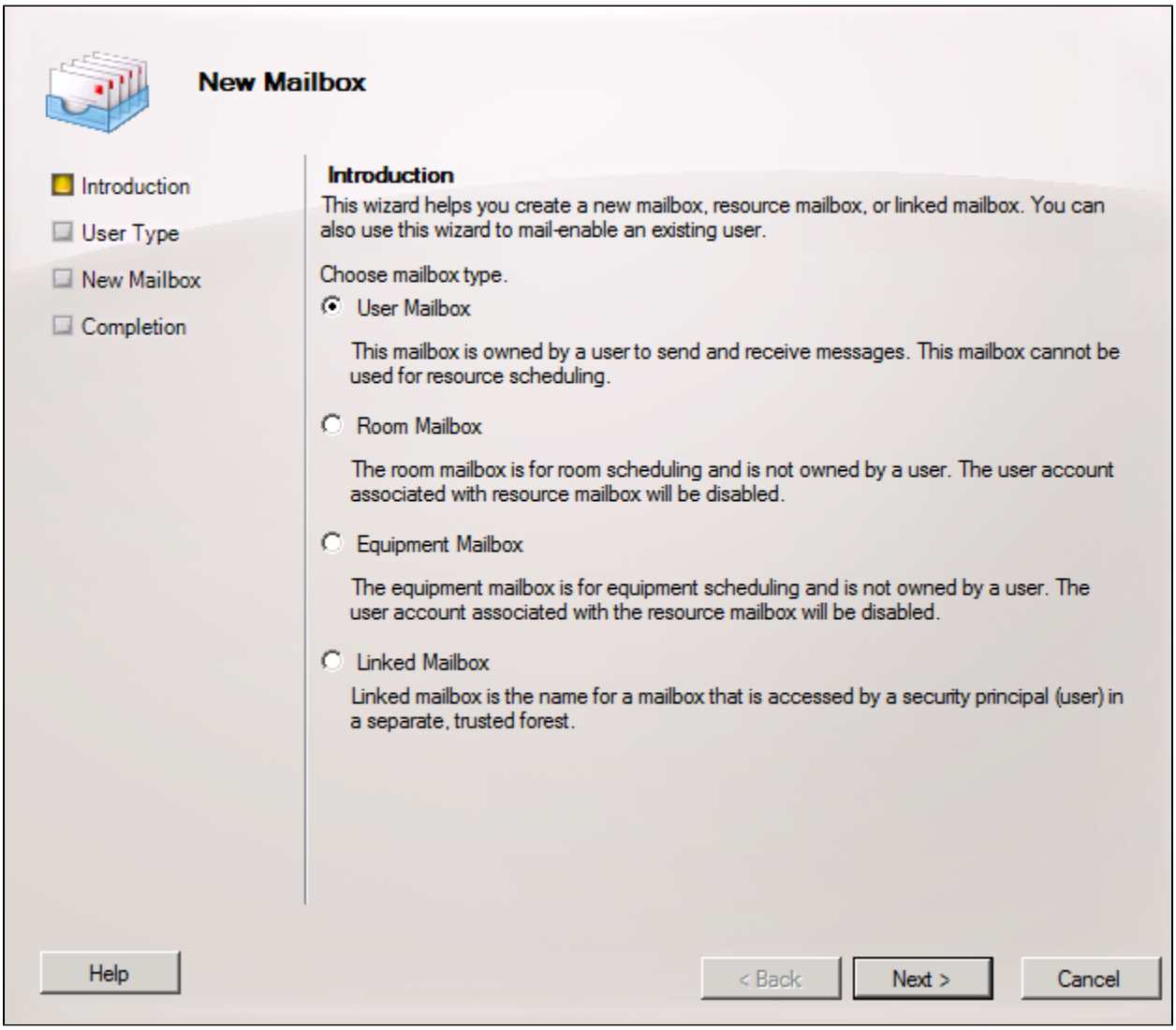
Creating a service account

A service account is needed to access the user's mailboxes. An account can be created in your on-premises Active Directory, or in Azure AD.

