

Microsoft Connector Setup for Admins

MyWhistleBox by default will send out requests emails (Upload, Signature etc.) using it's own SMTP mail server. This has the side effect of emails "coming from" mywhistlebox.com and not the customer's email account. Using the MyWhistleBox email settings, the default can be overridden to use an email server of the customers choice. Using the Microsoft connector, you can connect to Microsoft/Office Online via OAuth2.

This article will cover the required steps necessary for an administrator to configure the connector for outbound email messages.

Prerequisites

- You must have a MyWhistleBox Professional level plan.
- You must have an active Microsoft Azure account to configure the Microsoft connector for your instance.

Before We Start

Before users can connect MyWhistleBox to their Microsoft accounts, an administrative user must first create and register a new application in the Microsoft Azure Portal. The following sections explain how the administrator can acquire the proper API credentials from Microsoft Azure and use those credentials to set up the connector in MyWhistleBox.

Heads up. While completing these steps, you will need to gather the following items. We will point them out as we go along so you can copy them down.

- **Tenant Id**
- **Client Id**
- **Client Secret**
- **Object Id**
- **Application Id**

It may be helpful to open a notepad and copy and paste the values as you gather them.

Create an Application in Microsoft Azure Portal

To set up the MyWhistleBox Microsoft Connector, you first need to create a new "application" and obtain the necessary values used in the connector configuration (i.e. **Tenant ID**, **Client ID**, **Client Secret**). Just follow these steps.

1. Navigate to the [Microsoft Azure portal](https://portal.azure.com) (portal.azure.com) in your web browser.
2. Sign in using the Microsoft account under which you would like to register the MyWhistleBox application.

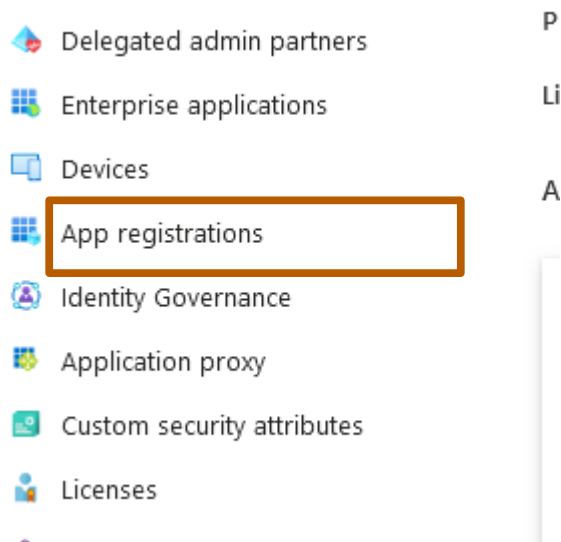
3. On the Microsoft Azure homepage, click the **Microsoft Entra Id icon**.

Azure services

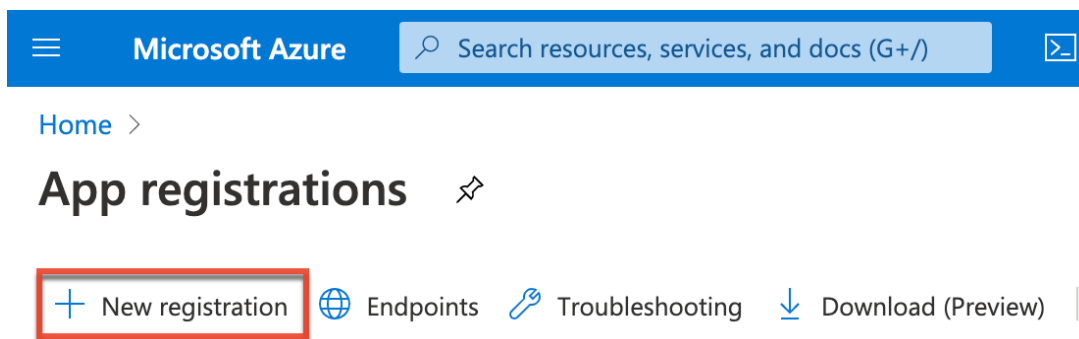


4. Create a new app registration.

a. Click **App Registrations** from the side bar menu



b. On the App registrations page, click the "+ **New registration**" button.



