Set Up Federated Login for LastPass Using Azure Active Directory

Table of Contents

Set Up Federated Login for LastPass Using Azure Active Directory

This guide provides setup instructions for using LastPass with Azure Active Directory (Azure AD) for your LastPass Enterprise or LastPass Identity account.

Summary

LastPass supports the following provisioning features:

- Create Users
- Update User Attributes
- Sync User Groups
- Deactivate or Disable Users

Federated login for LastPass Enterprise and LastPass Identity accounts allows users to log in to LastPass using their Azure AD account (instead of a username and separate Master Password) to access their LastPass Vault.

System Requirements

The enable federated login for LastPass using Azure AD, the following is required:

- An active Premium subscription to Microsoft Azure AD
- An active trial or paid LastPass Enterprise or LastPass Identity account
- An active LastPass Enterprise or LastPass Identity admin (required when activating your trial or paid account)

The LastPass Azure AD SCIM endpoint for federated login does not require any software installation.

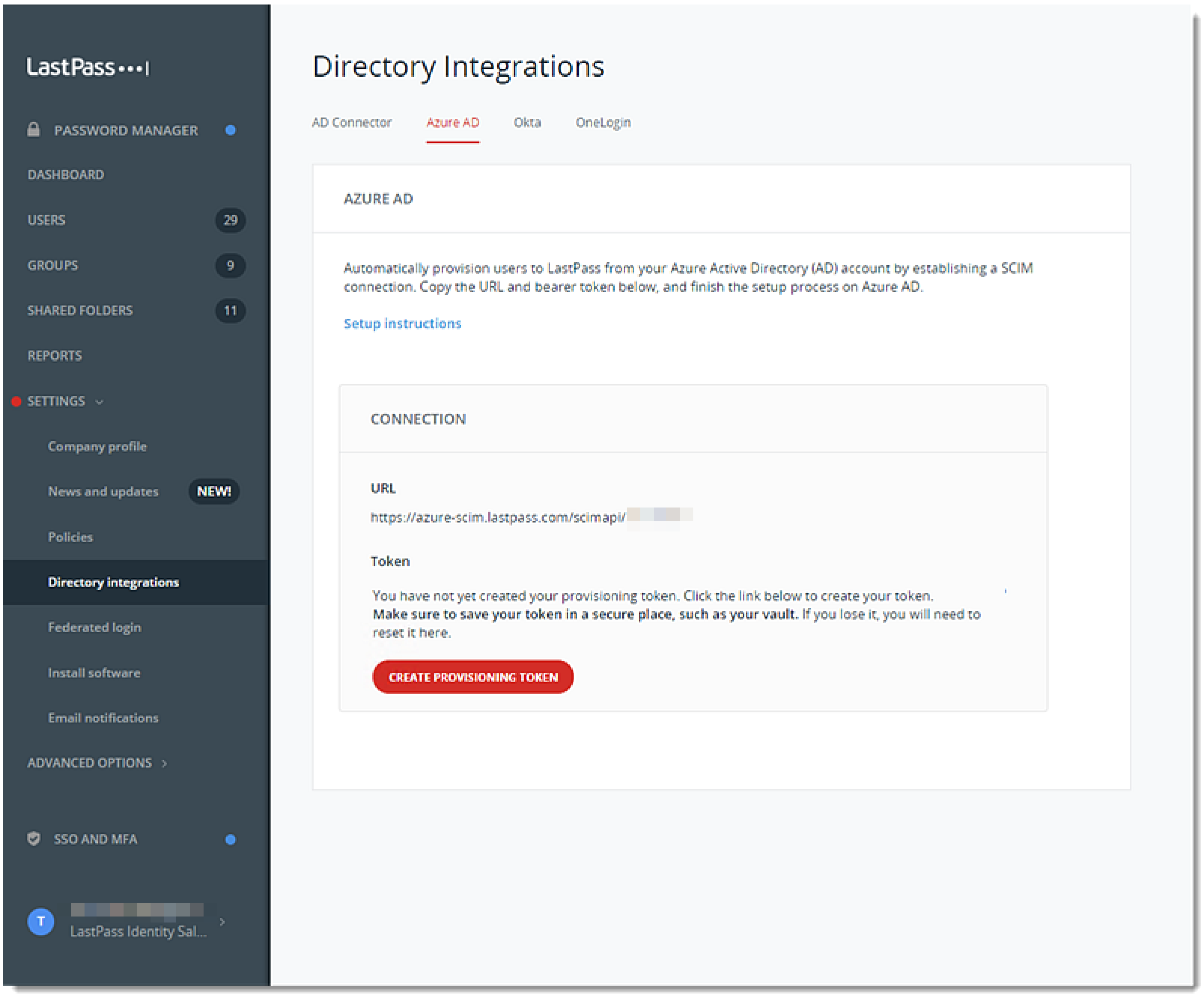
Before you begin…

- It is **required** that you enable the “Permit super admins to reset Master Passwords” policy for at least 1 LastPass admin (who is also a non-federated admin) in the LastPass Admin Console. This ensures that all LastPass user accounts can still be recovered (via Master Password reset) if a critical setting is misconfigured or changed for federated login after setup is complete.
- It is helpful to open a text editor application so that you can copy and paste values that will be used between your LastPass Admin Console and the Azure AD Admin portal.