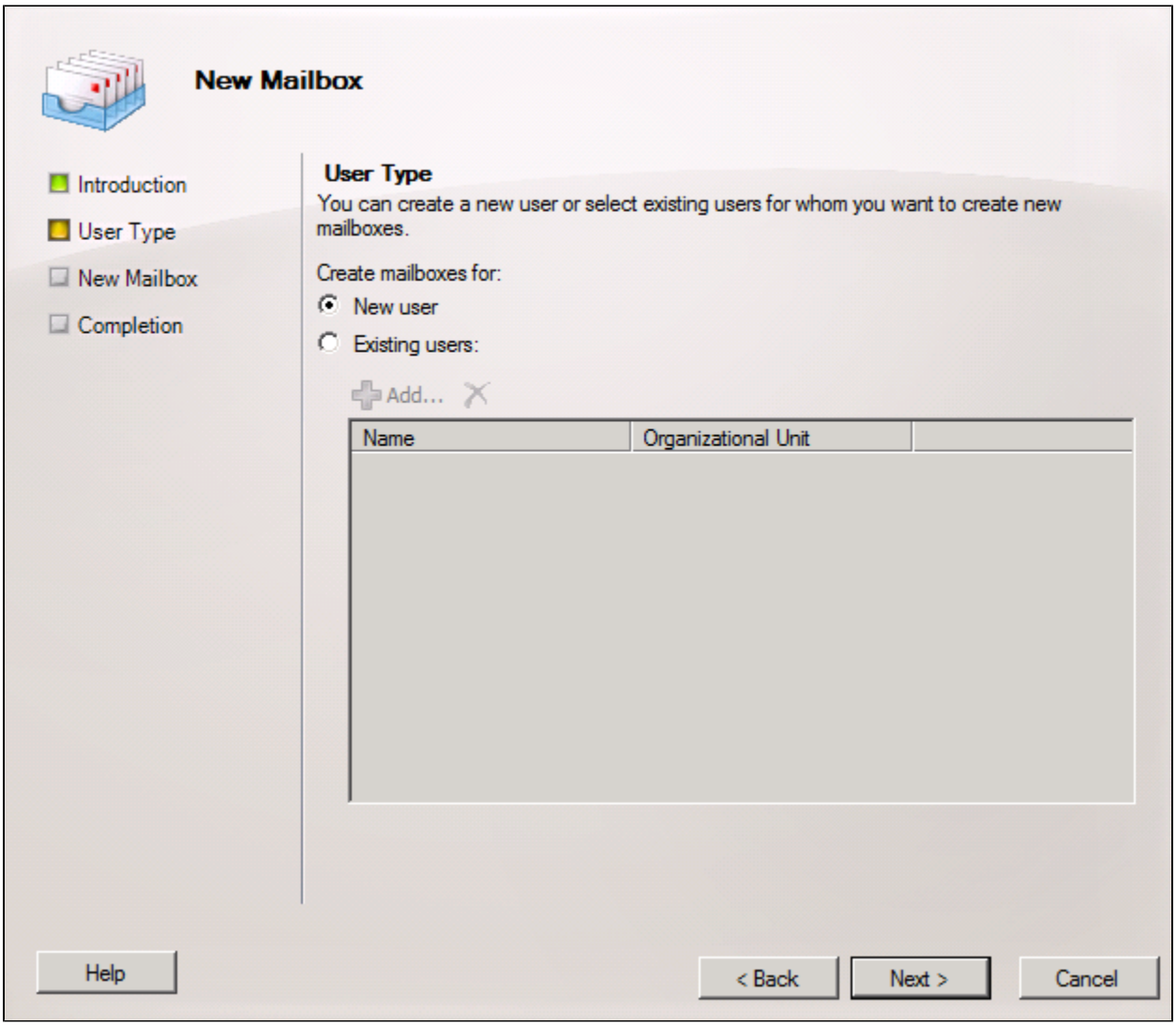
On-premises Exchange

Using the Exchange Management Console

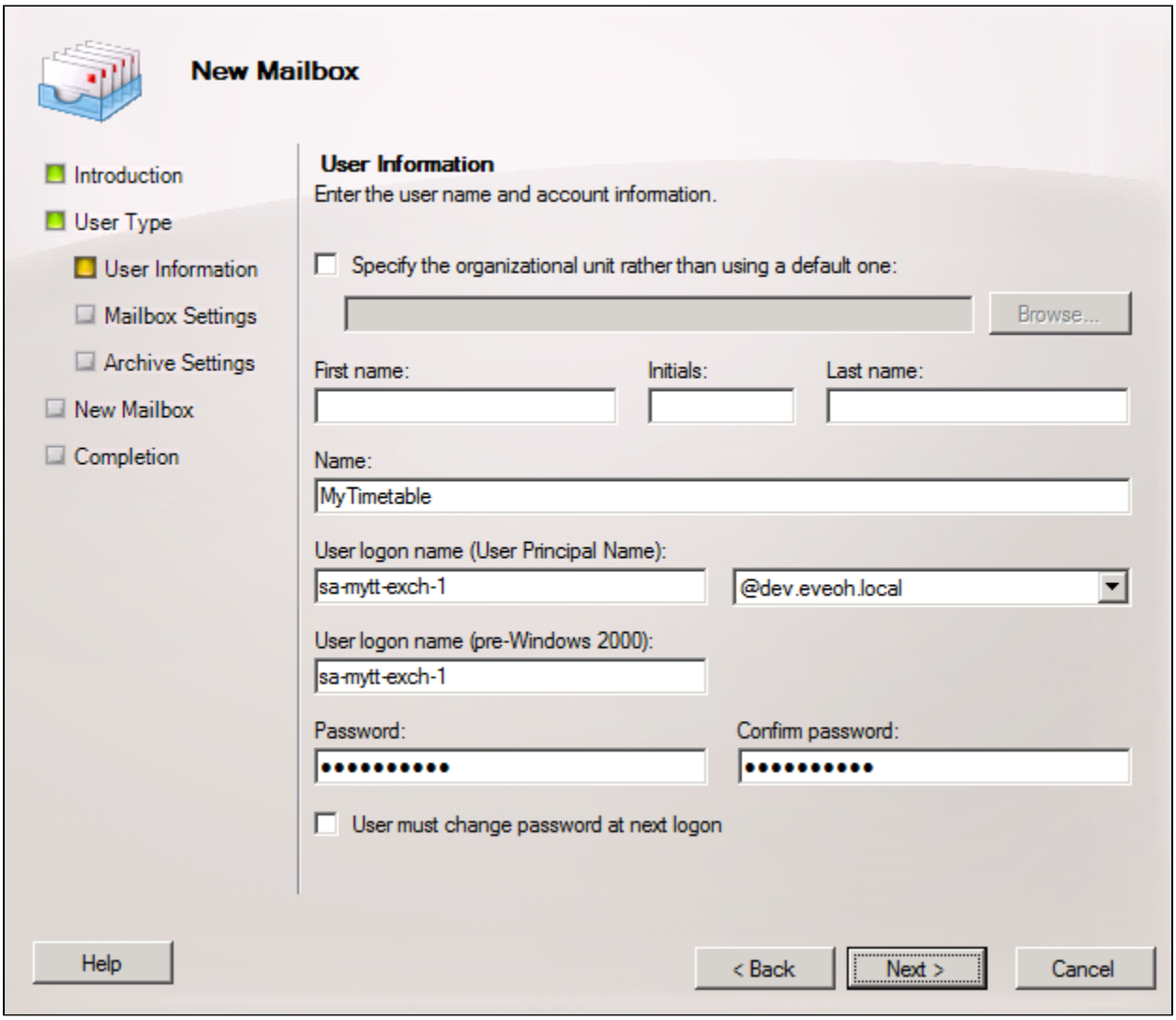
- Create a new Mailbox.
- Choose "User Mailbox" as mailbox type.



- Choose "New User".



- Enter mailbox details. As a logon name use, for example, "sa-mytt-exch-1@eveoh.onmicrosoft.com".



- Click Next.
- Click Next.
- Click New.

Using Powershell

- Open the Exchange Management Shell.
- Create a new mailbox using the New-Mailbox cmdlet. Replace the parameters to match your situation and preferences:

```
PS C:\> New-Mailbox -DisplayName "MyTimetable" -Name "MyTimetable Exchange Service Account 1" -Alias "sa-mytt-exch-1" -UserPrincipalName "sa-mytt-exch-1@dev.eveoh.local" -Password (Read-Host -AsSecureString "Password") -ResetPasswordOnNextLogon $false
Password: *******
```

Office 365

Create the account

Using the Azure Management Portal

- Visit the Microsoft Azure Management Portal at https://portal.azure.com, using the credential of your Microsoft tenant that has the subscription to Office 365 you wish to use.
- Click "Browse all" to browse all resources.
- Click "Activity Directory". You will now be redirected to the classic Azure Management Portal.
- Click the Active Directory you would like to manage.
- Click "Add user" in the bottom bar.