c. On the “**Register an Application**” page, complete the following fields:

- i. **Name:** Enter an application name of your choice (e.g. MyWhistleBox).
- ii. **Supported account types:** “Accounts in this organizational directory only (Your Company only - Single tenant)”

iii. Leave **Redirect URI** empty

It should look something like so:

Register an application ...

* Name

The user-facing display name for this application (this can be changed later).

MyWhistleBox

Supported account types

Who can use this application or access this API?

- ☒ Accounts in this organizational directory only (Default Directory only - Single tenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- ☐ Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Select a platform



e.g. https://example.com/auth

d. Click "**Register**"

e. Note the “**Application (client) ID**” and “**Directory (tenant) ID**”, ignore **Object ID** (the real one will be obtained further down).

^ Essentials

Display name : [MyWhistleBox](#)

Application (client) ID : 1f912a50-8df5-48da-986a-c9ce7098ca17

~~Object ID : 16aac2e6-fc7b-4713-816b-ed4ea98bf234~~

Directory (tenant) ID : c4b943b9-e3b0-4b5d-97f1-aa8e3e4cc431

Supported account types : [My organization only](#)

5. Add a **Client Secret**.

a. Go to sidebar “**Certificates & secrets**”



b. Click on “+ **New client secret**”.

c. Provide a description and select an expiration. **Important:** Microsoft is now restricting expiration periods to a maximum of 2 years. We suggest you select 2 years. When the Client Secret expires, you will need to come back to this section and generate a new one. The section at the bottom of this document explains how to update the Client Secret.

Certificates (0) Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

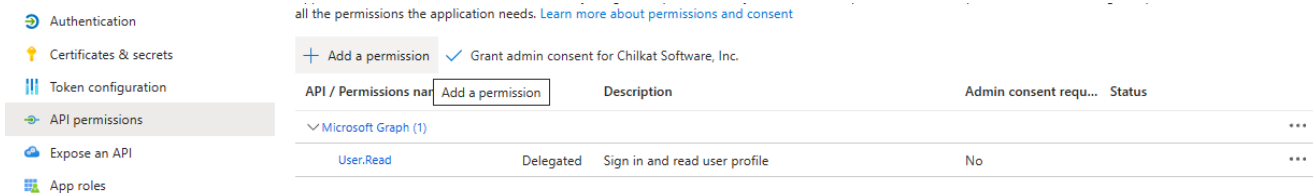
Description	Expires	Value ①	Secret ID
Password uploaded on Sun Jan 07 2024	7/5/2024	GWL*****	e69ffe47-c8c0-4020-b67b-029e4ebdb16  

d. Copy the **Secret Value** (e.g. GWL....) to the clipboard and save. This is your **Client Secret**. You won’t have a chance to copy the secret Value again, so you must get it now.

6. Add permissions.

a. On Sidebar, Click “**API permissions**”

