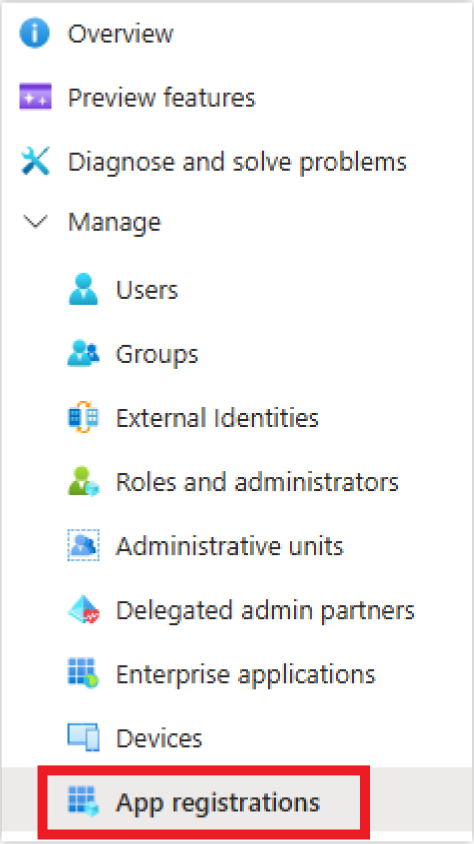
How to create a custom private client app in Azure

A custom private client app can be used for PD to sync user information, as well as sending emails (reports, user PINs, etc) and receiving emails (email-to-print).

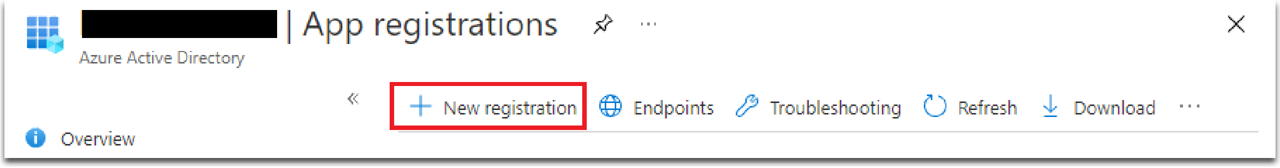
The advantage of using a custom private client app over a user delegated app is that the 'client secret' can be valid for up to 2 years. A user delegated token can only last a maximum of 90 days before a new one must be created.

Register an Application

1. Log in to portal.azure.com with an account that has sufficient rights to register new applications.
2. Go to **Microsoft Entra ID**.
3. Click **App registrations**.