- Select "New user in your organisation" as type of user, and enter a username (e.g. sa-mytt-exch-1).
- Enter a first name, last name and display name (e.g. "MyTimetable"). Select "User" as role. Do not select "Enable Multi-Factor Authentication".
- Click "Create" to assign a temporary password. Write down the password.
- Logout from the Azure Management Portal.
- Go to https://login.microsoftonline.com/
- Login in using the account you have just created, and set a password for the service account.

Using Powershell

Create a service account using the following Powershell command. Of course you can also create an account in the Azure Portal.

- Open the Windows Azure Active Directory Powershell prompt
- Connect to Microsoft Online Services using your tenant admin account:

```
PS C:\> Connect-MsolService
```

- Create the service account. Replace the "UserPrincipalName" and "Password" parameters to match your situation and preferences:

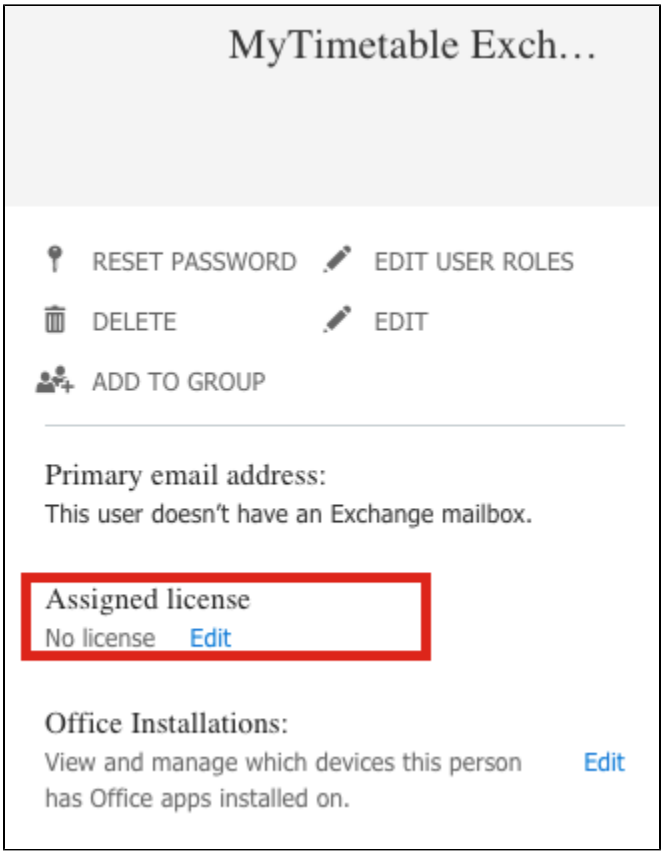
```
PS C:\> New-MsolUser -DisplayName "MyTimetable" -Name "MyTimetable Exchange Service Account 1" -
UserPrincipalName "sa-mytt-exch-1@eveoh.onmicrosoft.com" -Password "xxx" -PasswordNeverExpires $true -
StrongPasswordRequired $true
```

Assigning an Exchange Online license

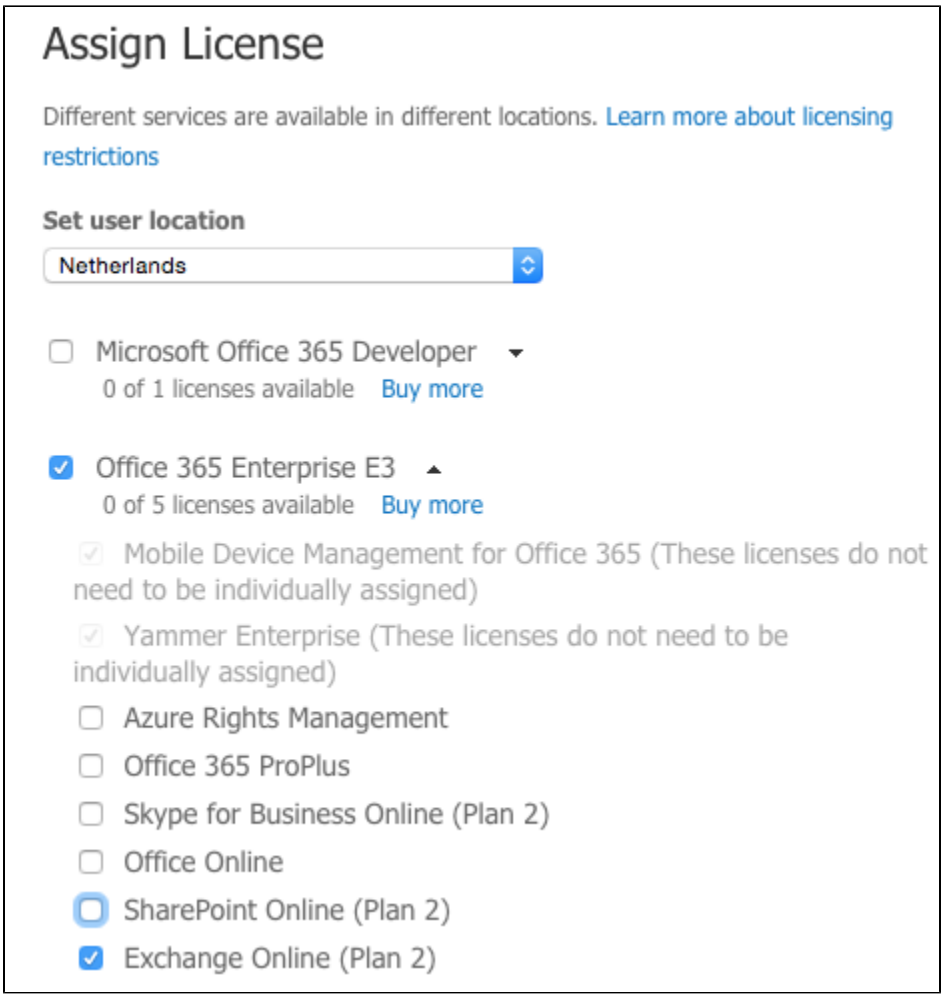
The service account needs to have a Exchange Online license assigned. After assigning a license, the service account will have a mailbox.