b. Click “+**Add a permission**” in the content pane

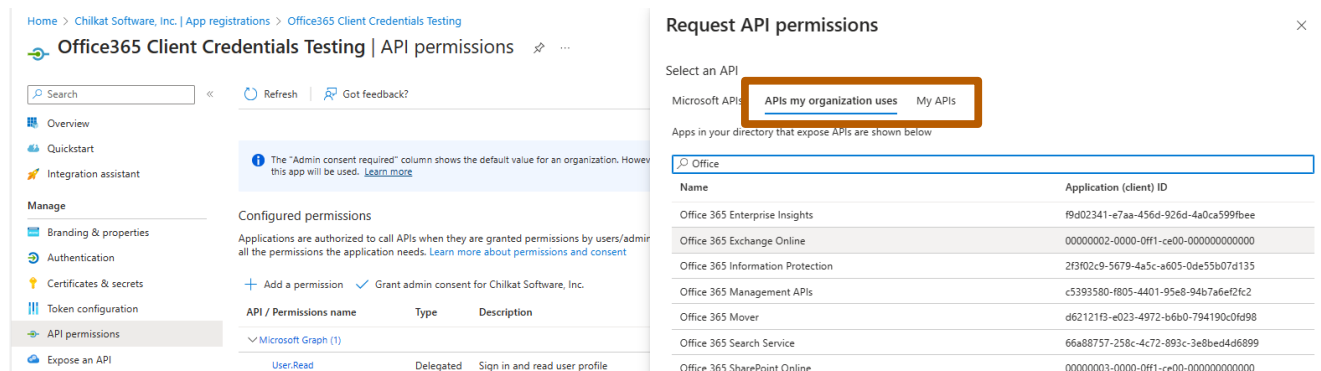


all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ☒ Grant admin consent for Chilkat Software, Inc.

API / Permissions name	Add a permission	Description	Admin consent required	Status
▼ Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	...

c. Select the “**APIs my organization uses**” tab, then type “**Office**” in the search bar, and select “**Office 365 Exchange Online**” to display the appropriate permissions.



Home > Chilkat Software, Inc. | App registrations > Office365 Client Credentials Testing

Office365 Client Credentials Testing | API permissions

Search Refresh Got feedback?

Overview Quickstart Integration assistant

Manage Branding & properties Authentication Certificates & secrets Token configuration **API permissions** Expose an API

The “Admin consent required” column shows the default value for an organization. However, this app will be used. [Learn more](#)

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins. All the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ☒ Grant admin consent for Chilkat Software, Inc.

API / Permissions name	Type	Description
▼ Microsoft Graph (1)		
User.Read	Delegated	Sign in and read user profile

Request API permissions

Select an API

Microsoft APIs **APIs my organization uses** My APIs

Apps in your directory that expose APIs are shown below


Office

Name	Application (client) ID
Office 365 Enterprise Insights	f9d02341-e7aa-456d-926d-4a0ca599fbee
Office 365 Exchange Online	00000002-0000-0ff1-ce00-000000000000
Office 365 Information Protection	2f302c9-5679-4a5c-a605-0de55b07d135
Office 365 Management APIs	c5393580-f805-4401-95e8-94b7a6e2fc2
Office 365 Mover	d62121f3-e023-4972-b6b0-794190c0fd98
Office 365 Search Service	66a88757-258c-4c72-893c-3e8bed4d6899
Office 365 SharePoint Online	00000003-0000-0ff1-ce00-000000000000

d. Click “Application Permissions”

Request API permissions

< All APIs



Office 365 Exchange Online
https://outlook.office.com

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Application permissions Your application

e. Select “SMTP.SendAsApp” from the list, then click “Add permissions”

SMTP (1)

☒

SMTP.SendAsApp ⓘ
Application access for sending emails via SMTP AUTH

Yes

> Tasks

> User

Add permissions

Discard

f. Click on “Grant admin consent for <your Azure organization’s name>”

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions shows all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission

✓ Grant admin consent for ██████████, Inc.

API / Permissions name	Type	Description	Grant admin consent for ██████████, Inc.	Consent requ...	Status
Microsoft Graph (1)					
User.Read	Delegated	Sign in and read user profile		No	
Office 365 Exchange Online (3)					
IMAP.AccessAsApp	Application	IMAP.AccessAsApp	Yes		⚠ Not granted for
POP.AccessAsApp	Application	POP.AccessAsApp	Yes		⚠ Not granted for
SMTP.SendAsApp	Application	Application access for sending emails via SMTP AUTH	Yes		⚠ Not granted for

Now you will see something like the following.

