Concord
TECHNOLOGIES

Federated SSO Quick Start Guide

Azure AD

V1.4

Concord Technologies

2025 First Avenue Suite 800

Seattle, WA, 98121, USA

Call Us: +1 206-441-3346

Concord.net

Contents

Overview

This document is intended to provide information on setting up federation in your Azure AD and Concord environments. The process of creating a federated relationship between Azure AD and Concord consists of registering an application on the Azure AD portal and copying bits of information between the Azure AD portal and the Federation tab on the Concord admin portal.

This document will provide a basic set of instructions for registering the application on the Azure AD portal as well as what information needs to be copied from the Azure AD portal to the Concord portal and vice versa.

Note that Concord does not control or maintain the configuration settings within Azure AD. While the steps depicted in the following document have been used to successfully configure Federation in Azure AD, these may be subject to change and additional configuration may be needed to achieve your own desired results. You are advised to consult Microsoft's Azure AD OpenID Connect documentation for further assistance as needed.

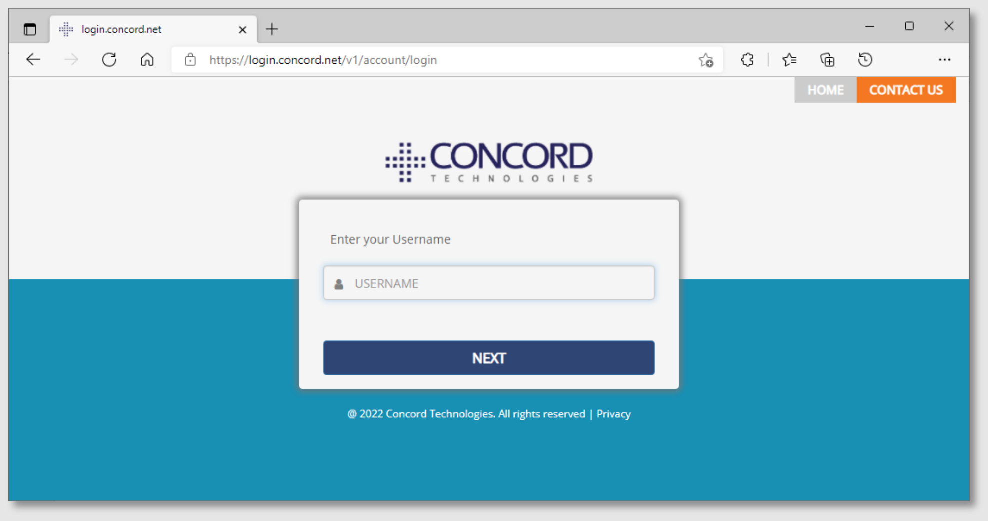
Concord is not responsible for any issues or service interruptions resulting from configuration steps taken in Azure AD or the Concord Admin Portal to configure or enable federated access to Concord services.

For more detailed information on this integration, or for integrating other federation providers, please see the more comprehensive Concord Federation Admin & User Guide.

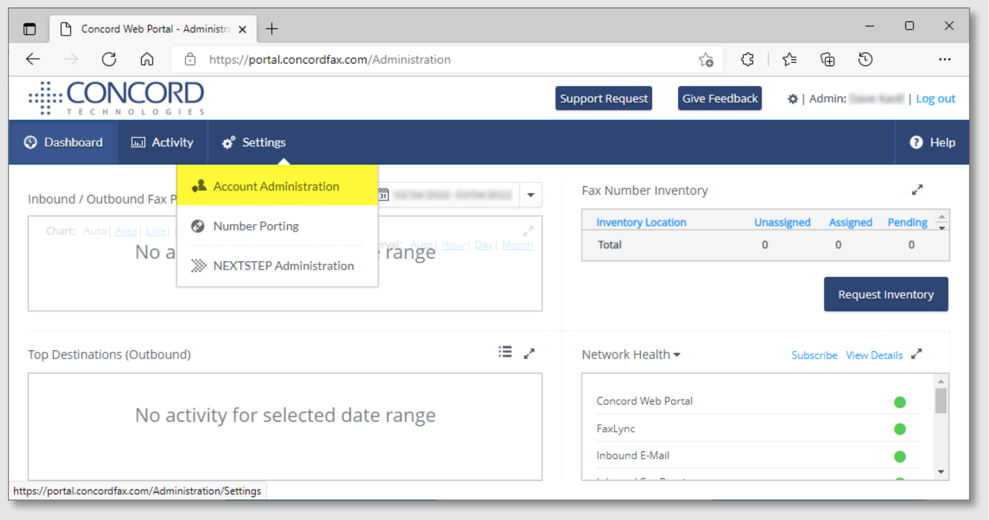
Azure AD Application Registration

Section 1: Concord Portal Federation Configuration – Phase 1

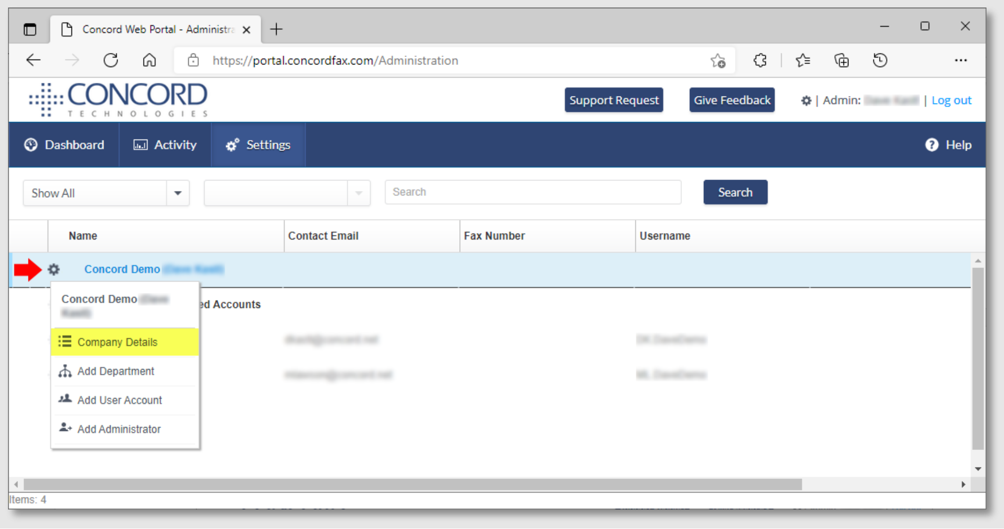
1. Login to the Concord portal using an administrative account with permissions to access the Federated tab.


login.concord.net
https://login.concord.net/v1/account/login
HOME
CONTACT US
CONCORD TECHNOLOGIES
Enter your Username
USERNAME
NEXT
@ 2022 Concord Technologies. All rights reserved | Privacy

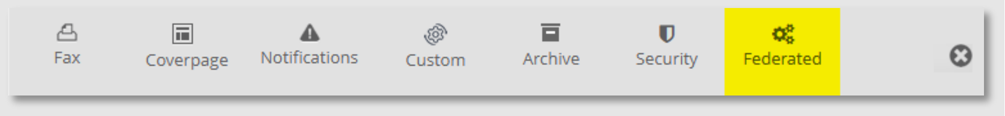
2. Click Setting->Account Administration.



3. On the Account Administration page, click the gear icon next to the company account name and select **Company Details**.



4. On the **Company Details** page, select the **Federated** tab. ***NOTE:*** If you do not see the Federated tab, ensure you are logged in with an administrative account and that account has been granted access to the Federated tab. You may need to contact Concord Premium Support to have your account enabled for federation access.