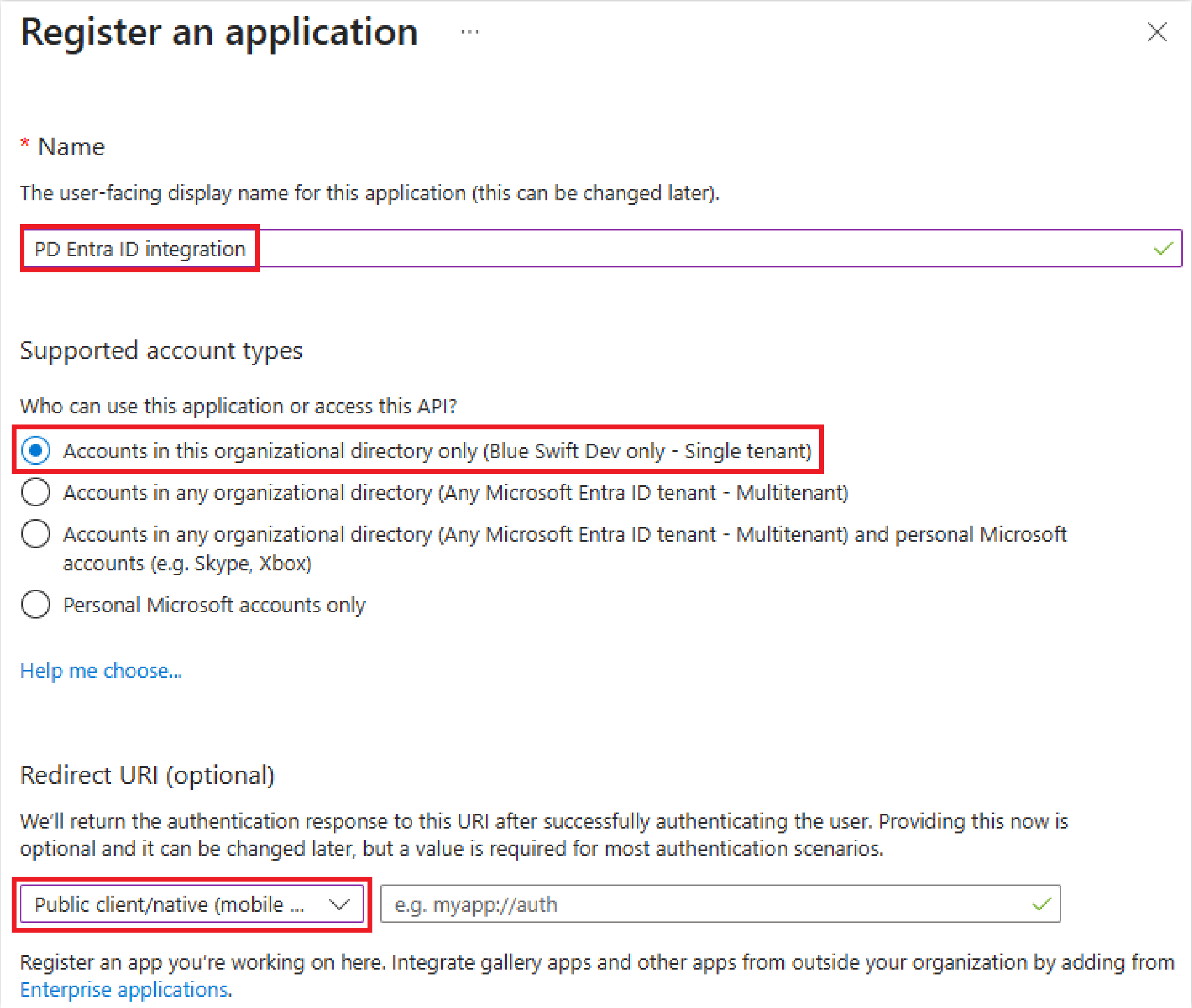


4. Click **New registration.**



5. Enter a name for your custom application (e.g. PD Entra ID integration).
6. Choose **Accounts in this organizational directory only (Single tenant).**

7. Change the Redirect URI to **Public client/native (mobile & desktop)**. Leave the URI field blank.



Register an application

* Name

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

PD Entra ID integration

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Blue Swift Dev 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.

Public client/native (mobile ... | e.g. myapp://auth

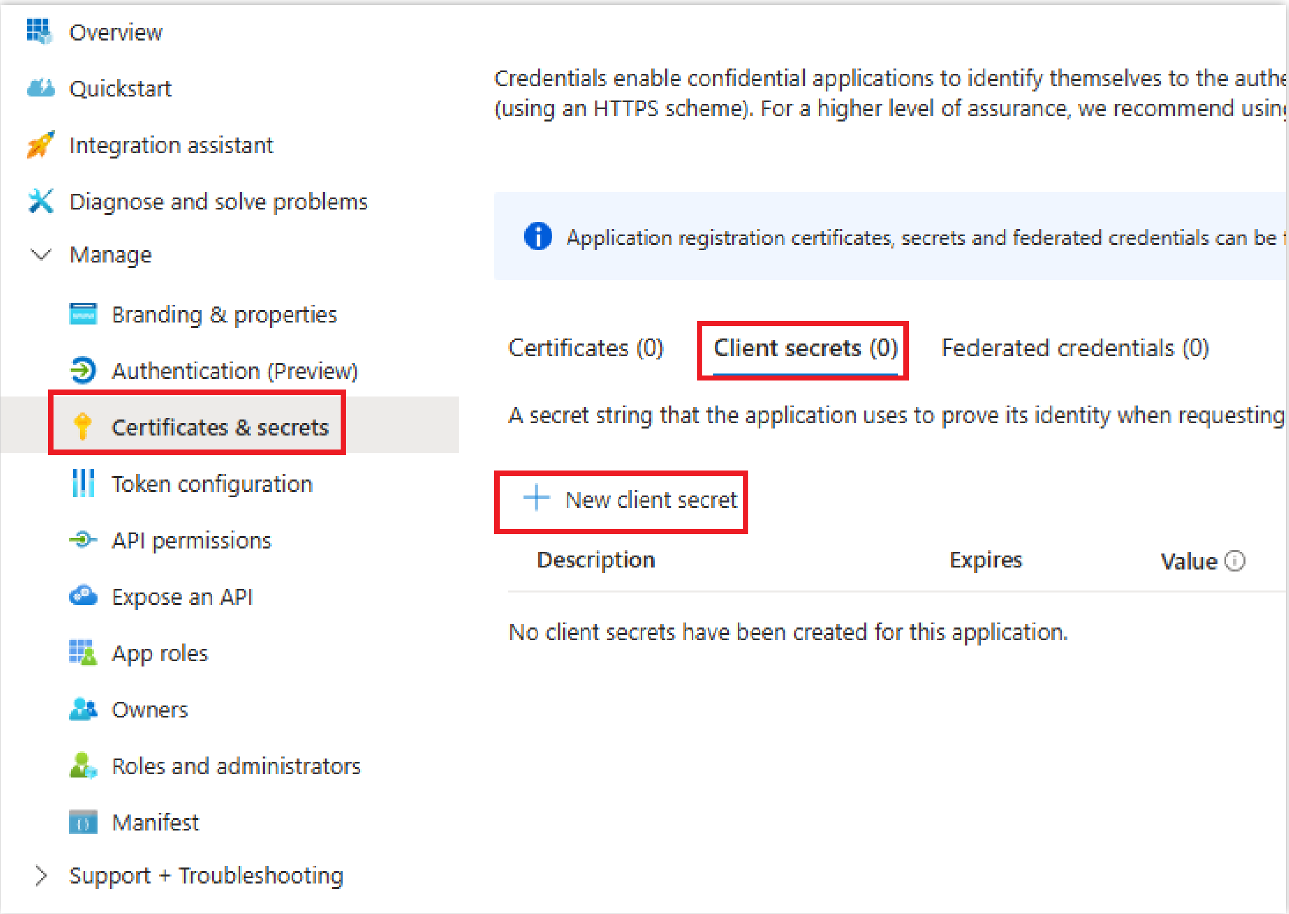
Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.

8. Click **Register.**

Create the application Client Secret

Now that we've registered a private client app, we must add a secret key that PD will use to prove it is allowed to use the app.