Using the Office 365 Admin portal

- Open the Office 365 Admin portal.
- Click "Users" -> "Active Users".
- Click the service account you have just created.
- In the right bar, find "Assigned license" and click "Edit".



- Click the license you would like to assign. Make sure "Exchange Online" is checked.



Using Powershell

- Open the Windows Azure Active Directory Powershell prompt
- Connect to Microsoft Online Services using your tenant admin account:

```
PS C:\> Connect-MsolService
```

- List your Office 365 plans. Pick the AccountSkuId you would like to use.

```
PS C:\> Get-MsolAccountSku
AccountSkuId                 ActiveUnits     WarningUnits    ConsumedUnits
------------                 -----------     ------------    -------------
Eveoh:DEVELOPERPACK          1               0               1
Eveoh:ENTERPRISEPACK         5               0               2
```

- List all service plans that are included in your Office 365 plan. In this case, the EXCHANGE_S_ENTERPRISE service plan refers to Exchange Online.

```
PS C:\> Get-MsolAccountSku | Where-Object {$_.AccountSkuId -eq "Eveoh:ENTERPRISEPACK"} | ForEach-Object {$_.
ServiceStatus}
ServicePlan                            ProvisioningStatus
-----------                            ------------------
INTUNE_O365                            PendingActivation
YAMMER_ENTERPRISE                      PendingInput
RMS_S_ENTERPRISE                       Success
OFFICESUBSCRIPTION                     Success
MCOSTANDARD                            Success
SHAREPOINTWAC                          Success
SHAREPOINTENTERPRISE                   Success
EXCHANGE_S_ENTERPRISE                  Success
```

- We will now assign an Office 365 license with only the Exchange Online service plan selected. Since we can only assign a plan and all service plans disabled, we first create a object reference that holds all disabled service plans. After that, we assign the license to the service account.

```
PS C:\> $O365Licences = New-MsolLicenseOptions -AccountSkuId Eveoh:ENTERPRISEPACK -DisabledPlans INTUNE_O365,
YAMMER_ENTERPRISE, RMS_S_ENTERPRISE, OFFICESUBSCRIPTION, MCOSTANDARD, SHAREPOINTWAC, SHAREPOINTENTERPRISE
PS C:\> Set-MsolUserLicense -UserPrincipalName "sa-mytt-exch-1@eveoh.onmicrosoft.com" -AddLicenses "Eveoh:
ENTERPRISEPACK" -LicenseOptions $O365Licences
```

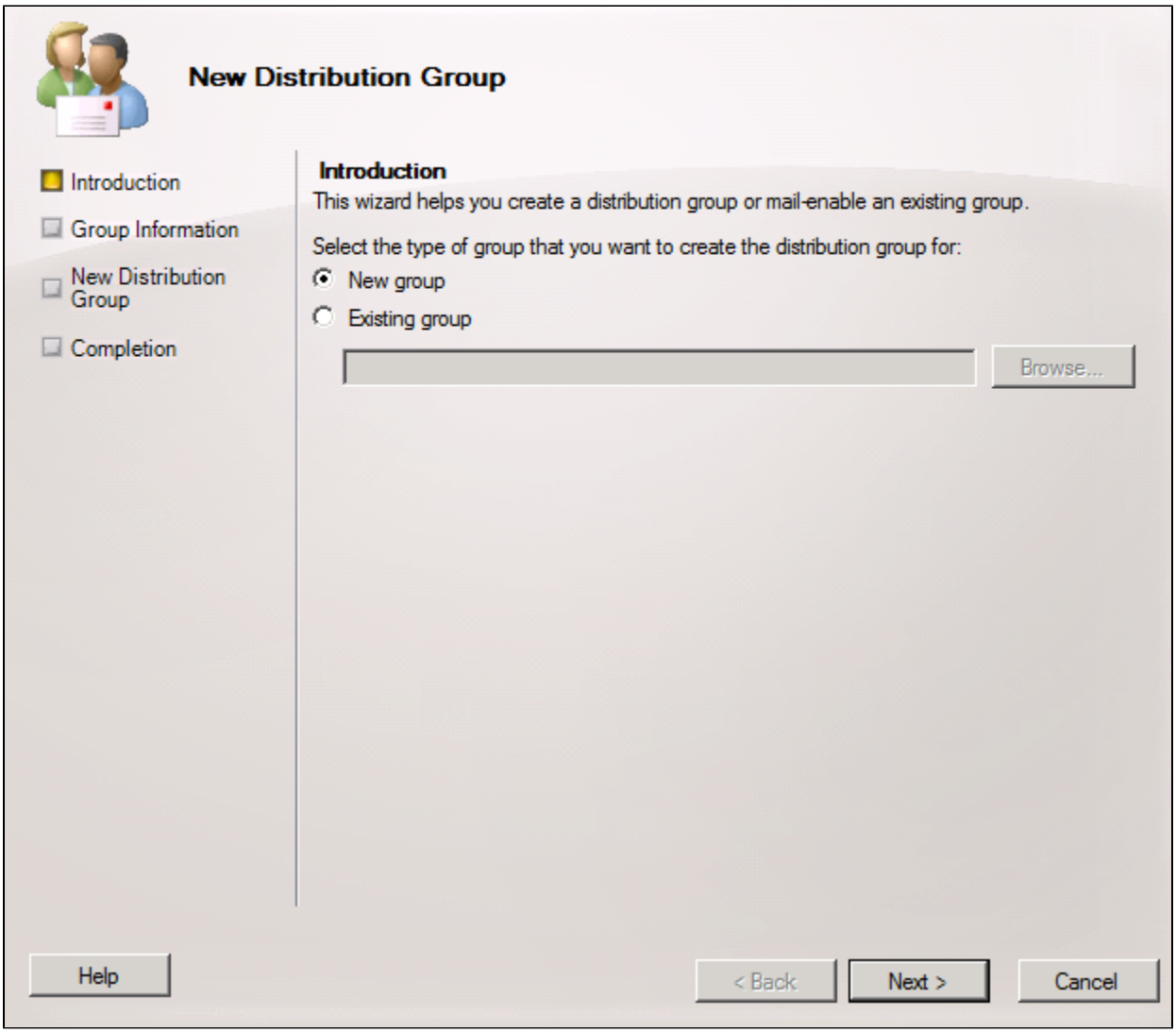
Creating a mail-enabled universal security group

It is recommended to create a mail-enabled universal security group containing the previously created service account. Microsoft throttles the number of requests allowed to Exchange Web Services on a per account basis. By using multiple service accounts, we are able to increase the number of requests to EWS. In the next step, we will delegate calendar permissions to the security group, instead of delegating permissions to the separate service accounts.