Step #1: Generate a Provisioning Token

1. Access the LastPass Admin Console by opening a web browser and navigating to either of the following:
   - For accounts using US data centers: https://lastpass.com/company/#!/dashboard
   - For accounts using EU data centers: https://lastpass.eu/company/#!/dashboard
2. Enter your administrator username and Master Password, then click **Log In**.
3. Select **Settings > Directory integrations** in the left navigation.
4. Click on the **Azure AD** tab.
5. Under Connection, copy the *URL* and paste it into your text editor application.
6. Click the **Create Provisioning Token** to generate it, then copy the *Token* and paste it into your text editor application.

   Important: *If you navigate away from the Azure AD tab within the Directory Integrations page, the Provisioning Token will no longer be accessible through the LastPass Admin Console. If the Token is lost, a new one can be generated, but this will invalidate the previous code. Any process that used the old Token will need to be updated with the new one. A new Provisioning Token can be generated by navigating back to the Azure AD tab and clicking* ***Reset Provisioning Token****.*



Directory Integrations

AZURE AD

Automatically provision users to LastPass from your Azure Active Directory (AD) account by establishing a SCIM connection. Copy the URL and bearer token below, and finish the setup process on Azure AD.

Setup instructions

CONNECTION

URL

https://azure-scim.lastpass.com/scimapi/[redacted]

Token

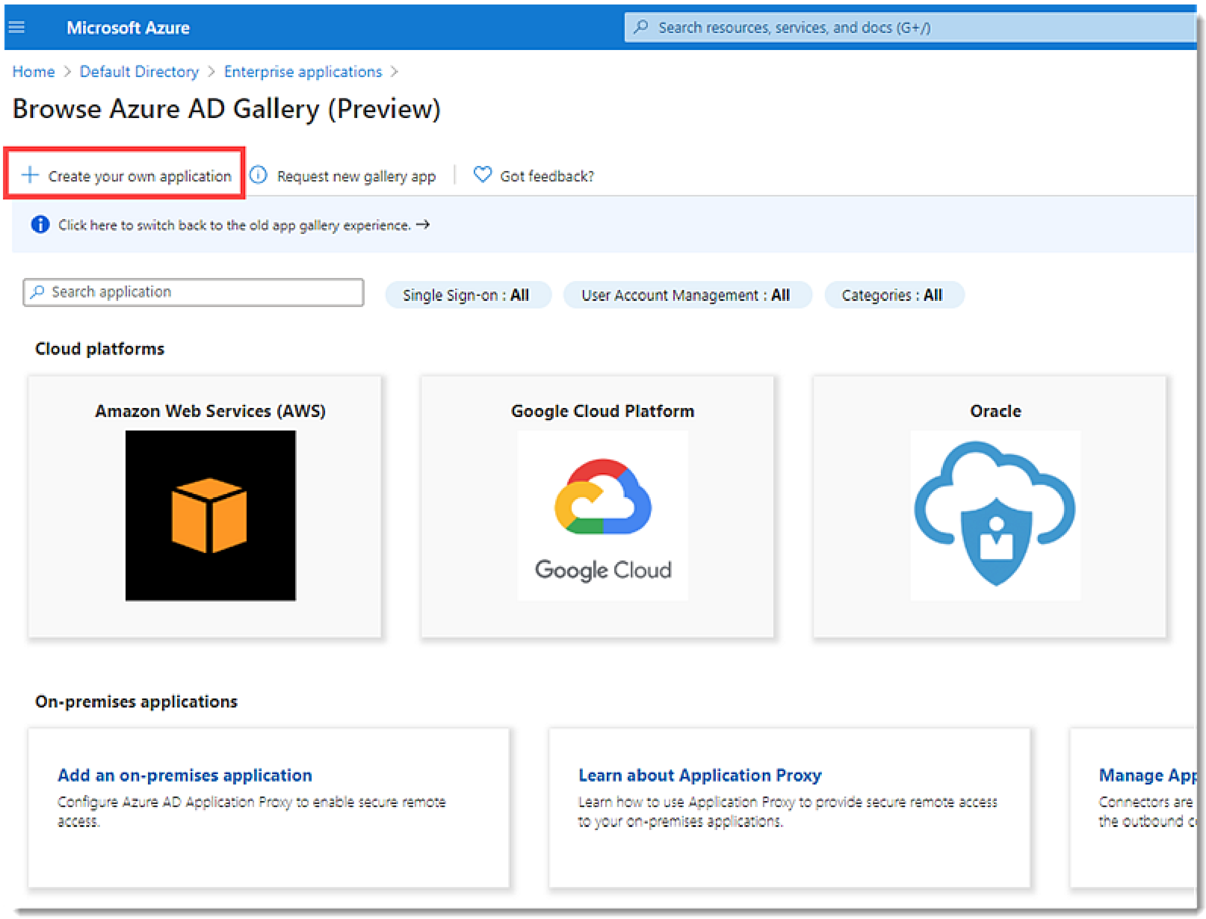
You have not yet created your provisioning token. Click the link below to create your token. **Make sure to save your token in a secure place, such as your vault.** If you lose it, you will need to reset it here.