1. Click **Certificates & secrets.**
2. Click **New client secret.**



Credentials enable confidential applications to identify themselves to the authe
(using an HTTPS scheme). For a higher level of assurance, we recommend usin

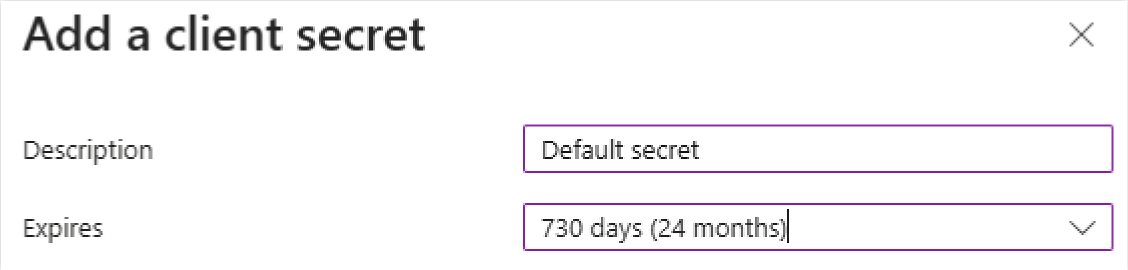
Application registration certificates, secrets and federated credentials can be

A secret string that the application uses to prove its identity when requesting

Description	Expires	Value ⓘ

No client secrets have been created for this application.

3. Enter a **Description** for the secret, e.g. Default secret
4. Choose the validity length from the **Expiry** dropdown. The maximum is 2 years. After expiry, a new secret must be created.



Add a client secret

Description: Default secret

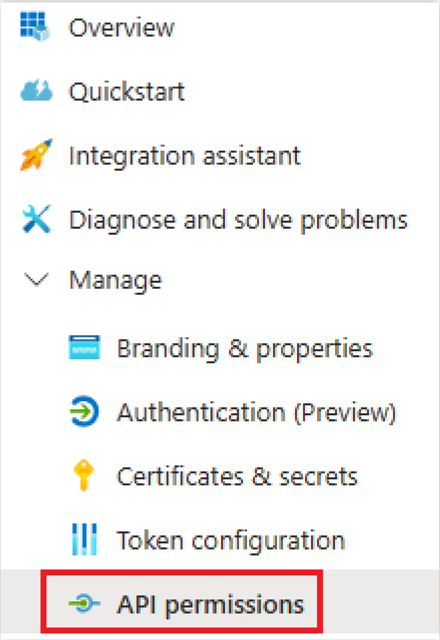
Expires: 730 days (24 months)

5. Click **Add**.
6. IMPORTANT: Once the secret is created, copy the **Value** (not the Secret ID) and save it somewhere. After a while, the Azure portal will hide this value, so make sure you have recorded it. This will be needed in the app configuration in PD.

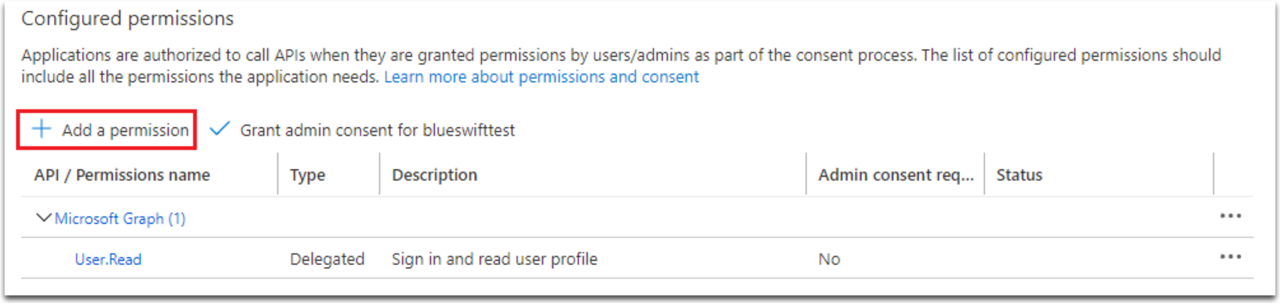
Permissions for Entra ID sync (user information sync)

Two API permissions are required for the Entra ID user information sync: **User.Read.All** and **Group.Read.All**. Both of these require admin consent.

1. Click **API permissions.**



2. Click **Add a permission.**



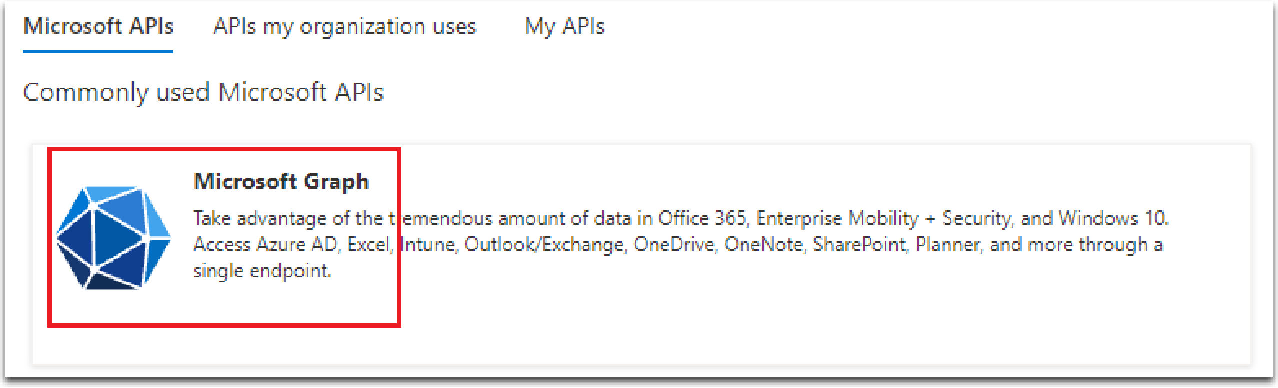
Configured permissions

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 should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission ✓ Grant admin consent for blueswifttest