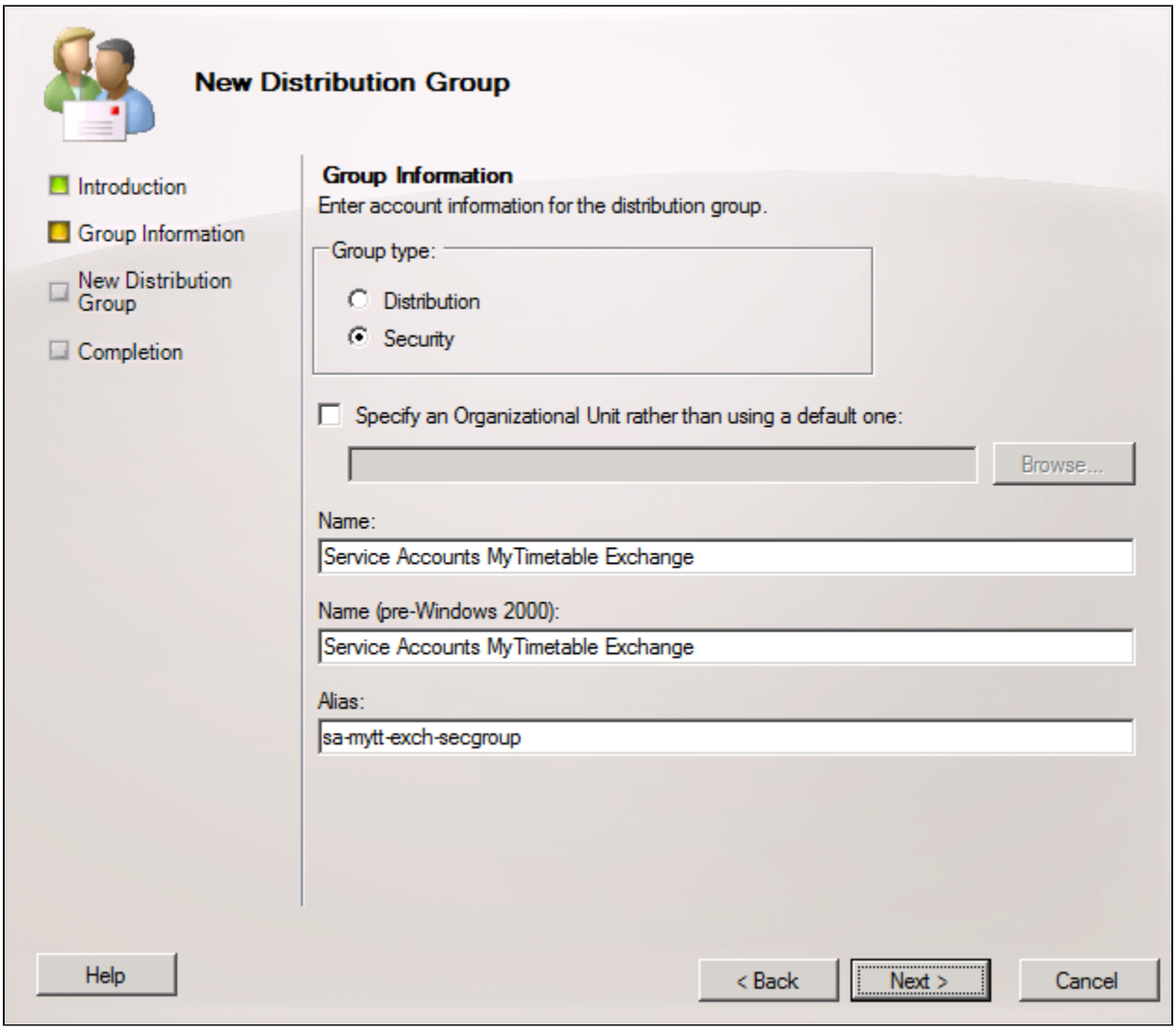
On-premises Exchange

Using the Exchange Management Console

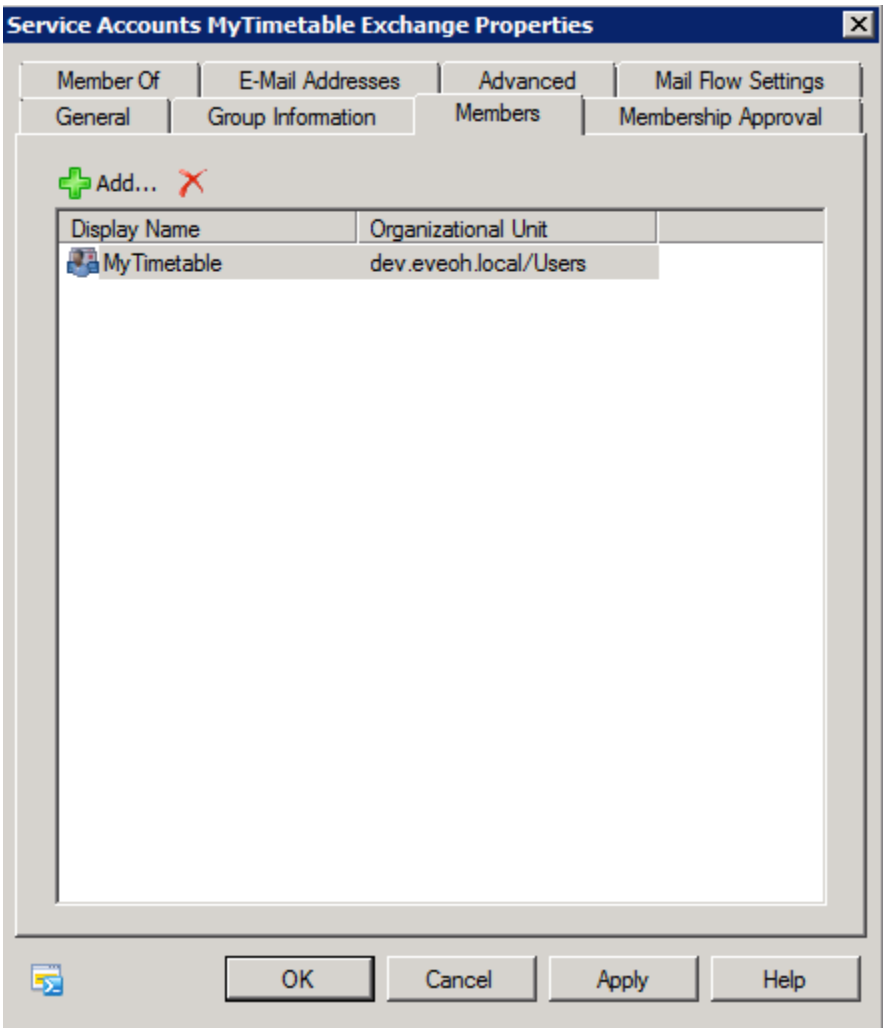
- Create a new Distribution Group.
- Select "New Group":