[redacted]

Step #2: Configure Azure AD with LastPass

Once you have acquired the URL and Provisioning Token, you will need to enter them into the Azure AD Admin portal.

1. Log in to your Azure AD portal with your administrator account credentials at https://portal.azure.com.
2. Navigate to **Azure Active Directory** > **Enterprise Applications** > **New application**.
3. Click **Create Your Own Application**.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Default Directory > Enterprise applications >
Browse Azure AD Gallery (Preview)
+ Create your own application
Request new gallery app
Got feedback?
Search application
Cloud platforms
Amazon Web Services (AWS)
Google Cloud Platform
Google Cloud
Oracle
On-premises applications

Click here to switch back to the old app gallery experience. →

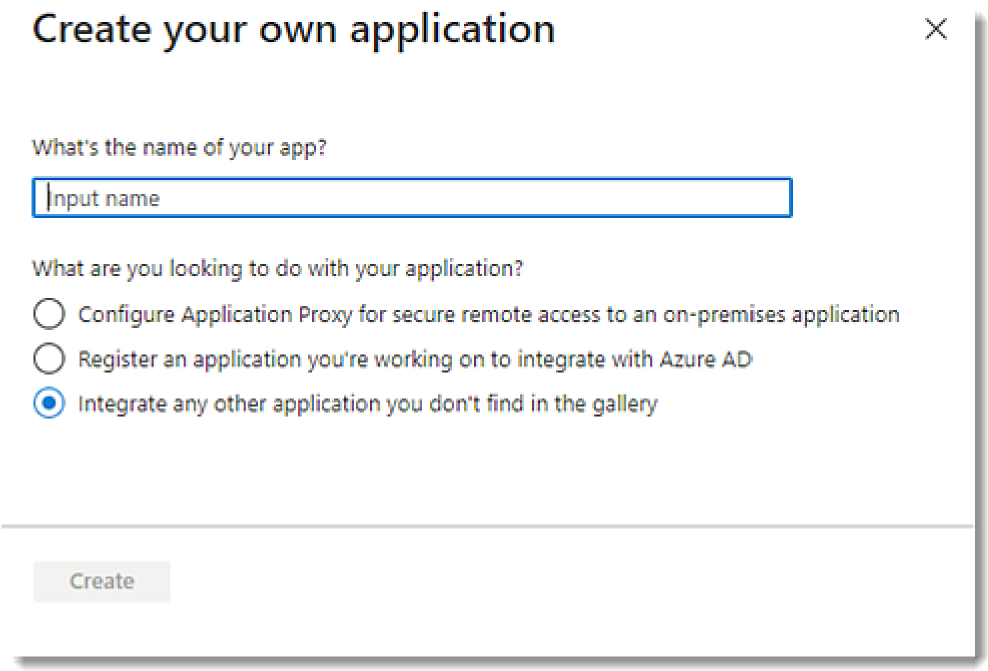
Single Sign-on : **All**   User Account Management : **All**   Categories : **All**

Add an on-premises application
Configure Azure AD Application Proxy to enable secure remote access.

Learn about Application Proxy
Learn how to use Application Proxy to provide secure remote access to your on-premises applications.

Manage App
Connectors are
the outbound c

4. Enter a name for your application (LastPass) and click **Create** to create an app object. The application object created is intended to represent the target app (for which you would be provisioning and setting up single sign-on, not just as the SCIM endpoint).
5. Select the radio button for the **Integrate any other application you don't find in the gallery** option.



Create your own application

What's the name of your app?

Input name

What are you looking to do with your application?