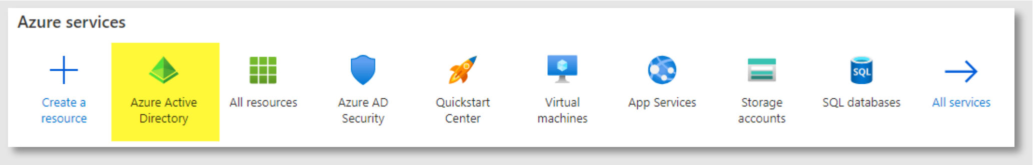
5. On the **Federated** tab, click the **Copy URL** button to copy the **Redirect URL**. It is recommended to copy this value as it will be used when we create the Azure AD application in the next phase.

This completes the first stage of configuration within the Concord admin portal. The next stage is to create and configure the Azure AD application that will be used to enable federation.

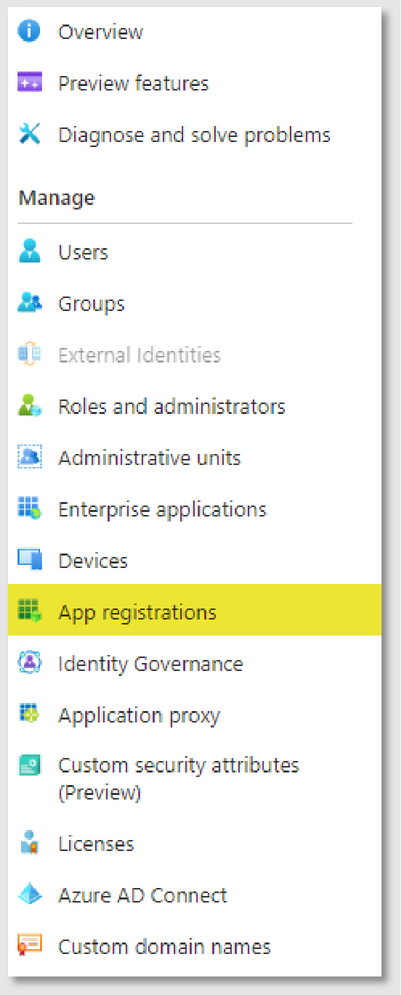
Section 2: Azure AD Application Creation

1. Connect to the Azure portal.

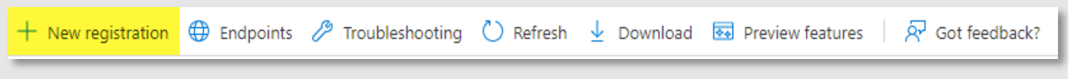
2. Click "Azure Active Directory".



3. Select "App Registrations" from the menu on the left of the screen:

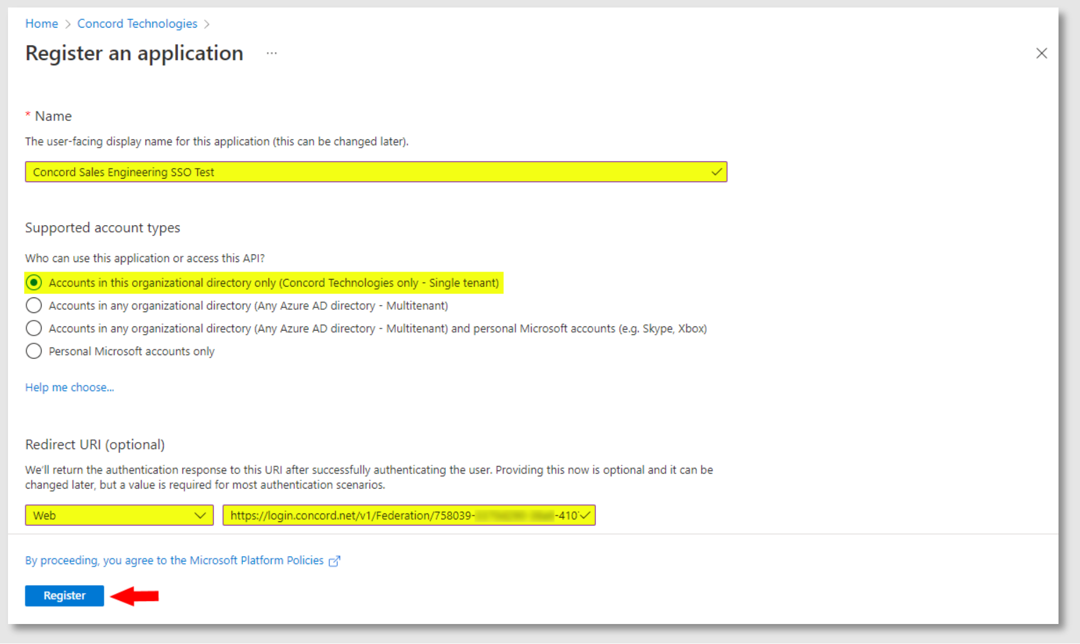


4. Click “New registration” to create a new application:



5. On the **Register an Application** page, perform the following actions:
   - Enter a descriptive name for the new application
   - Ensure **Single Tenant** is selected
   - Select **Web** as the Redirect URL platform type
   - Paste the redirect URL from Step #5 in the previous section (Concord Portal Federation Configuration – Phase 1) into the redirect URL textbox

6. Click the **Register** button to register the new application



Register an application ···

* Name

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

Concord Sales Engineering SSO Test ✓

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Concord Technologies only - Single tenant)
○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)
○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
○ Personal Microsoft accounts only

Redirect URI (optional)

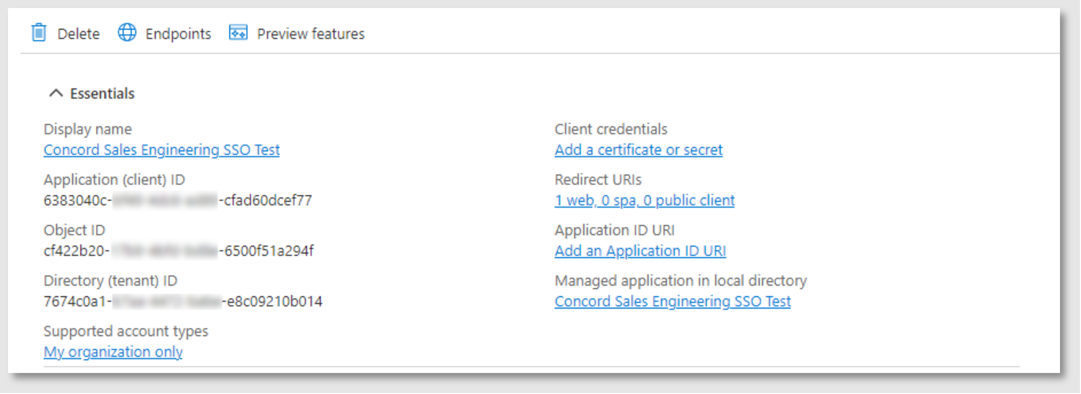
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web

https://login.concord.net/v1/Federation/758039-[redacted]-410 ✓

By proceeding, you agree to the Microsoft Platform Policies

7. Once the application is registered, a page similar to the following should be displayed:



Essentials

Display name
Concord Sales Engineering SSO Test

Application (client) ID
6383040c-[redacted]-cfad60dcef77

Object ID
cf422b20-[redacted]-6500f51a294f

Directory (tenant) ID
7674c0a1-[redacted]-e8c09210b014

Supported account types
My organization only

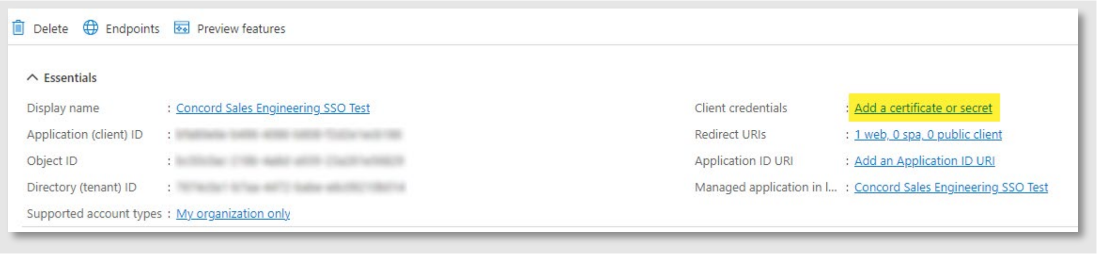
Client credentials
Add a certificate or secret