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	

3. Click **Microsoft Graph.**

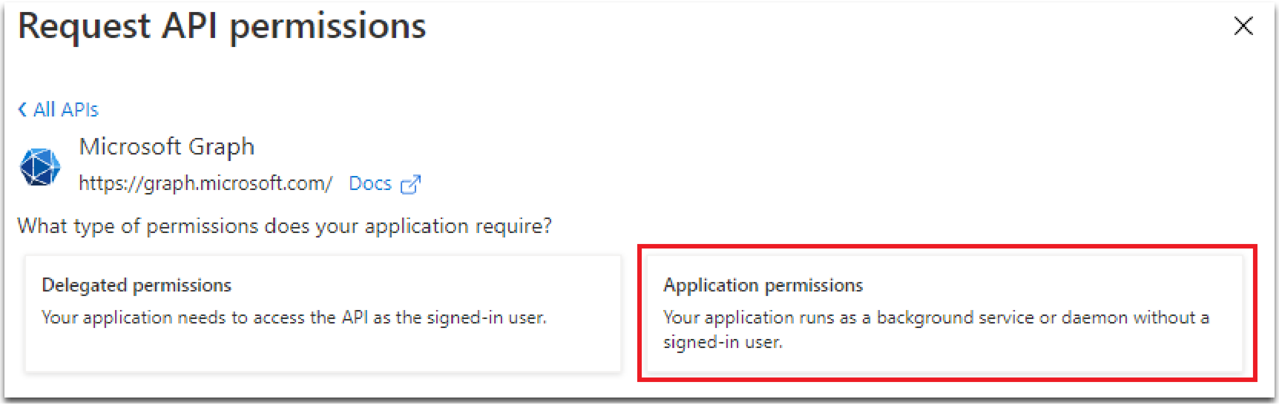


Commonly used Microsoft APIs

Microsoft Graph

Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

4. Click **Application permissions.**



Request API permissions

‹ All APIs

Microsoft Graph

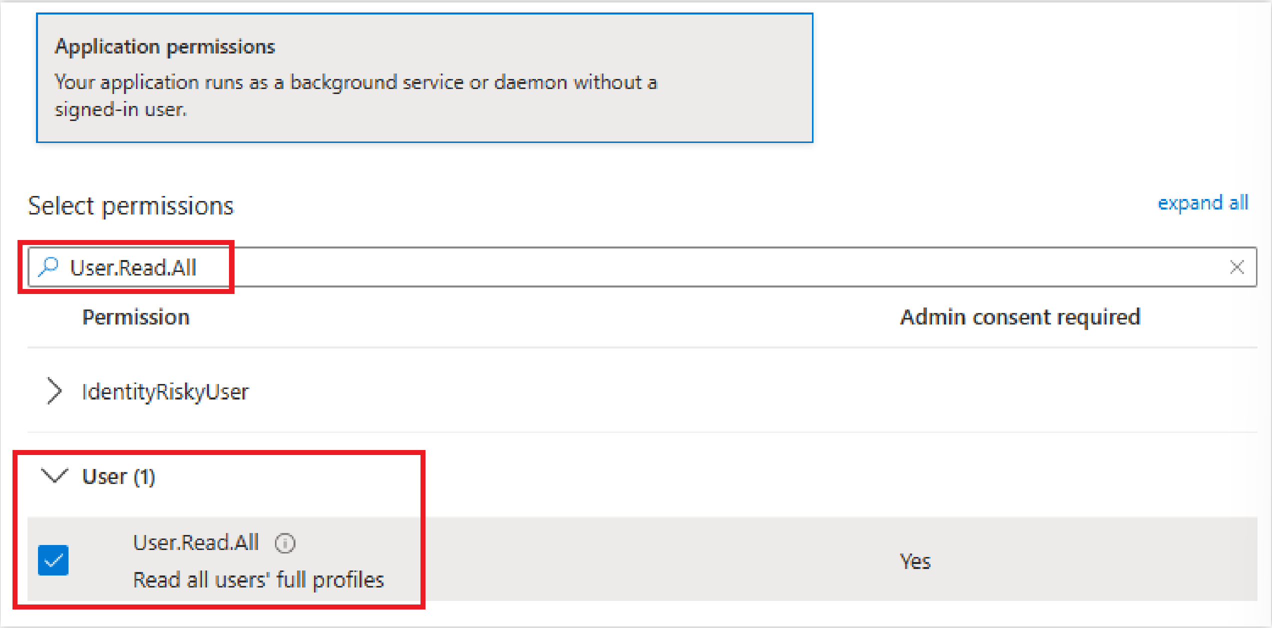
https://graph.microsoft.com/ Docs

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.

5. In the **Select permissions** search box, type **User.Read.All**. Click the down symbol next to **User** and then tick the **User.Read.All** checkbox.