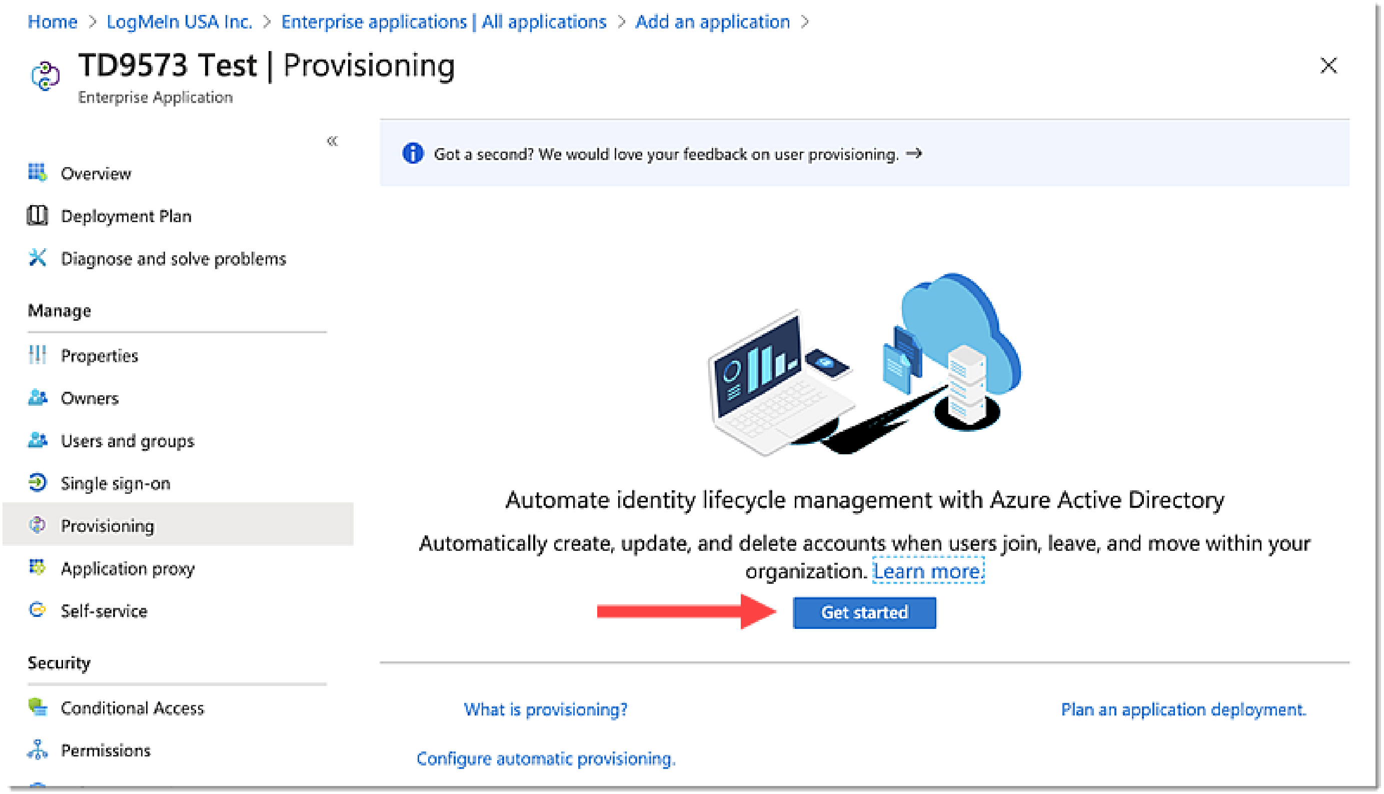
○ Configure Application Proxy for secure remote access to an on-premises application

○ Register an application you're working on to integrate with Azure AD

● Integrate any other application you don't find in the gallery

Create

6. Select the **Provisioning** tab in the left navigation.
7. Click **Get started**.



Home > LogMeIn USA Inc. > Enterprise applications | All applications > Add an application >

TD9573 Test | Provisioning

Enterprise Application

- Overview
- Deployment Plan
- Diagnose and solve problems

Manage

- Properties
- Owners
- Users and groups
- Single sign-on
- Provisioning
- Application proxy
- Self-service

Security

- Conditional Access
- Permissions

Got a second? We would love your feedback on user provisioning. →

Automate identity lifecycle management with Azure Active Directory

Automatically create, update, and delete accounts when users join, leave, and move within your organization. Learn more

Get started

What is provisioning?