Redirect URIs
1 web, 0 spa, 0 public client

Application ID URI
Add an Application ID URI

Managed application in local directory
Concord Sales Engineering SSO Test

8. Click the **Add a certificate or secret link** shown next to the Client credentials item.

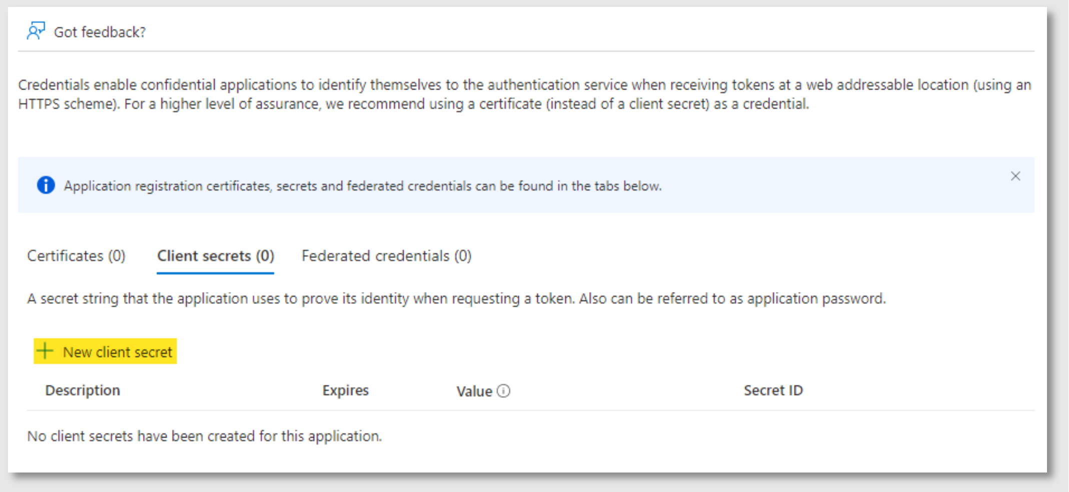


Essentials

Display name : Concord Sales Engineering SSO Test
Application (client) ID : [redacted]
Object ID : [redacted]
Directory (tenant) ID : [redacted]
Supported account types : My organization only

Client credentials : Add a certificate or secret
Redirect URIs : 1 web, 0 spa, 0 public client
Application ID URI : Add an Application ID URI
Managed application in l... : Concord Sales Engineering SSO Test

9. On the Certificates & Secrets page, click the + New client secret link:



Got feedback?

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

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

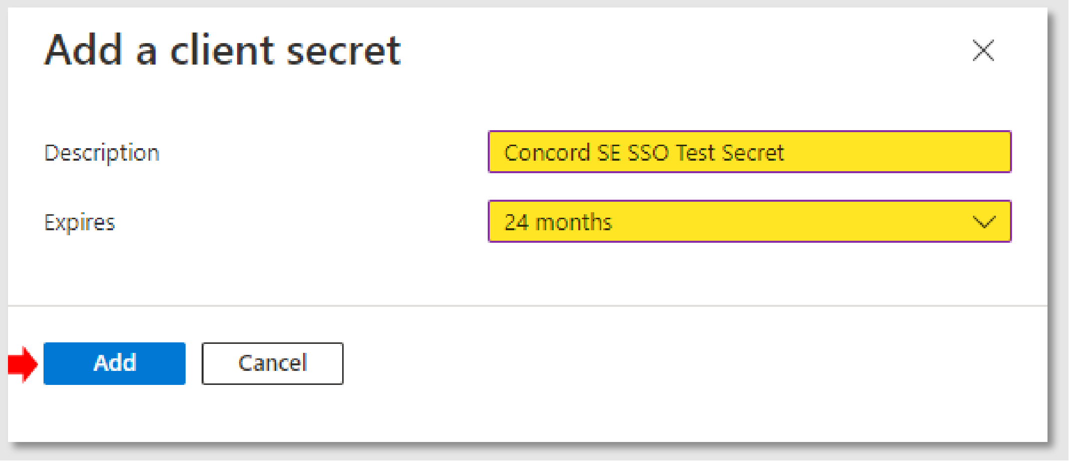
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value ⓘ	Secret ID

No client secrets have been created for this application.

10. On the **Add a Client Secret** page, enter a descriptive name for the secret, set the expiration date for the secret, and click the **Add** button to create the secret. ***NOTE:*** the maximum duration for a client secret is 24 months. Once a secret has expired, you will need to generate a new client secret and update the Federated tab with the new secret value in the Concord admin portal.



Add a client secret

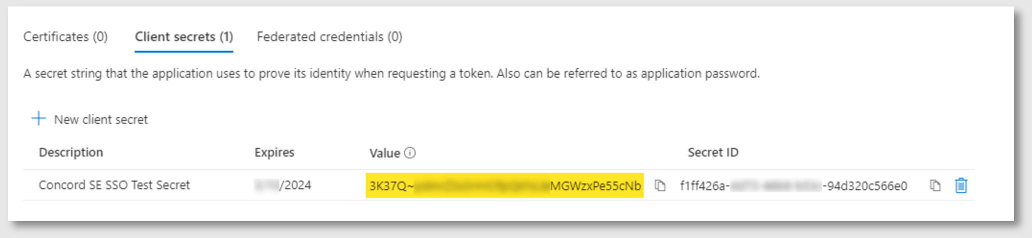
Description: Concord SE SSO Test Secret

Expires: 24 months

11. Once the **Add** button has been clicked, a summary of the client secret will be shown. Copy the client secret **Value** as it will be needed when completing the federation configuration within the Concord admin portal.

    NOTE: This is the only opportunity to copy the client secret value, if the value is not copied at this point, a new client secret will need to be generated.

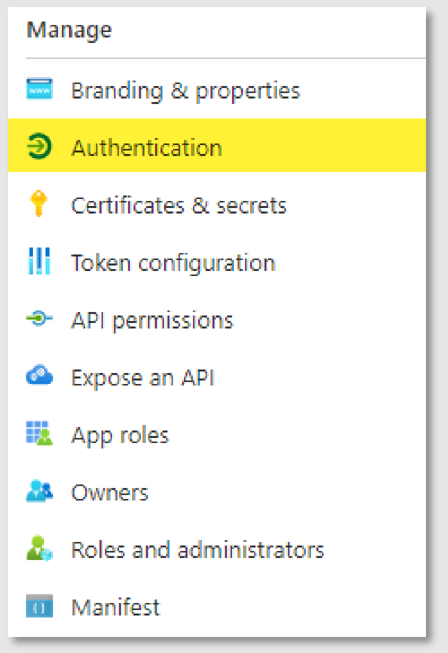
    NOTE: The client secret is a sensitive value and should not be shared.