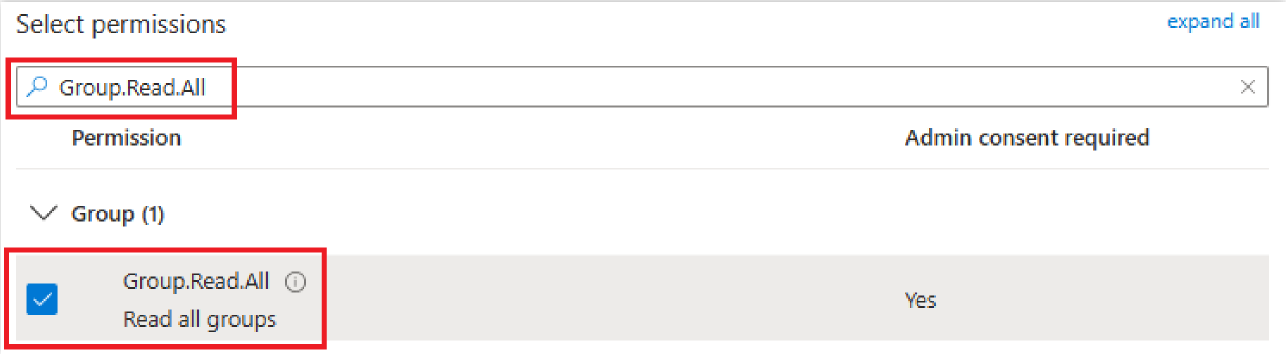
Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions

User.Read.All

	Permission	Admin consent required
	IdentityRiskyUser	
	User (1)	
☑	User.Read.All Read all users' full profiles	Yes

6. Now enter **Group.Read.All** and tick the **Group.Read.All** checkbox.

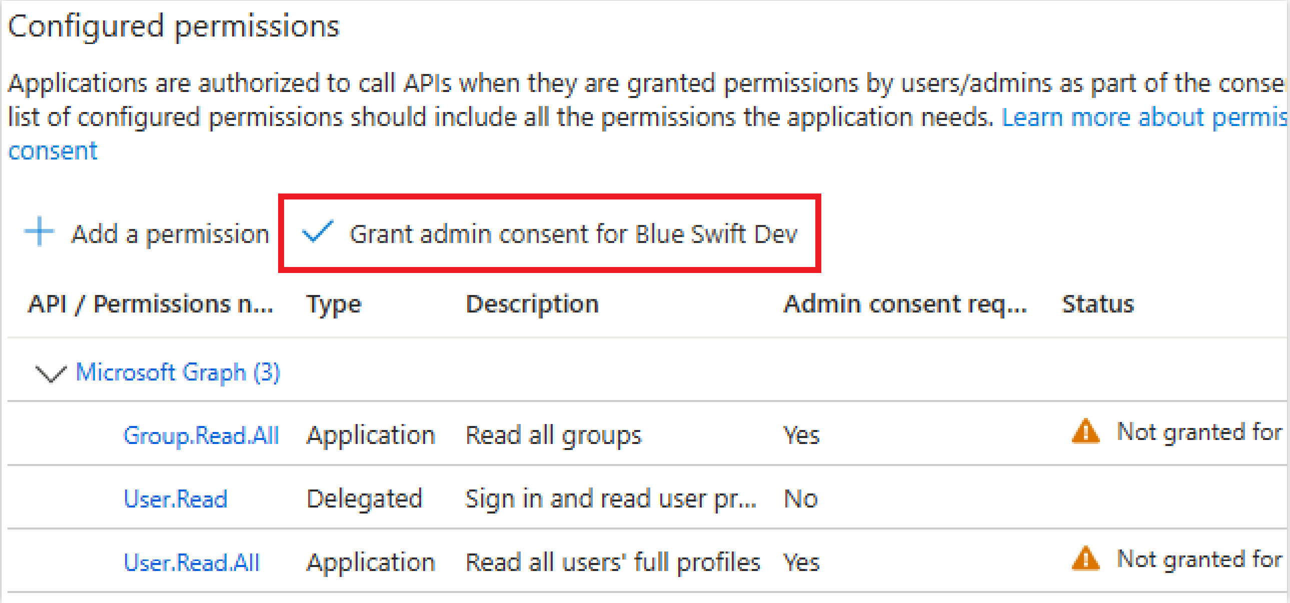


Select permissions

Group.Read.All

	Permission	Admin consent required
	Group (1)	
☑	Group.Read.All Read all groups	Yes

7. Click **Add permissions.**
8. Now we need to grant admin consent for permissions. Click the **Grant admin consent for tenantname** button.



Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the conse
list of configured permissions should include all the permissions the application needs. Learn more about permis
consent

API / Permissions n...	Type	Description	Admin consent req...	Status
Microsoft Graph (3)				
Group.Read.All	Application	Read all groups	Yes	⚠ Not granted for
User.Read	Delegated	Sign in and read user pr...	No	
User.Read.All	Application	Read all users' full profiles	Yes	⚠ Not granted for

9. Click **Yes** to confirm.

Permissions for email send/receive

For PD to send emails, the client app must have both the **Mail.Send** and **Mail.ReadWrite** permissions. The simplest way to allow this, is to add these permissions just like we did for reading user information. However, these permissions have tenant-wide scope. This means that the client app credentials could allow reading all users’ emails for the whole tenant, as well as sending emails as any user.

A more secure method is to only allow these permissions for the specific mailbox. This requires the configuration of Microsoft Exchange Online Role-Based Access Control (RBAC) for Applications.