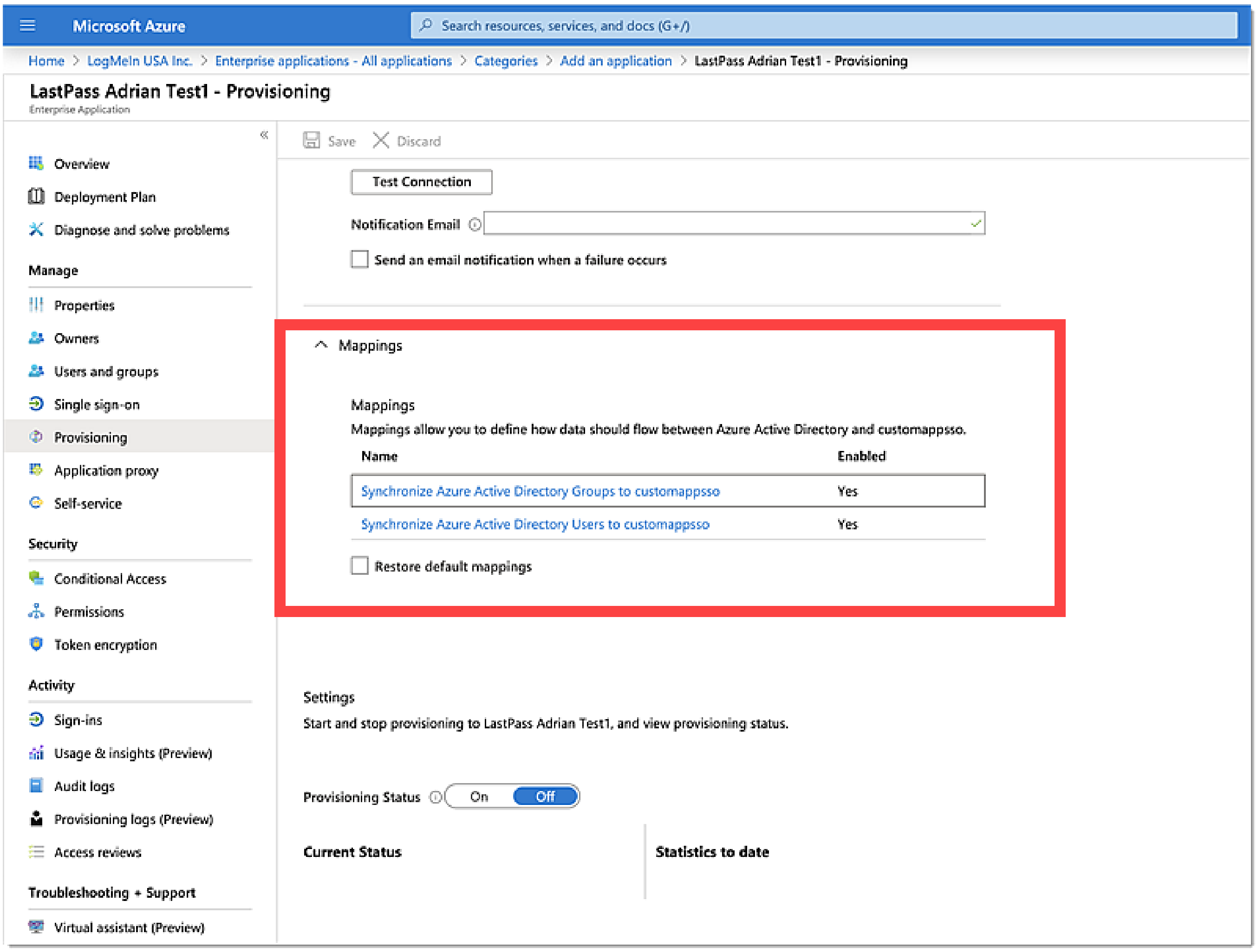
Plan an application deployment.

Configure automatic provisioning.

8. For Provisioning Mode, use the drop-down menu and select **Automatic**.
9. Under Admin Credentials, enter the following:
   a. Locate the “Tenant URL” field and paste the **URL** you copied from the LastPass Admin Console.

   b. Locate the “Secret Token” field and paste the Provisioning **Token** you copied from the LastPass Admin Console.

10. Click **Test Connection** to have Azure AD attempt to connect to the SCIM endpoint. If the attempts fail, error information is displayed.
11. If the connection test succeeds, click **Save** to store the admin credentials.
12. Next, select **Mappings**.
13. Click **Synchronize Azure Active Directory Users to <app name>** to modify user object mappings.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > LogMeIn USA Inc. > Enterprise applications - All applications > Categories > Add an application > LastPass Adrian Test1 - Provisioning
LastPass Adrian Test1 - Provisioning
Enterprise Application
Overview
Deployment Plan
Diagnose and solve problems
Manage
Properties
Owners
Users and groups
Single sign-on
Provisioning
Application proxy
Self-service
Security
Conditional Access
Permissions
Token encryption
Activity
Sign-ins
Usage & insights (Preview)
Audit logs
Provisioning logs (Preview)
Access reviews
Troubleshooting + Support
Virtual assistant (Preview)
Save
Discard
Test Connection

Notification Email

□ Send an email notification when a failure occurs

Mappings

Mappings

Mappings allow you to define how data should flow between Azure Active Directory and customappsso.

Name	Enabled
Synchronize Azure Active Directory Groups to customappsso	Yes
Synchronize Azure Active Directory Users to customappsso	Yes

□ Restore default mappings

Settings

Start and stop provisioning to LastPass Adrian Test1, and view provisioning status.

Provisioning Status: On [✓ Off]

Current Status

Statistics to date

14. Modify the User Attribute Mappings, as follows:

   a) **ExternalID** – Use the objectID attribute from Azure AD and set this as a matching attribute with Precedence set as **1**.

      - ***Note:*** *This should be the only mapping with any Precedence set. In order to change the ExternalID Precedence to 1, you may need to modify another attribute that might already have a Precedence set to 1.* ***After you find such attribute, you can change its precedence from 1 to 2, then go back to ExternalID and set its Precedence to 1****. Finally, to remove the Precedence entirely from the other attribute (now set to 2), you can now edit it once again and set the ”Match objects using this attribute” to* ***No****.*

   b) **Active** – The default Azure AD mapping can be used, or a custom one which will be used to set the user as enabled/disabled in