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions shows all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for [REDACTED]

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for i
▼ Office 365 Exchange Online (3)				
IMAP.AccessAsApp	Application	IMAP.AccessAsApp	Yes	✓ Granted for i
POP.AccessAsApp	Application	POP.AccessAsApp	Yes	✓ Granted for i
SMTP.SendAsApp	Application	Application access for sending emails via SMTP AUTH	Yes	✓ Granted for i

7. Before we leave the Portal, let's retrieve the real **Object ID**.

- Click the “**Microsoft Azure**” logo and select the **Microsoft Entra Id** icon.
- Click “**Enterprise Applications**” from the sidebar.

Overview

- Overview
- Diagnose and solve problems

Manage

- All applications
- Private Network connectors
- User settings
- App launchers

View, filter, and search applications in your organization that are set up to use your Microsoft Entra tenant as their Identity Provider.

The list of applications that are maintained by your organization are in [application registrations](#).

Search by application name or object ID Application type == Enterprise Applications Application ID starts with Add filters

1 application found

Name	Object ID	Application ID	Homepage URL	Created on	Certi
MY MyWhistleBox	8c271677-85a2-480e-9f83-27a...	1f912a50-8df5-48da-986a-c9ce...		1/7/2024	-

- Click the App Name (e.g. MyWhistleBox) to display some Id's.

Properties

MY

Name ⓘ

MyWhistleBox

Application ID ⓘ

1f912a50-8df5-48da-986a-c ...

Object ID ⓘ

8c271677-85a2-480e-9f83-2 ...

- Copy the **Object ID**. You will need this for the Exchange Online Powershell Command in the next section.

Configure Exchange Online

The final step is to configure the Exchange Online settings. Unfortunately, Microsoft requires you to do this via Powershell and not the portal. Please have your **Tenant Id, Application ID and Object ID** handy from the previous sections.

From Windows, start a PowerShell command prompt **“As Admin”**.

Begin by running the following commands in your Powershell window. Insert your **Tenant Id** in the 3rd command (no brackets).

```
Install-Module -Name ExchangeOnlineManagement -allowprerelease
Import-module ExchangeOnlineManagement
Connect-ExchangeOnline -Organization <tenantId>
```

Next, you'll need to create a new service principle using your **Application (client) ID**, and the **Object ID** (from the final step above).

Run the following Powershell command using your **Application ID** and **Object ID** (no brackets).

```
New-ServicePrincipal -AppId <your app id> -ObjectId <your object id>
```

You can verify your newly created service principle by running the Powershell command:

Get-ServicePrincipal | fl

Finally, add FullAccess mailbox permissions for the sending email address to be accessed via SMTP. The value for -User is your **Object ID**.

```
Add-MailboxPermission -Identity "<your sending email address>" -User <your object id> -AccessRights
FullAccess
```

Configuring MyWhistleBox Email Settings

Once you have completed configuring Microsoft 365, you can configure MyWhistleBox's email settings using MyWhistleBox's Email Settings Page. You will need your **Directory (Tenant) Id**, **Application (Client) Id** and **Client Secret**. No authorization is required using this setup. After saving the credentials on the MyWhistleBox's Email Settings Page, you should be able to successfully send a Test Email.

Note: The “From Address” field in the MyWhistleBox Connector Setup should be the same email address you used in the **Add-MailboxPermission** powershell command.

When Client Secret Expires

Microsoft is now limiting Client Secrets to a maximum of 2 years. What does this mean for your email connection? When the Client Secret expires, no emails will be allowed to be sent through the 365 account until a new Client Secret has been created. Fortunately, this is a simple procedure to correct.

1. Go to step 5 above and create a new Client Secret with a new 2 year expiration.
2. Next, you will need to go to your MyWhistleBox Email Settings and update the Client Secret with the new one. Don't forget to save the new value and test the email connection.

We suggest you create a calendar reminder a week before your expiration date reminding you to create a new Client Secret. This will help prevent service interruption.