Prerequisites for RBAC setup

- Exchange Administrator role (or Organization Management role group membership)
- PowerShell running as Administrator
- Exchange Online Management PowerShell module
- Application ID and Object ID from the Enterprise Applications page in Entra ID

Installing the Exchange Online Module in PowerShell

If you know that the Exchange Online Module has already been installed, skip to the next section.

- Open PowerShell as Administrator (right-click PowerShell and select "Run as administrator")
- Run the following command:

```
Install-Module -Name ExchangeOnlineManagement -Force
```

- Wait for the installation to complete.

Connect to Exchange Online

Once we’ve ensured the Exchange Online Module is installed, we can connect to Exchange Online.

In an elevated PowerShell window, run this command:

```
Connect-ExchangeOnline
```

This will start the Microsoft credential flow UI. Ensure you log in with an account that has the Exchange Administrator role or Organization Management role group membership.

Enabling Organization Customization

We now need to create management scopes for the app. This requires 'organization customization' to be enabled for the tenant. It is disabled by default. To check if it is enabled, run this command:

```
Get-OrganizationConfig | Format-List IsDehydrated
```

- **True** = Organization customization needs to be enabled
- **False** = Already enabled, skip to the next section

If it is **True**, run this command:

```
Enable-OrganizationCustomization
```

This command may take up to 30 minutes to complete. Wait for the prompt to return before proceeding.

You may encounter the following error:

```
Resource 'Processor (servername)' is unhealthy and shouldn't be accessed.
```

This is a transient backend infrastructure issue on Microsoft's side, not a configuration problem. To resolve:

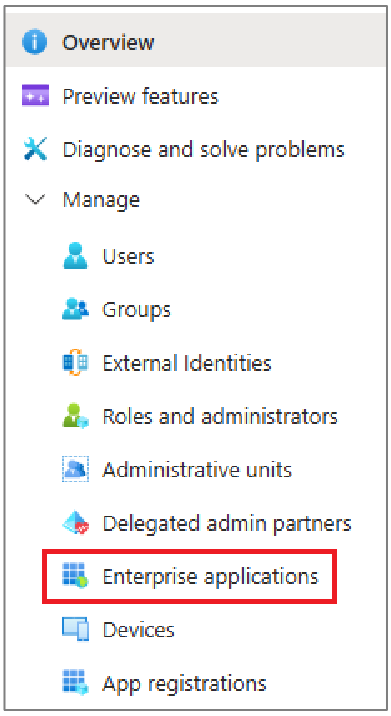
1. Wait 15-30 minutes and retry the command.
2. If it persists, wait up to 24 hours as these issues are usually temporary.
3. Check the Microsoft 365 Service Health Dashboard for any ongoing incidents.
4. If the issue continues beyond 24 hours, contact Microsoft Support through the Microsoft 365 admin centre.

IMPORTANT: You only need to run Enable-OrganizationCustomization once per tenant. After it completes successfully, you will not need to run it again.

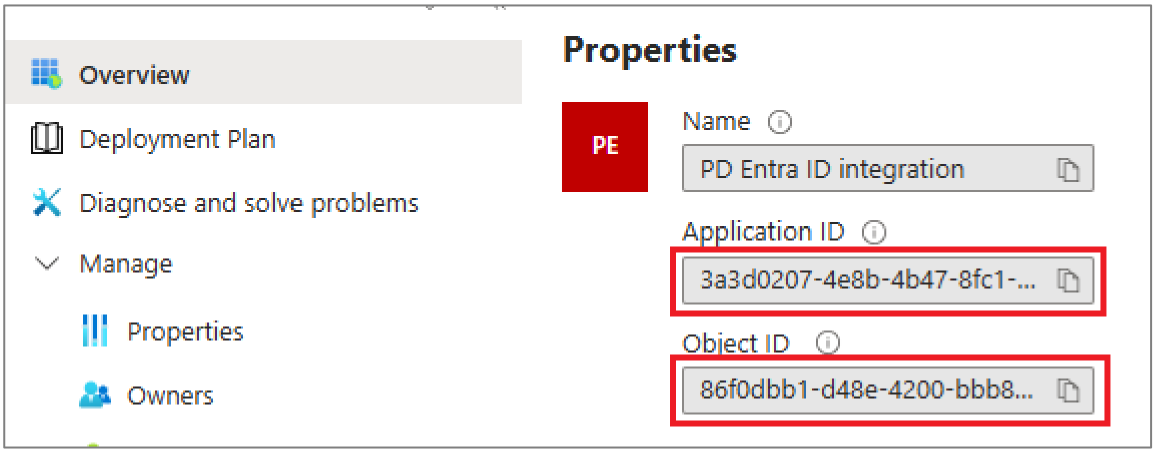
Obtain Application IDs from Entra ID

We need these for creating the Service Principal and to assign the Application Roles.

- Navigate to the **Enterprise Applications** page in Entra ID (not App Registrations):



- In the list of applications, click the app you created.
- On the **Overview** page, copy both:
  - **Application ID** (also called Client ID)
  - **Object ID** (the Service Principal Object ID)



Name: PD Entra ID integration

Application ID: 3a3d0207-4e8b-4b47-8fc1-...

Object ID: 86f0dbb1-d48e-4200-bbb8...

- **IMPORTANT:** Use the IDs from **Enterprise Applications**, not App Registrations. The Object IDs are different.

Create Service Principal in Exchange

Create a pointer to the Entra ID application in Exchange Online. Run this command:

```
New-ServicePrincipal -AppId "<Application-ID>" -ObjectId "<Object-ID>" -DisplayName
"Print Director"
```

Replace the placeholder values with your actual IDs.