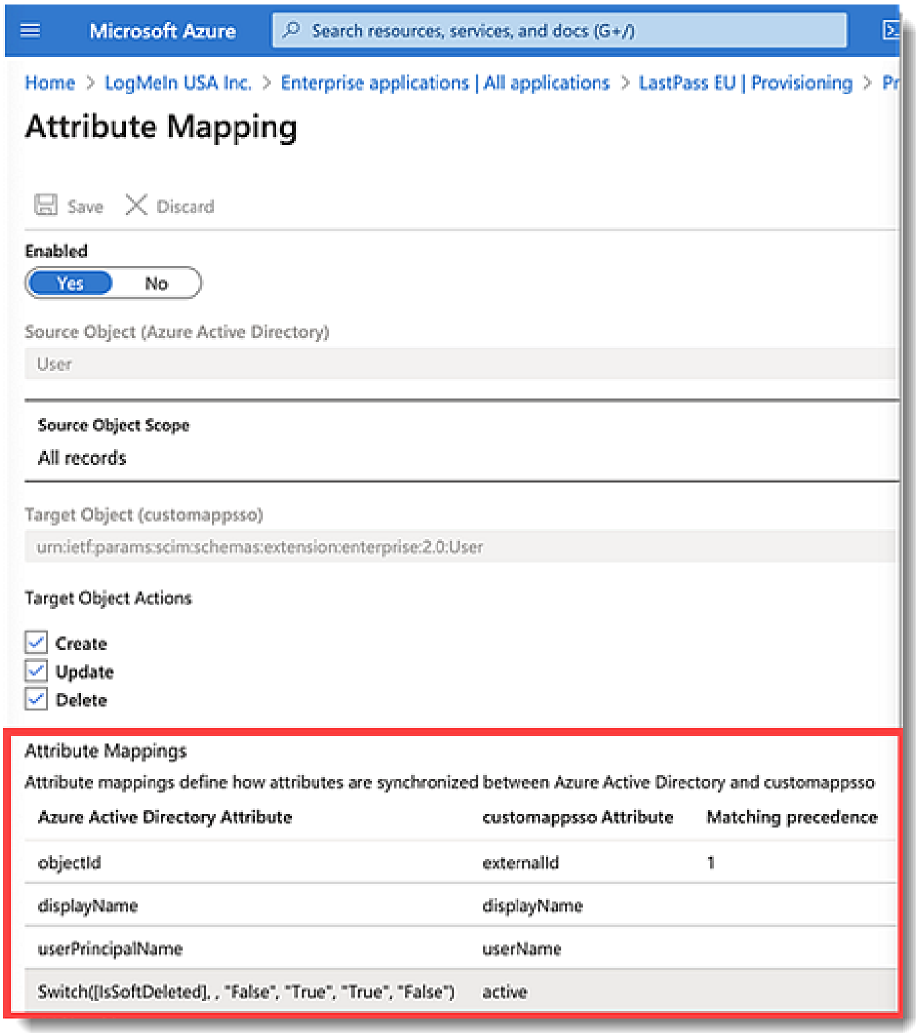
LastPass.

c) **DisplayName** – Use any property from Azure AD. This should be a string which will be the synchronized user's name in LastPass.

d) **UserName** – Map the user's email address from Azure AD. Please note that the userPrincipalName might not be equal to the email address. In this case, use an attribute from Azure AD which contains the email address the user will utilize and can read (e.g., Mail or in most cases, userPrincipalName should be fine).

Important: *If you already have users in LastPass, their email address **MUST** match the Azure AD attribute mapped to the userName value.* ***If this is not mapped correctly, a duplicate user will be created for every existing user in LastPass.***

Important: *Only the 4 mappings (shown below) should be present after editing, and must be configured correctly. You **MUST** delete all extra mappings except for the ones listed below, otherwise you will encounter synchronization issues.*



Home > LogMeIn USA Inc. > Enterprise applications | All applications > LastPass EU | Provisioning > Pr

Attribute Mapping

Enabled

● Yes ○ No

Source Object (Azure Active Directory)

User

Source Object Scope

All records

Target Object (customappsso)