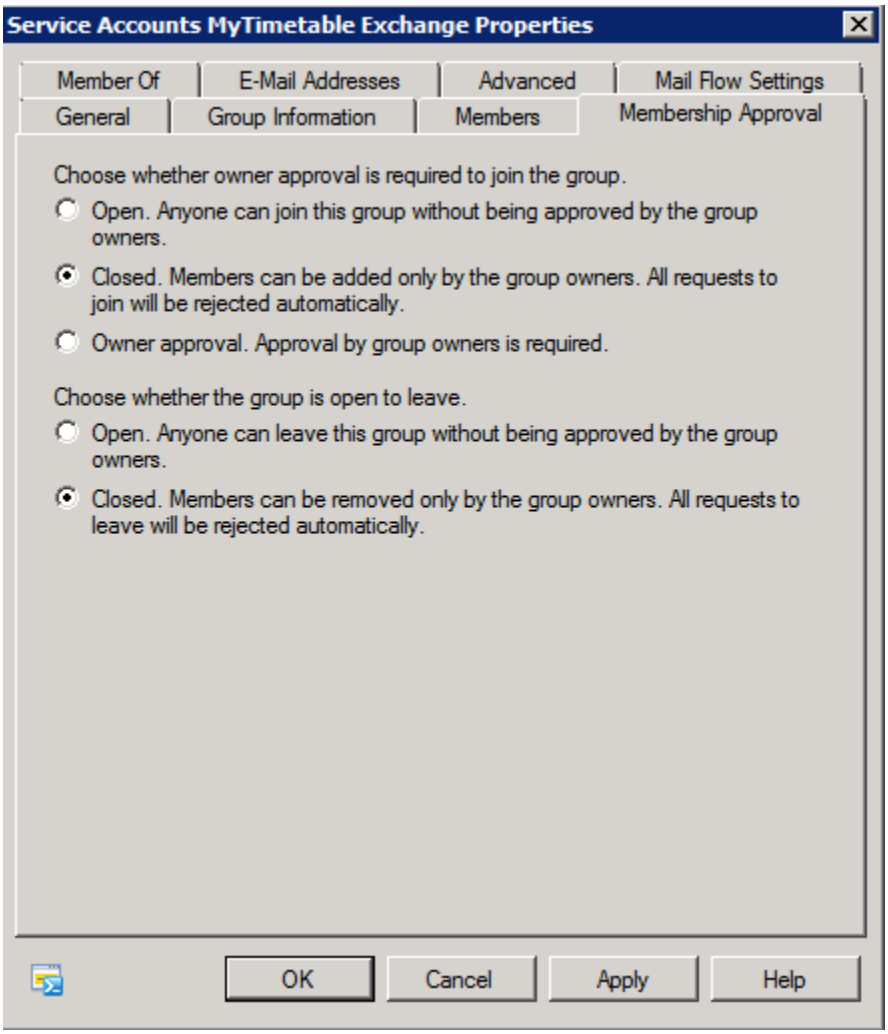
- Select "Security" as group type and enter a name and alias (e.g. sa-mytt-exch-secgroup):



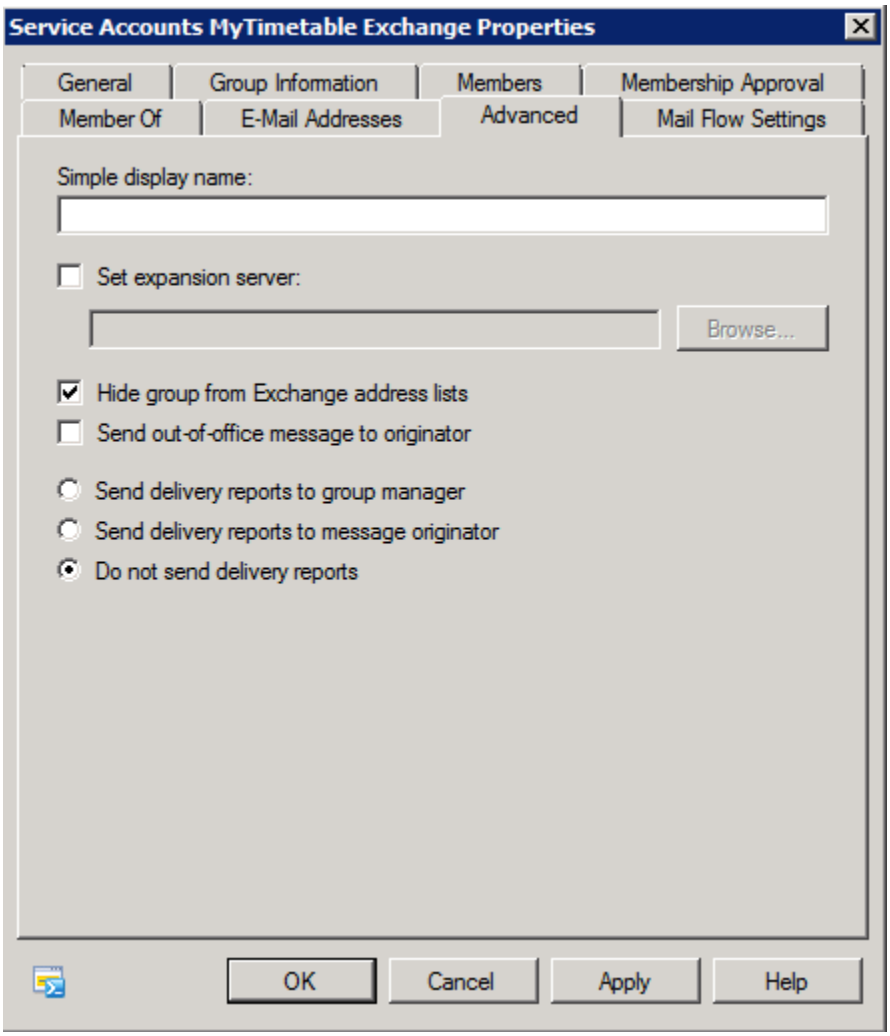
- Click Next.
- Click New.
- Click Finish.
- Open the properties of the newly created distribution group.
- Add the service account created in the previous step as a member:



- Make sure Membership Approval is set to "Closed" for both options:



- Optionally, hide the distribution group from the Exchange address lists:



Using Powershell

- Open the Exchange Management Shell.
- Create a new mail-enabled universal security group using the New-DistributionGroup cmdlet. Replace the parameters to match your situation and preferences:

```
PS C:\> New-DistributionGroup -Name "Service Accounts MyTimetable Exchange" -Type Security -PrimarySmtpAddress
"sa-mytt-exch-secgroup@dev.eveoh.local" -DisplayName "MyTimetable" -MemberDepartRestriction Closed -
MemberJoinRestriction Closed
Name                          DisplayName                 GroupType                   PrimarySmtpAddress
----                          -----------                 ---------                   ------------------
Service Accounts MyTimetab... MyTimetable                 Universal, SecurityEnabled  sa-mytt-exch-
secgroup@dev....
```

- As an optional step, you can hide the security group from the global address list.

```
PS C:\> Set-DistributionGroup -Identity "sa-mytt-exch-secgroup@dev.eveoh.local" -HiddenFromAddressListsEnabled
$true
```

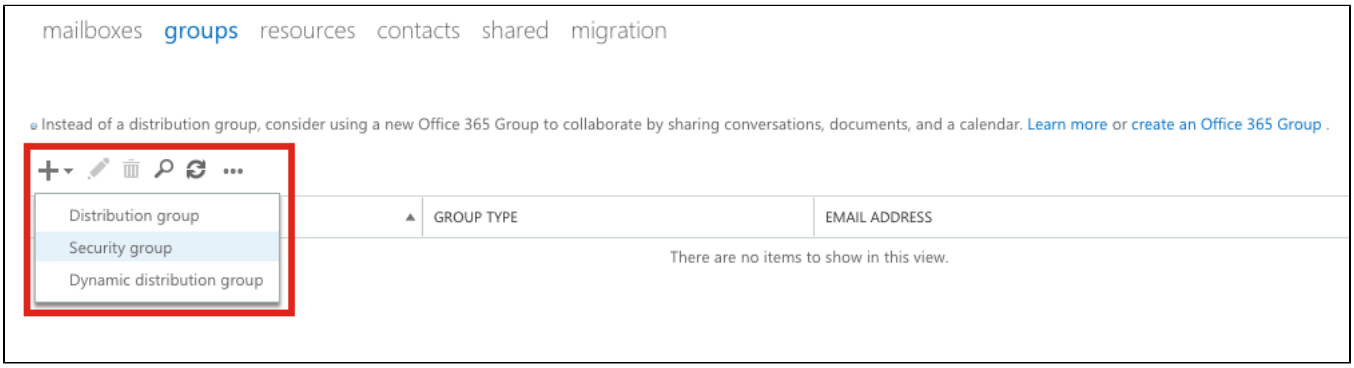
- Finally, add the service account to the security group:

```
PS C:\> Add-DistributionGroupMember -Identity "sa-mytt-exch-secgroup@dev.eveoh.local" -Member "sa-mytt-exch-
1@dev.eveoh.local"
```

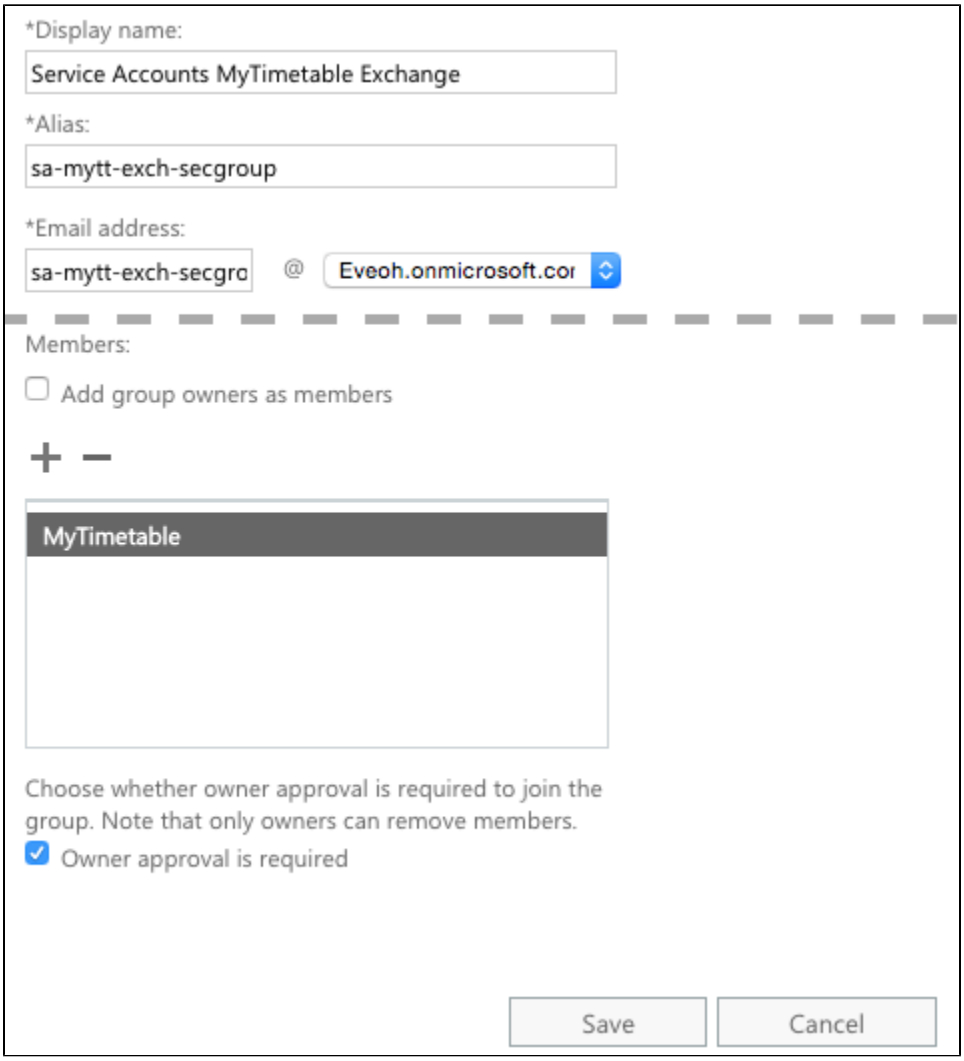
Office 365

Using the Office 365 Admin portal

- Open the Office 365 Admin portal.
- Click "Groups" in the left menu bar.
- Click the "Set up distribution lists and other Exchange groups in the Exchange admin center." link, which will redirect you to the Exchange admin center.
- Click the + sign and pick "Security group":