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

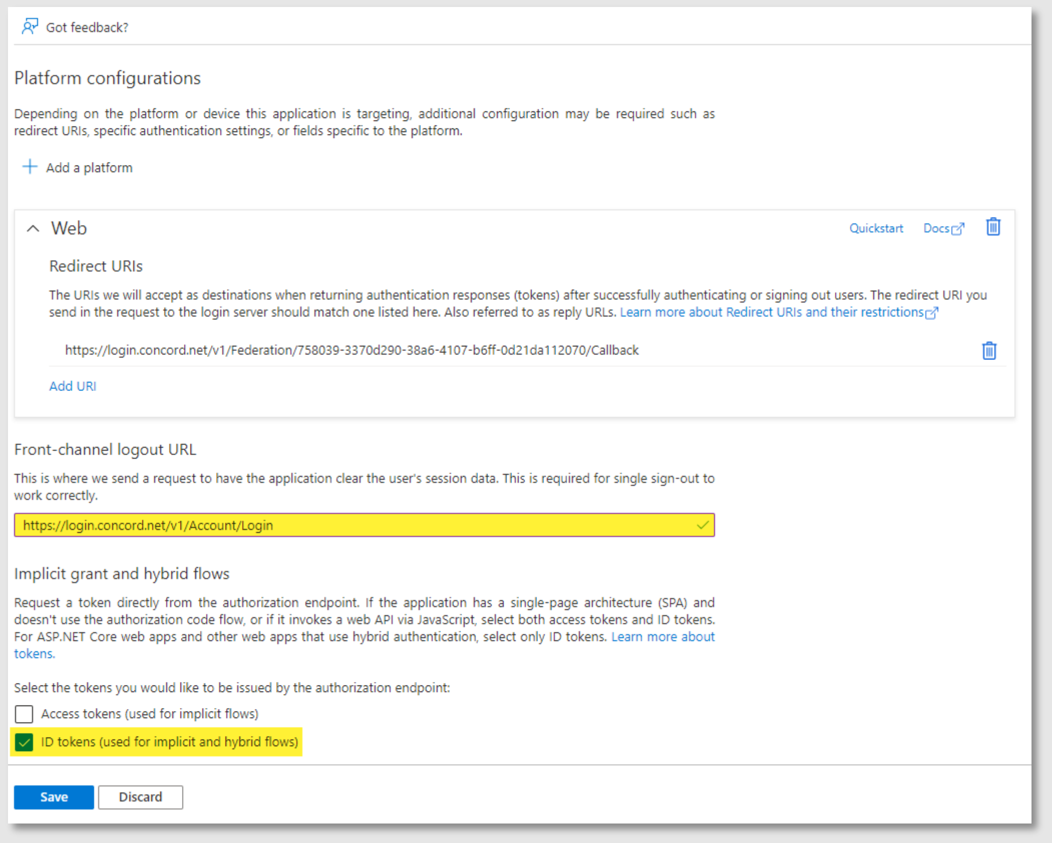
Description	Expires	Value ⓘ	Secret ID
Concord SE SSO Test Secret	[redacted]/2024	3K37Q~[redacted]MGWzxPe55cNb	f1ff426a-[redacted]-94d320c566e0

12. Click on **Authentication** from the menu on the right.



13. Enter the URL to the Concord login screen in the logout URL section. Also, ensure the ID tokens checkbox is checked. Click the Save button to save the changes.

    NOTE: The Concord login URL is:
    https://login.concord.net/v1/Account/Login



Got feedback?

Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

Web

Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating or signing out users. The redirect URI you send in the request to the login server should match one listed here. Also referred to as reply URLs. Learn more about Redirect URIs and their restrictions

https://login.concord.net/v1/Federation/758039-3370d290-38a6-4107-b6ff-0d21da112070/Callback

Front-channel logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

https://login.concord.net/v1/Account/Login ✓

Implicit grant and hybrid flows

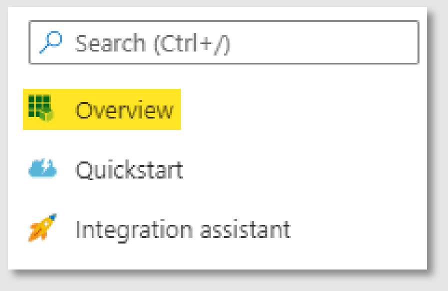
Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. Learn more about tokens.

Select the tokens you would like to be issued by the authorization endpoint:

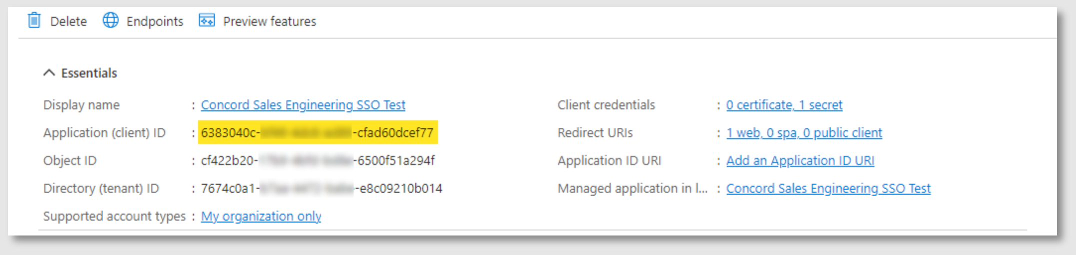
☐ Access tokens (used for implicit flows)

☑ ID tokens (used for implicit and hybrid flows)

14. Click on **Overview** from the menu on the right of the screen to display the application overview page.



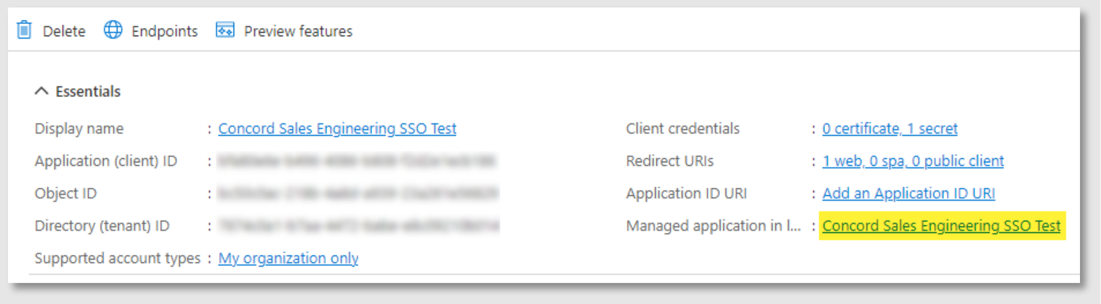
15. On the **Overview** page, copy the **Application (client) ID** value as you will enter that value into the Concord federation tab.



Essentials

Display name	: Concord Sales Engineering SSO Test	Client credentials	: 0 certificate, 1 secret
Application (client) ID	: 6383040c-[redacted]-cfad60dcef77	Redirect URIs	: 1 web, 0 spa, 0 public client
Object ID	: cf422b20-[redacted]-6500f51a294f	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: 7674c0a1-[redacted]-e8c09210b014	Managed application in l...	: Concord Sales Engineering SSO Test
Supported account types	: My organization only		

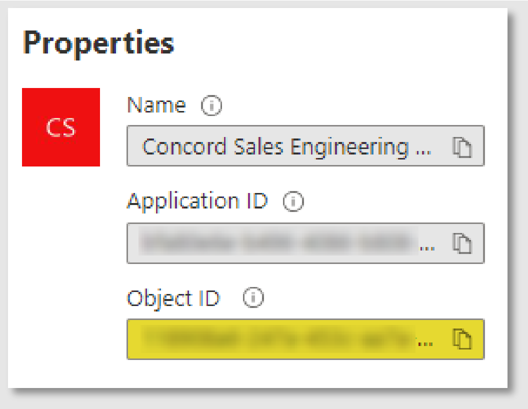
16. Click the link shown next to the **Managed application in local directory** item (this value will vary depending on how you named your Azure AD application):



Essentials

Display name	: Concord Sales Engineering SSO Test	Client credentials	: 0 certificate, 1 secret
Application (client) ID	: [redacted]	Redirect URIs	: 1 web, 0 spa, 0 public client
Object ID	: [redacted]	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: [redacted]	Managed application in l...	: Concord Sales Engineering SSO Test
Supported account types	: My organization only		

17. Copy the **Object ID** value as this value may be needed if you would like to use custom claims when associating a claim policy with this application.