urn:ietf:params:scim:schemas:extension:enterprise:2.0:User

Target Object Actions

☑ Create

☑ Update

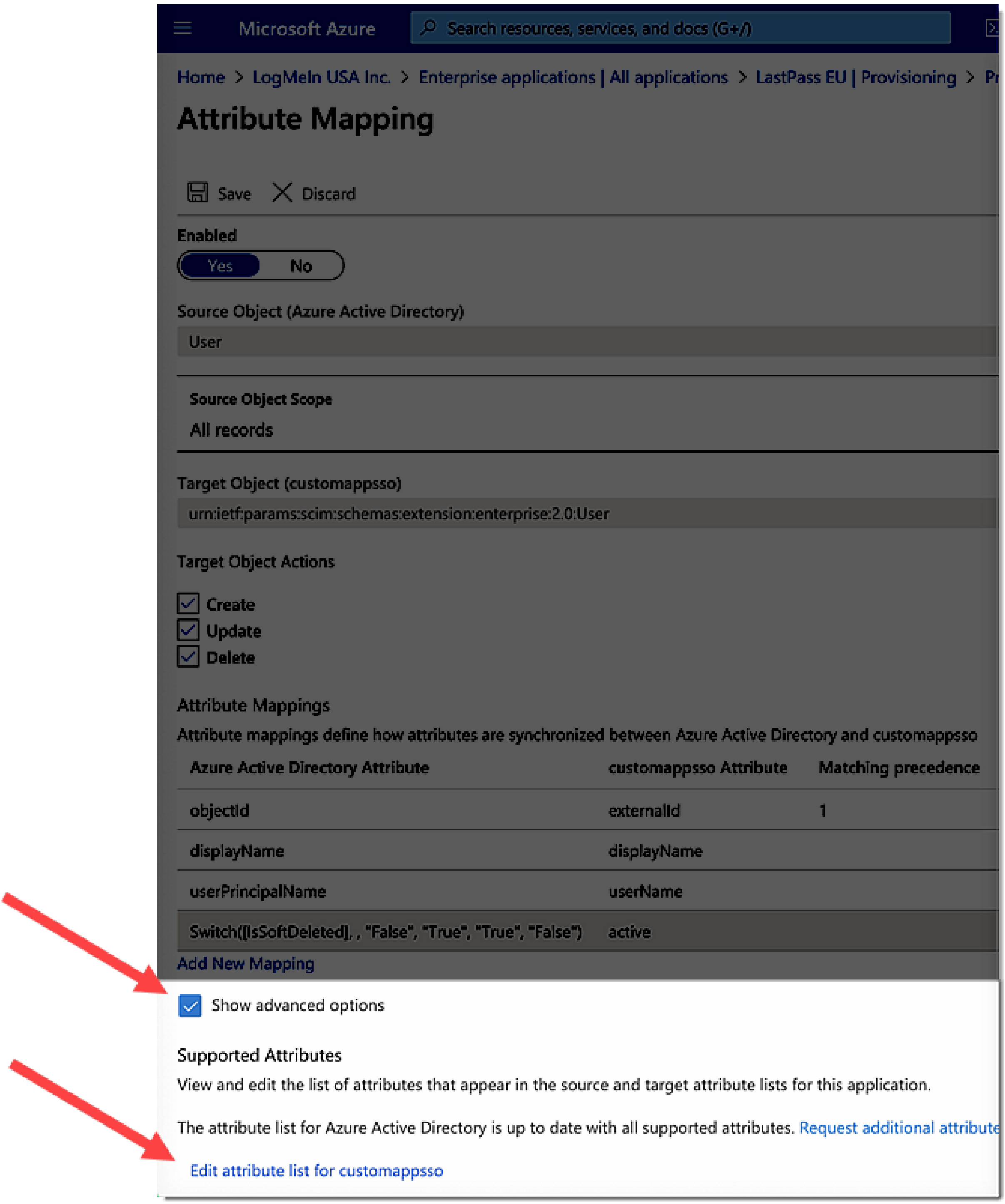
☑ Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

Azure Active Directory Attribute	customappsso Attribute	Matching precedence
objectId	externalId	1
displayName	displayName	
userPrincipalName	userName	
Switch([IsSoftDeleted], , "False", "True", "True", "False")	active	

15. Modify the Attribute List, as follows:
    a. Check the box for **Show advanced options** at the bottom of Attribute Mapping.
    b. Click **Edit attribute list for <app name>**.



Home > LogMeIn USA Inc. > Enterprise applications | All applications > LastPass EU | Provisioning > Pr

Attribute Mapping

Enabled

Source Object (Azure Active Directory)

User

Source Object Scope

All records

Target Object (customappsso)

urn:ietf:params:scim:schemas:extension:enterprise:2.0:User

Target Object Actions

☑ Create
☑ Update
☑ Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

Azure Active Directory Attribute	customappsso Attribute	Matching precedence
objectId	externalId	1
displayName	displayName	
userPrincipalName	userName	
Switch([IsSoftDeleted], , "False", "True", "True", "False")	active	

Add New Mapping

☑ Show advanced options

Supported Attributes

View and edit the list of attributes that appear in the source and target attribute lists for this application.

The attribute list for Azure Active Directory is up to date with all supported attributes. Request additional attribute