- Enter a display name and an alias (e.g. "sa-mytt-exch-secgroup"), uncheck "Add group owners as members", add the service account created in the previous step as member:



- Optionally, open the newly created security group properties and check "Hide this group from address lists".

Using Powershell

- Connect Powershell to Office 365.
- Create a new mail-enabled universal security group using the New-DistributionGroup cmdlet. Replace the parameters to match your situation and preferences:

```
PS C:\> New-DistributionGroup -Name "Service Accounts MyTimetable Exchange" -Type Security -PrimarySmtpAddress
"sa-mytt-exch-secgroup@eveoh.onmicrosoft.com" -DisplayName "MyTimetable" -MemberDepartRestriction Closed -
MemberJoinRestriction Closed
Name                           DisplayName                 GroupType                    PrimarySmtpAddress
----                           -----------                 ---------                    ------------------
Service Accounts MyTimetab... MyTimetable                  Universal, SecurityEnabled    sa-mytt-exch-
secgroup@eveo...
```

- As an optional step, you can hide the security group from the global address list.

```
PS C:\> Set-DistributionGroup -Identity "sa-mytt-exch-secgroup@eveoh.onmicrosoft.com" -
HiddenFromAddressListsEnabled $true
```

- Finally, add the service account to the security group:

```
PS C:\> Add-DistributionGroupMember -Identity "sa-mytt-exch-secgroup@eveoh.onmicrosoft.com" -Member "sa-mytt-
exch-1@eveoh.onmicrosoft.com"
```

Delegating calendar permissions to the security group

Finally, we need to give the security group containing the service account(s) delegated calendar permissions on the mailboxes of the users. We assume that all users that are allowed to use the calendar integration are member of an existing security group.

Exchange on-premises

- Open the Exchange Management Shell.
- Import the ActiveDirectory module, when necessary.
- Select all mailboxes to set the delegation permissions on. We assume that these accounts are grouped in a security group. In the following example, all users are in the security group "staff".

Get mailboxes by OU

```
PS C:\> $mailboxes = Get-ADGroupMember -Identity staff | Get-ADUser | ForEach-Object {Get-Mailbox -Identity $_.
UserPrincipalName -errorAction silentlyContinue}
```

- Finally, allow Author rights for the service account security group to all selected mailboxes. On line 1, we set the security group created in a previous step. Then we loop through all mailboxes we have retrieved in the previous step. For each mailbox, we get the path to the calendar folder (line 4). We have to explicitly retrieve this name, since the calendar folder name is localised. We then check if permissions have already been set (line 5). If not, we add Author permissions (line 8). If already set, we update the permissions (line 12).

```
PS C:\> $secgroup = "sa-mytt-exch-secgroup@dev.eveoh.local"
foreach ($m in $mailboxes)
{
    $path = ($m | Select-Object -ExpandProperty PrimarySmtpAddress).ToString() + ":\" + (Get-
MailboxFolderStatistics $m.UserPrincipalName | Where-Object { $_.Foldertype -eq "Calendar" } | Select-Object -
First 1).Name
    $permissions = @(Get-MailboxFolderPermission -Identity $path -User $secgroup -ErrorAction SilentlyContinue).
count
    if ($permissions -eq 0) {
        # not in ACL, add permission
        Add-MailboxFolderPermission -Identity $path -User $secgroup -AccessRights Author
    }
    else {
        # user is already in ACL, change permission
        Set-MailboxFolderPermission -Identity $path -User $secgroup -AccessRights Author
    }
}
```

Office 365

- Connect Powershell to Office 365.
- Select all accounts to set the delegation permissions on. We assume that these accounts are grouped in a security group. In the following example, all users are in the security group "Staff".

Get all mailboxes

```
PS C:\> Get-MsolGroup | Where-Object {$_.DisplayName -eq "Staff"}
ObjectId                             DisplayName           GroupType                 Description
--------                             -----------           ---------                 -----------
64731c32-f1df-4b92-8dbe-1809c23ff85b Staff                 Security
```

- Get all mailboxes of the selected security group:

Get mailboxes by OU

```
PS C:\> $mailboxes = Get-MsolGroupMember -GroupObjectId 64731c32-f1df-4b92-8dbe-1809c23ff85b | Get-MsolUser |
ForEach-Object {Get-Mailbox -Identity $_.UserPrincipalName -errorAction silentlyContinue}
```

- Finally, allow Author rights for the service account security group to all selected mailboxes. On line 1, we set the security group created in a previous step. Then we loop through all mailboxes we have retrieved in the previous step. For each mailbox, we get the path to the calendar folder (line 4). We have to explicitly retrieve this name, since the calendar folder name is localised. We then check if permissions have already been set (line 5). If not, we add Author permissions (line 8). If already set, we update the permissions (line 12).

```
PS C:\> $secgroup = "sa-mytt-exch-secgroup@eveoh.onmicrosoft.com"
foreach ($m in $mailboxes)
{
    $path = $m.PrimarySmtpAddress + ":\" + (Get-MailboxFolderStatistics $m.PrimarySmtpAddress | Where-Object {
$_.Foldertype -eq "Calendar" } | Select-Object -First 1).Name
    $permissions = @(Get-MailboxFolderPermission -Identity $path -User $secgroup -ErrorAction SilentlyContinue).
count
    if ($permissions -eq 0) {
        # not in ACL, add permission
        Add-MailboxFolderPermission -Identity $path -User $secgroup -AccessRights Author
    }
    else {
        # user is already in ACL, change permission
        Set-MailboxFolderPermission -Identity $path -User $secgroup -AccessRights Author
    }
}
```