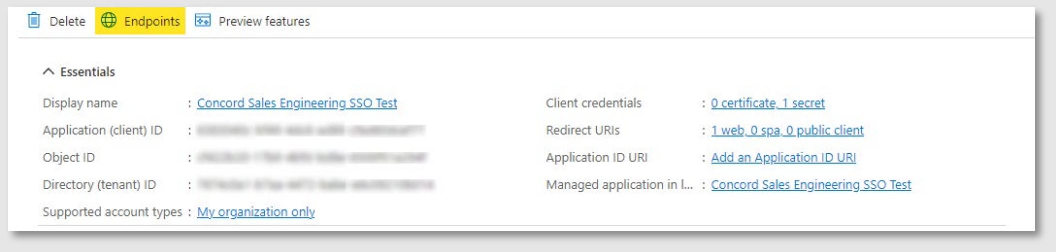
Properties

Name: Concord Sales Engineering ...

Application ID: [redacted]

Object ID: [redacted]

18. Click the **Endpoints** link to display the URL endpoints exposed by this tenant:



Display name : Concord Sales Engineering SSO Test

Application (client) ID : [redacted]

Object ID : [redacted]

Directory (tenant) ID : [redacted]

Supported account types : My organization only

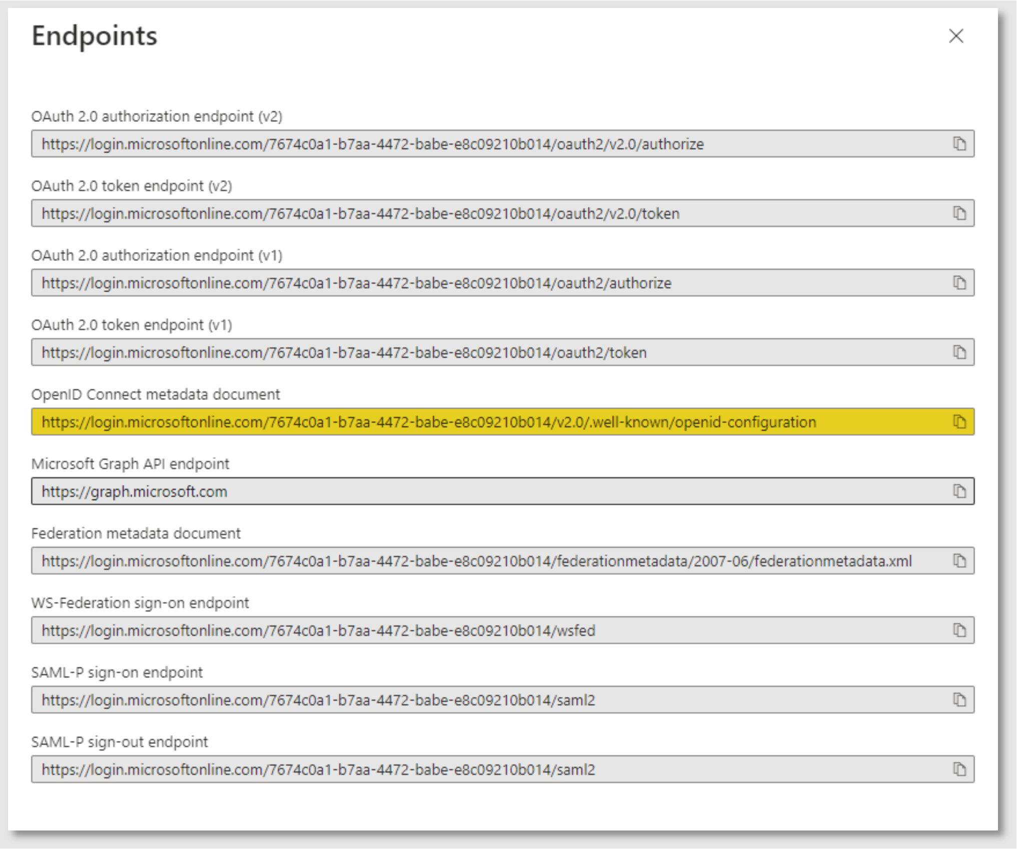
Client credentials : 0 certificate, 1 secret

Redirect URIs : 1 web, 0 spa, 0 public client

Application ID URI : Add an Application ID URI

Managed application in l... : Concord Sales Engineering SSO Test

19. On the **Endpoints** page, copy the **OpenID Connect metadata document** URL as you will enter that value into the Concord federation tab:



Endpoints

OAuth 2.0 authorization endpoint (v2)
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/oauth2/v2.0/authorize

OAuth 2.0 token endpoint (v2)
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/oauth2/v2.0/token

OAuth 2.0 authorization endpoint (v1)
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/oauth2/authorize

OAuth 2.0 token endpoint (v1)
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/oauth2/token

OpenID Connect metadata document
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/v2.0/.well-known/openid-configuration

Microsoft Graph API endpoint
https://graph.microsoft.com

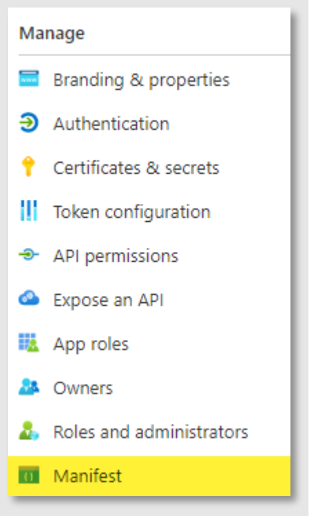
Federation metadata document
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/federationmetadata/2007-06/federationmetadata.xml

WS-Federation sign-on endpoint
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/wsfed

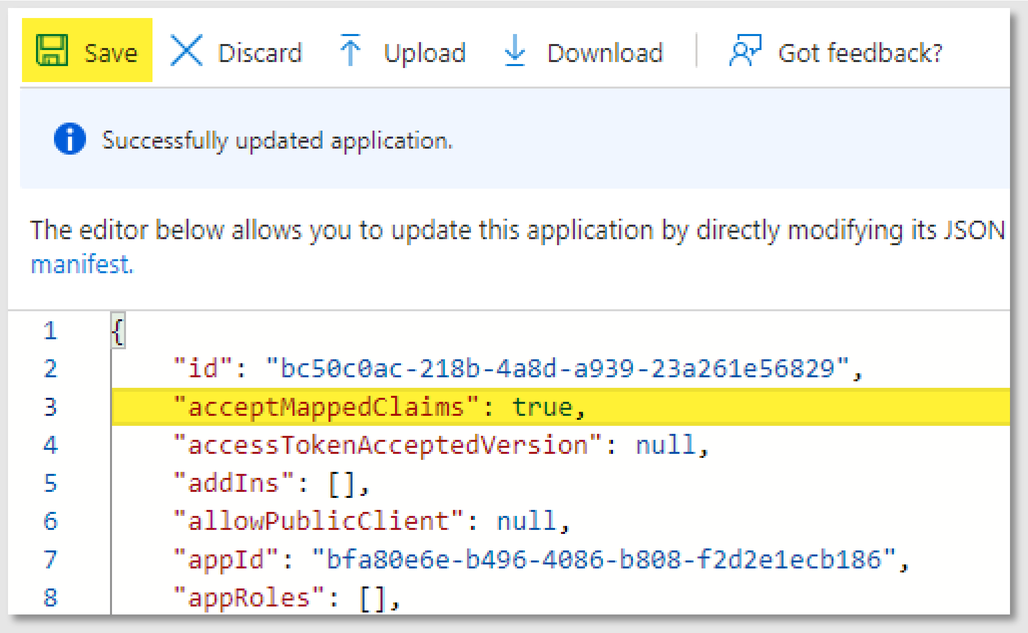
SAML-P sign-on endpoint
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/saml2

SAML-P sign-out endpoint
https://login.microsoftonline.com/7674c0a1-b7aa-4472-babe-e8c09210b014/saml2

20. Click on **Manifest** from the menu on the right of the screen to display the application manifest settings.



21. Modify the **acceptMappedClaims** value and set it to **true** and click the **Save** link to save the changes:



Successfully updated application.