Create Management Scope

Create a scope that identifies only the mailbox to be used by Print Director. If the mailbox to be used by PD has the address printdirector@yourdomain.com, this would be the command:

```
New-ManagementScope -Name "PrintDirector-Mailbox" -RecipientRestrictionFilter
"PrimarySmtpAddress -eq 'printdirector@yourdomain.com'"
```

Ensure you change the email address to the mailbox address you want to use.

Assign Application Roles

Assign the required permissions to the service principal with the management scope.

This command is not only for reading/downloading mails (for email-to-print), but also for sending emails with large attachments that need to be batch uploaded:

```
New-ManagementRoleAssignment -Name "PD-MailReadWrite" -Role "Application
Mail.ReadWrite" -App "<Application-ID>" -CustomResourceScope "PrintDirector-Mailbox"
```

Ensure you replace <Application-ID> with the Application ID you copied above.

This command is also needed for sending emails:

```
New-ManagementRoleAssignment -Name "PD-MailSend" -Role "Application Mail.Send" -App
"<Application-ID>" -CustomResourceScope "PrintDirector-Mailbox"
```

Again, ensure you replace <Application-ID> with the Application ID you copied above.

Test the Configuration

You could test the configuration by creating the record in PD and trying to send emails and receive email-to-prints. But to quickly check the configuration, we can run some commands in PowerShell. Note that the configuration changes can take up to 2 hours to propagate due to caching.

- Verify Access to the Configured Mailbox (change the <Application-ID> and mailbox address):

```
Test-ServicePrincipalAuthorization -Identity "<Application-ID>" -Resource
"printdirector@yourdomain.com"
```

- Expected result: **InScope = True**
- Verify No Access to Other Mailboxes:

```
Test-ServicePrincipalAuthorization -Identity "<Application-ID>" -Resource
"anotheruser@yourdomain.com"
```

- Expected result: **InScope = False**

IMPORTANT: Wait Time Required

After completing the RBAC configuration, you must wait 30 minutes to 2 hours before the permissions become active. The Test-ServicePrincipalAuthorization command shows results immediately because it bypasses the cache, but actual email operations will fail until the cache propagates.

If you receive "Access denied, check credentials" errors immediately after configuration, wait 2 hours and try again.

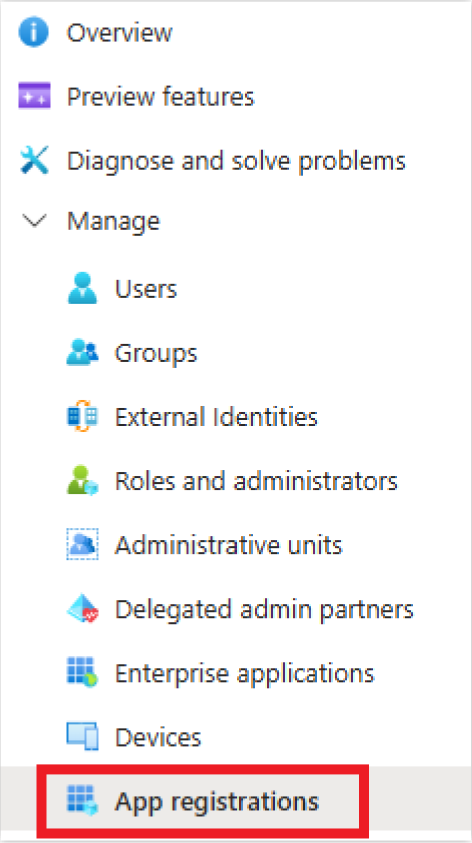
Create MS Graph application authentication record in PD

The application authentication record can be created in PD via the **Entra ID Sync** Scheduled Task, the **Email Send Accounts**, or the **Email Receive Accounts**. The same record can be used for all of them if the azure app has all the appropriate permissions. For example, if you created the record via an Entra ID Sync, you can just select the same one when creating an Email Send Account.

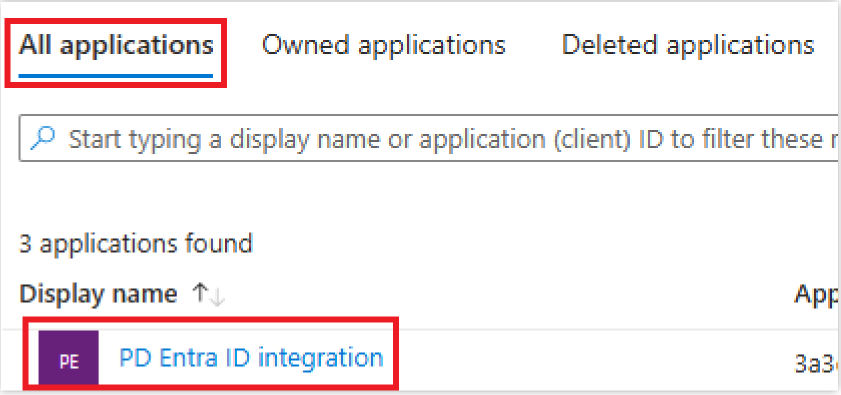
Obtain the Application ID (client ID) and the Directory ID (tenant ID)

Firstly, we'll need the Application ID (client ID) and the Directory ID (tenant ID) from the portal.