The editor below allows you to update this application by directly modifying its JSON manifest.

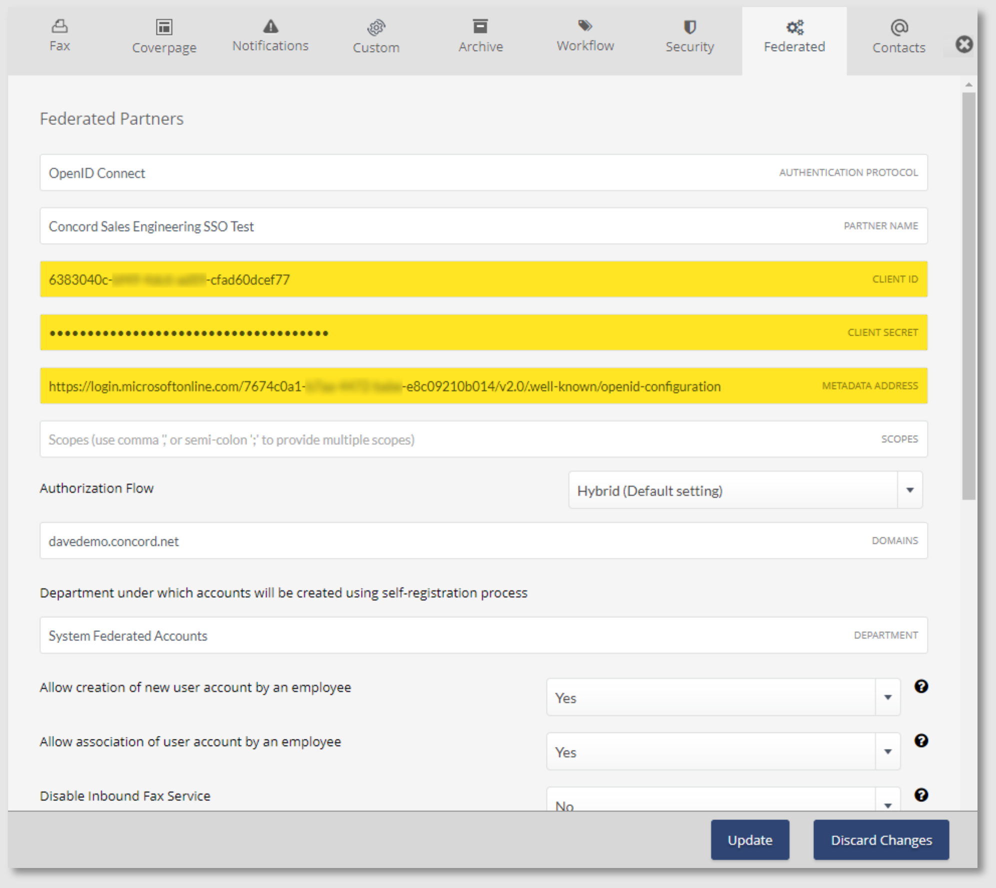
```
{
    "id": "bc50c0ac-218b-4a8d-a939-23a261e56829",
    "acceptMappedClaims": true,
    "accessTokenAcceptedVersion": null,
    "addIns": [],
    "allowPublicClient": null,
    "appId": "bfa80e6e-b496-4086-b808-f2d2e1ecb186",
    "appRoles": [],
```

This completes the Azure AD configuration stage. We will now take the values created and copied from the Azure application and apply them to the Concord Federated tab.

Section 3: Concord Portal Federation Configuration – Phase 2

1. In a browser, navigate to the Federated tab on the Concord admin portal as described in Phase 1 of this process.

2. Paste the following values copied from the Azure AD configuration into the federated tab:
   - Client ID
   - Client Secret
   - Metadata Address



Federated Partners

OpenID Connect — AUTHENTICATION PROTOCOL

Concord Sales Engineering SSO Test — PARTNER NAME

6383040c-[redacted]-cfad60dcef77 — CLIENT ID

•••••••••••••••••••••••••••••••••••••• — CLIENT SECRET

https://login.microsoftonline.com/7674c0a1-[redacted]-e8c09210b014/v2.0/.well-known/openid-configuration — METADATA ADDRESS

Scopes (use comma ',' or semi-colon ';' to provide multiple scopes) — SCOPES

Authorization Flow: Hybrid (Default setting)

davedemo.concord.net — DOMAINS

Department under which accounts will be created using self-registration process

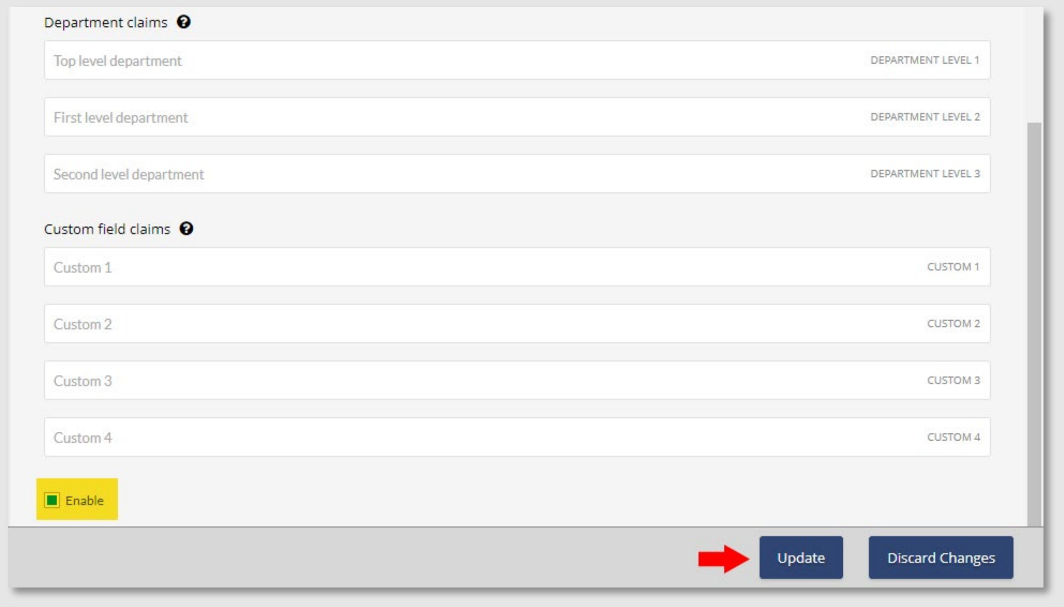
System Federated Accounts — DEPARTMENT

Allow creation of new user account by an employee: Yes

Allow association of user account by an employee: Yes

Disable Inbound Fax Service: No

3. Ensure the **Enable** checkbox is checked and click the **Update** button to enable federation:



Department claims

Top level department	DEPARTMENT LEVEL 1
First level department	DEPARTMENT LEVEL 2
Second level department	DEPARTMENT LEVEL 3

Custom field claims

Custom 1	CUSTOM 1
Custom 2	CUSTOM 2
Custom 3	CUSTOM 3
Custom 4	CUSTOM 4

☑ Enable

This completes the second phase of the federation configuration in the Concord admin portal. At this point, any user who attempts to login to the Concord portal with a username containing the redirect domain will be redirected to the configured identity management endpoint.

This may complete all the requirements you have for federation. If you are interested in using custom claims to create users in specific Concord departments, see **Appendix A** which describes the process of creating and assigning custom claims.

Best Practices

- It is highly recommended that you create an administrative account that has access to the federated tab but that itself does not use federation. The reason being is that if, for some reason, you enable federation and there is an issue, this admin account can easily login to the Concord portal and disable federation.

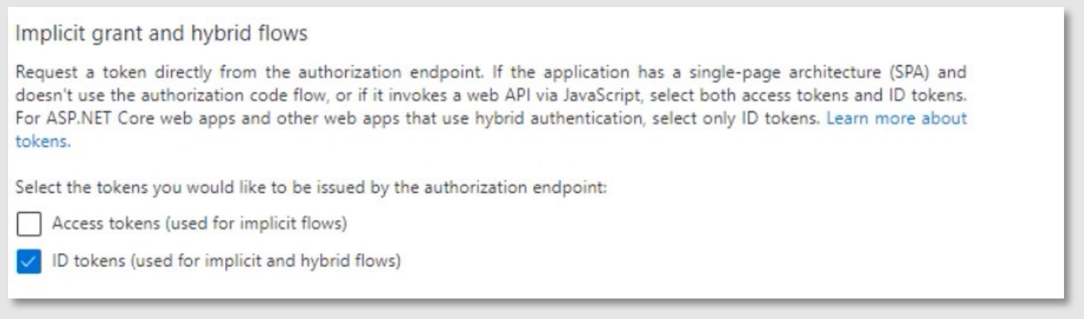
  Without this non-federated admin account, it is possible that you could lock all users out of the Concord portal with no ability to disable federation.

  An example of a non-federated admin account would be to create an admin with the username of “FirstName.LastName” rather than user@domain.com where “domain.com” is the federated domain.

- Related to the first bullet point, if you create admin and user accounts for the same person, we recommend that the admin account use this “FirstName.LastName” convention and the user account uses the e-mail address for that user.

  This is to ensure that if you choose to use any of the Concord client utilities, which require a user account to authenticate to the Concord platform, that you can use federation for that user account.

- Ensure that the “ID Tokens” checkbox is checked within the Azure AD configuration:



Implicit grant and hybrid flows

Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. Learn more about tokens.

Select the tokens you would like to be issued by the authorization endpoint:

☐ Access tokens (used for implicit flows)

☑ ID tokens (used for implicit and hybrid flows)

Getting Help

Concord's customer service team is available Monday–Friday from 6:00 AM to 6:00 PM (Pacific Time).

- Phone: +1 (206) 441-3346
- Email: premiumsupport@concord.net
- Web: https://concord.net/about/contact-us/

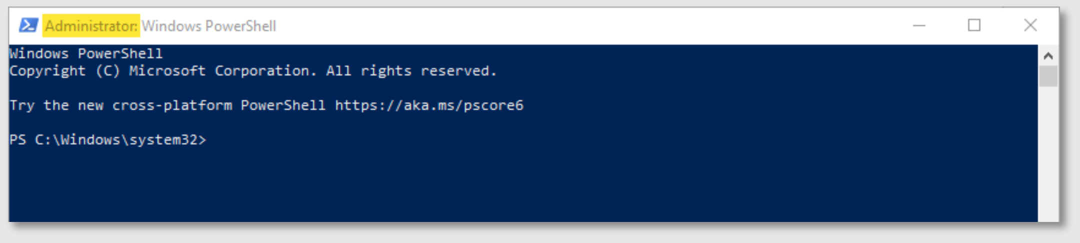
Appendices

Appendix A – Custom Claims

Custom claims can be created and associated with a registered Azure AD application. These claims can then be used to specify additional user properties such as the department to create the Concord user in (when auto-creating federated users), or to specify custom user fields that can be used for cover sheets or reporting.

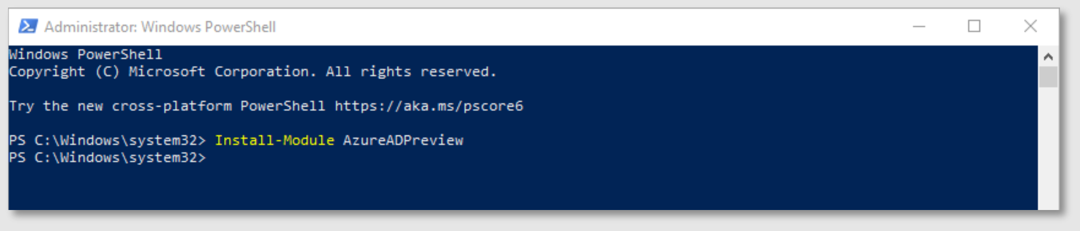
NOTE: The method described here to create custom claims requires PowerShell scripting with PowerShell running with administrative rights. This is understood to be the method supported for creating custom claims in Azure AD as described in [this documentation]() titled "Provide Optional Claims to Your App" from Microsoft, dated November 11, 2022.

1. Open a PowerShell window with administrative rights:



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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32>
```

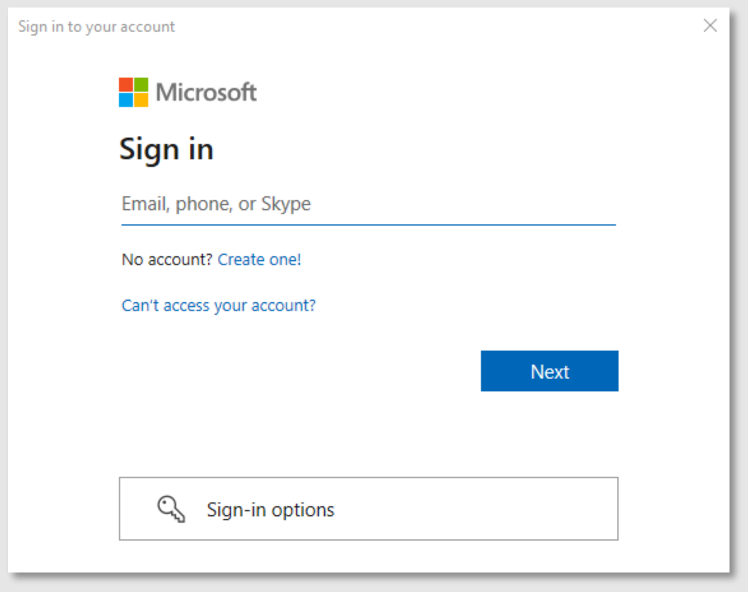
2. Install the AzureADPreview package with the command **Install-Module AzureADPreview**:



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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Install-Module AzureADPreview
PS C:\Windows\system32>
```

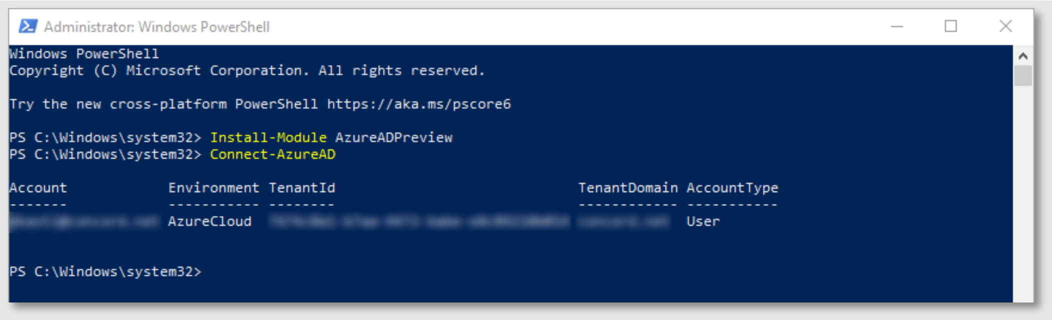
3. Connect to your Azure AD instance using the command **Connect-AzureAD:**

   NOTE: You may be prompted to authenticate to the Azure AD instance.



4. Once authenticated, a summary of your connection will be displayed:

   NOTE: Depending on your environment, you may need to go back to step 3 and specify additional parameters for the **Connect-AzureAD** command. Type **Connect-AzureAD -?** to get help information about the command.