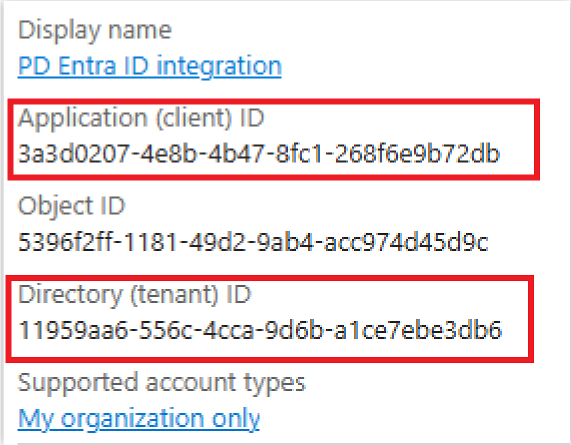
- In Entra ID, click **App registrations**:



- Click **All applications** and then click the app that you created.



- In the **Overview** page, copy the **Application (client) ID and** the **Directory (tenant) ID**.



Display name
PD Entra ID integration

Application (client) ID
3a3d0207-4e8b-4b47-8fc1-268f6e9b72db

Object ID
5396f2ff-1181-49d2-9ab4-acc974d45d9c

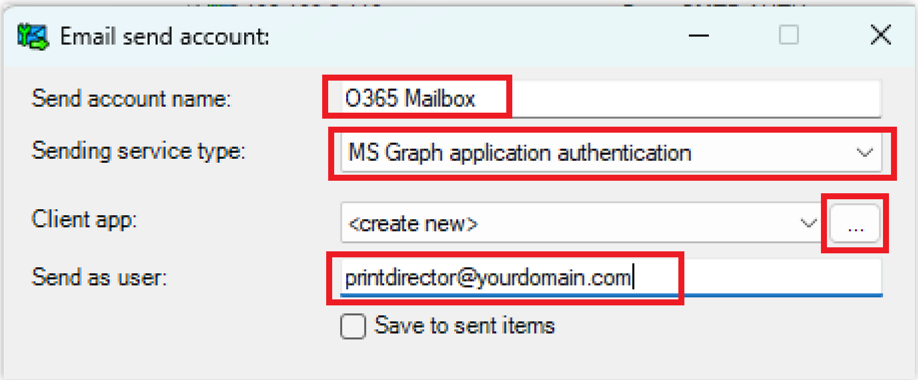
Directory (tenant) ID
11959aa6-556c-4cca-9d6b-a1ce7ebe3db6

Supported account types
My organization only

Creating the MS Graph application record for an Email Send Account

This example uses an Email Send Account to create the MS Graph application record for PD. But it can just as easily be created in an Entra ID Sync, or Email Receive Account.

1. In Resource Manager, navigate to **Email Management** > **Email Send Accounts.**
2. Click **Add**.
3. Enter a name for the account.
4. Change the **Sending service type** dropdown to **MS Graph application authentication**.
5. Enter the mailbox address in the **Send as user** textbox.
6. If you had already created the MS Graph app record via an Entra ID Sync scheduled task or Email Receive Account, you could select it from the Client app drop down. In this example, we'll be creating one.
7. Click the … button next to the **Client app** dropdown.



Send account name: O365 Mailbox

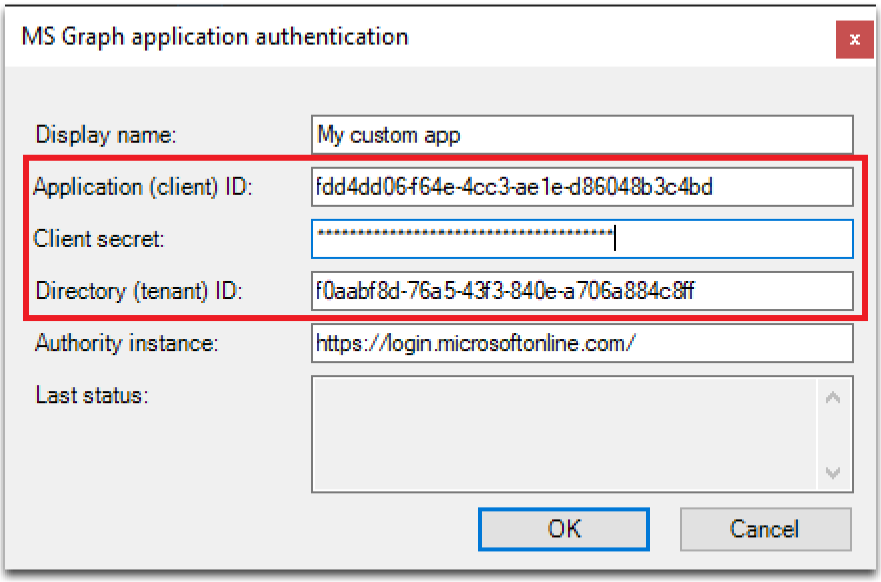
Sending service type: MS Graph application authentication

Client app: <create new>

Send as user: printdirector@yourdomain.com

□ Save to sent items

- In the search window that opens, click the **New** button.
- Enter any name for the custom application.
- Paste the values copied from the **Application Overview** details (along with the client secret) into the fields and click OK.



MS Graph application authentication

Display name: My custom app

Application (client) ID: fdd4dd06-f64e-4cc3-ae1e-d86048b3c4bd

Client secret: **

Directory (tenant) ID: f0aabf8d-76a5-43f3-840e-a706a884c8ff

Authority instance: https://login.microsoftonline.com/

Last status:

Now you can test sending emails.

Document revision date: 2026/02/03
Software version: 2.5.10.1