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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Install-Module AzureADPreview
PS C:\Windows\system32> Connect-AzureAD

Account            Environment TenantId                             TenantDomain AccountType
-------            ----------- --------                             ------------ -----------
[redacted]         AzureCloud  [redacted]                           [redacted]   User


PS C:\Windows\system32>
```

5. Create a new custom claim using the New-AzureADPolicy command.

   NOTE: As your custom claims may differ, the examples provided here are just for demonstration purposes only, you may need to investigate your organization's identity management configuration and adjust your custom claims accordingly.

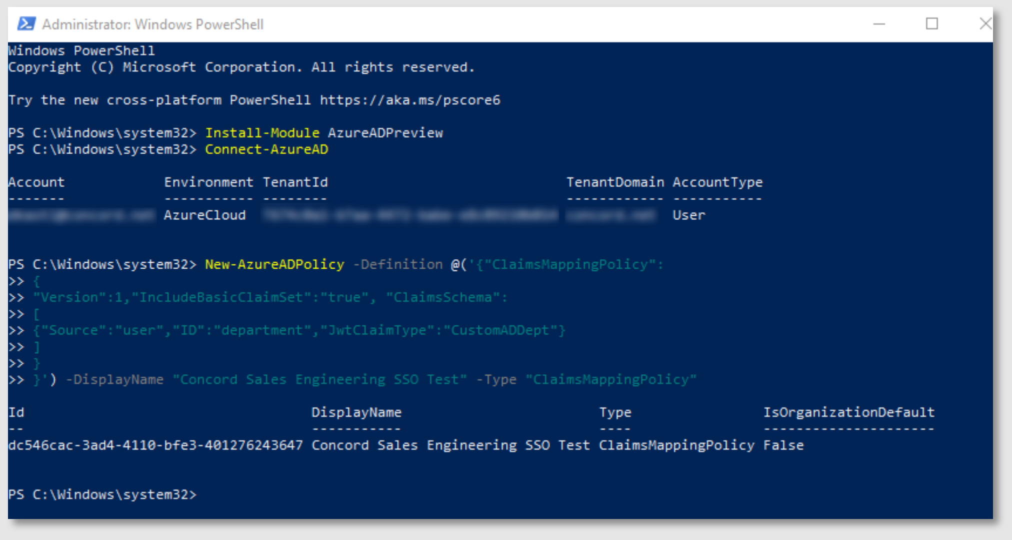
   NOTE: Additional information about the New-AzureADPolicy command can be found here.

6. An example custom claims request may look like this:

```
New-AzureADPolicy -Definition @('{"ClaimsMappingPolicy":
{
               "Version":1,"IncludeBasicClaimSet":"true", "ClaimsSchema":
               [
                      {"Source":"user","ID":"department","JwtClaimType":"CustomADDept"}
               ]
       }
}') -DisplayName "Concord Sales Engineering SSO Test" -Type "ClaimsMappingPolicy"
```

7. In the above sample, **Source** and **ID** indicate where the data in the claim is sourced from. In this case, we want to use the **department** property of the **user** object.

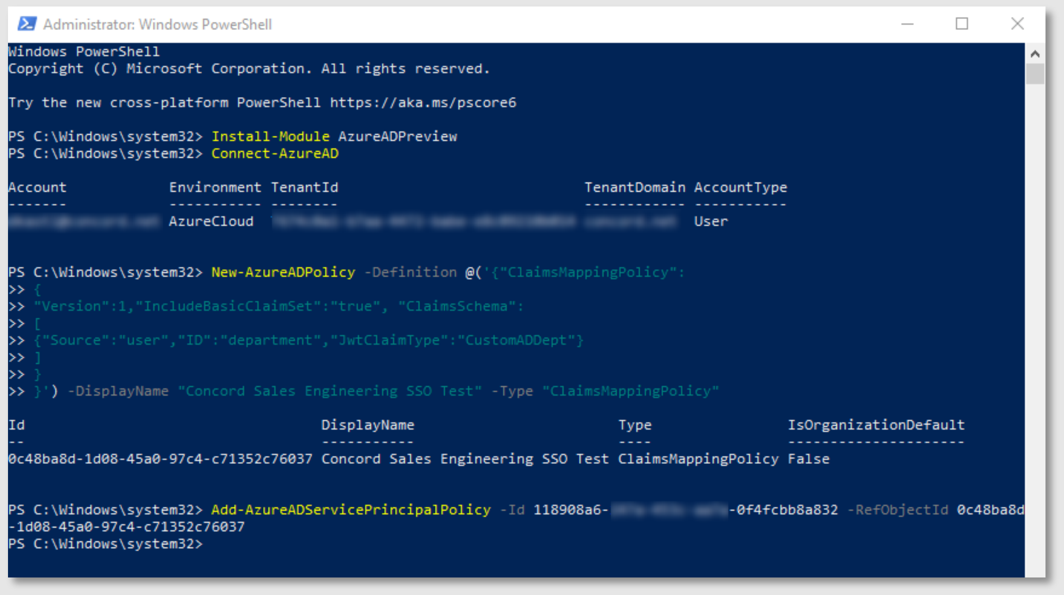
8. The **JwtClaimType** is the name of the claim to be emitted in the JSON Web Token (JWT). This is the value you will enter in the Concord federated tab as the custom claim to use.



Administrator: Windows PowerShell

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Install-Module AzureADPreview
PS C:\Windows\system32> Connect-AzureAD

Account              Environment TenantId                              TenantDomain AccountType
-------              ----------- --------                              ------------ -----------
[redacted]           AzureCloud  [redacted]                            [redacted]   User


PS C:\Windows\system32> New-AzureADPolicy -Definition @('{"ClaimsMappingPolicy":
>> {
>> "Version":1,"IncludeBasicClaimSet":"true", "ClaimsSchema":
>> [
>> {"Source":"user","ID":"department","JwtClaimType":"CustomADDept"}
>> ]
>> }
>> }') -DisplayName "Concord Sales Engineering SSO Test" -Type "ClaimsMappingPolicy"

Id                                   DisplayName                        Type                IsOrganizationDefault
--                                   -----------                        ----                ---------------------
dc546cac-3ad4-4110-bfe3-401276243647 Concord Sales Engineering SSO Test ClaimsMappingPolicy False


PS C:\Windows\system32>
```

9. Finally, assign the newly created custom claim created in the previous step to the Azure AD application, using the **Object ID** value defined in **the Azure AD Application Creation** section, step #16:



Administrator: Windows PowerShell

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Install-Module AzureADPreview
PS C:\Windows\system32> Connect-AzureAD

Account              Environment TenantId                              TenantDomain AccountType
-------              ----------- --------                              ------------ -----------
[redacted]           AzureCloud  [redacted]                            [redacted]   User


PS C:\Windows\system32> New-AzureADPolicy -Definition @('{"ClaimsMappingPolicy":
>> {
>> "Version":1,"IncludeBasicClaimSet":"true", "ClaimsSchema":
>> [
>> {"Source":"user","ID":"department","JwtClaimType":"CustomADDept"}
>> ]
>> }
>> }') -DisplayName "Concord Sales Engineering SSO Test" -Type "ClaimsMappingPolicy"

Id                                   DisplayName                        Type                IsOrganizationDefault
--                                   -----------                        ----                ---------------------
0c48ba8d-1d08-45a0-97c4-c71352c76037 Concord Sales Engineering SSO Test ClaimsMappingPolicy False


PS C:\Windows\system32> Add-AzureADServicePrincipalPolicy -Id 118908a6-[redacted]-0f4fcbb8a832 -RefObjectId 0c48ba8d
-1d08-45a0-97c4-c71352c76037
PS C:\Windows\system32>
```

This completes the federated SSO configuration. You should now be able to test associating or creating users via the federated SSO process.