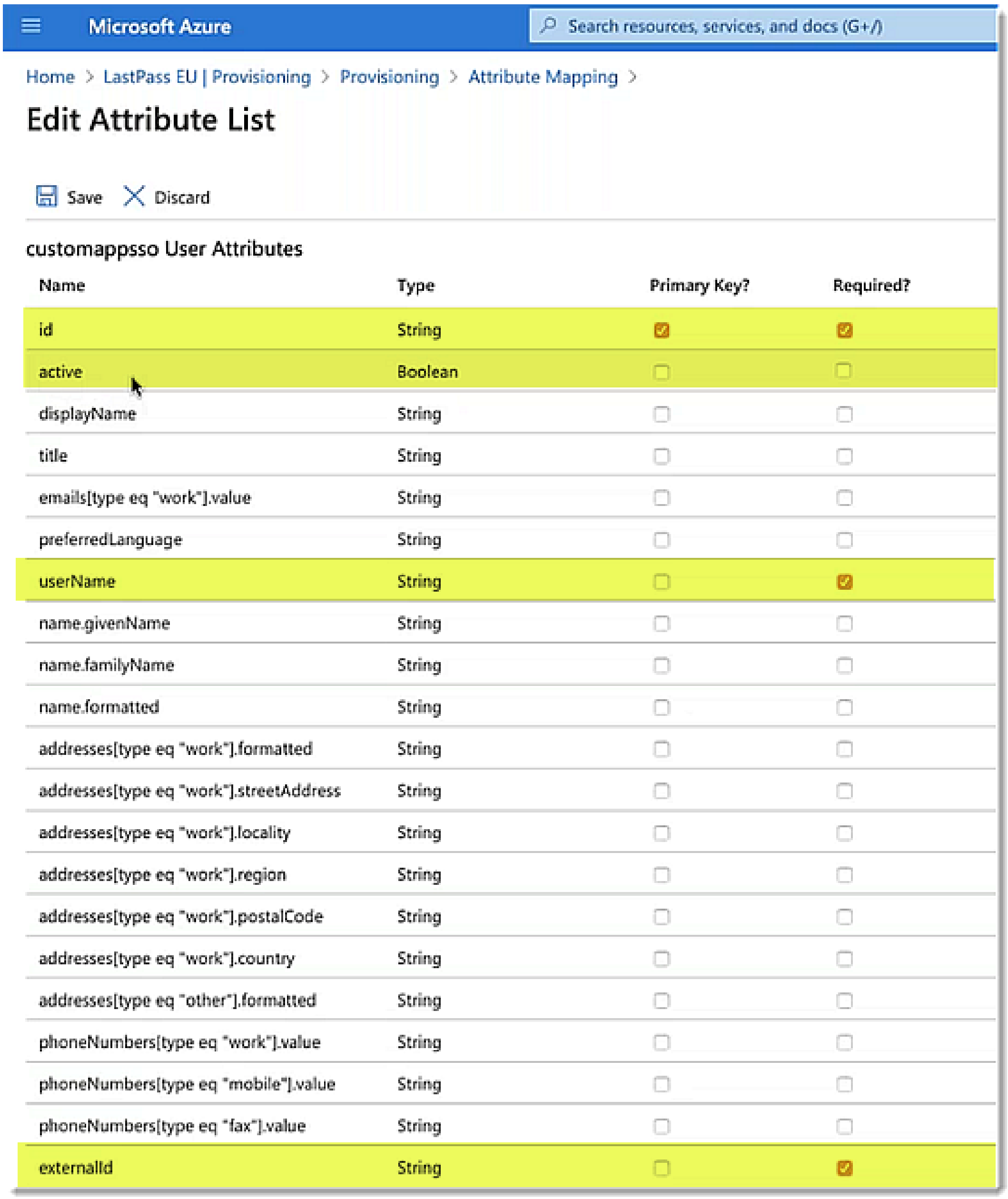
Edit attribute list for customappsso

c. In the **Edit Attribute List**, make the following selections:
- Name = id, Type = String – Check the boxes for **Primary** and **Required**
- Name = active, Type **=** Boolean **–** <u>Do not check the box for Required</u> *** ***Please see steps below if the "Active"attribute is not listed***
- Name = userName, Type = String – Check the box for **Required**
- Name = externalID, Type = String – Check the box for

Required

d. Click **Save** and return to Attribute Mapping.

Important: *Only the 4 mappings (shown below) should be present after editing, and must be configured correctly. You **MUST** delete all extra mappings except for the ones listed below, otherwise you will encounter synchronization issues.*


Microsoft Azure
Search resources, services, and docs (G+/)
Home > LastPass EU | Provisioning > Provisioning > Attribute Mapping >
Edit Attribute List
Save
Discard
customappsso User Attributes

Name	Type	Primary Key?	Required?
id	String	☑	☑
active	Boolean	□	□
displayName	String	□	□
title	String	□	□
emails[type eq "work"].value	String	□	□
preferredLanguage	String	□	□
userName	String	□	☑
name.givenName	String	□	□
name.familyName	String	□	□
name.formatted	String	□	□
addresses[type eq "work"].formatted	String	□	□
addresses[type eq "work"].streetAddress	String	□	□
addresses[type eq "work"].locality	String	□	□
addresses[type eq "work"].region	String	□	□
addresses[type eq "work"].postalCode	String	□	□
addresses[type eq "work"].country	String	□	□
addresses[type eq "other"].formatted	String	□	□
phoneNumbers[type eq "work"].value	String	□	□
phoneNumbers[type eq "mobile"].value	String	□	□
phoneNumbers[type eq "fax"].value	String	□	□
externalId	String	□	☑

****** If the "Active" attribute is not listed under Attribute Mappings, you can create the mapping by doing the following:***

i. Under "Supported Attributes" click **Edit attribute list for <app name>**.

ii. Within the "Edit Attribute List" page, at the bottom of the list of attributes, type **Active** in the first empty field, then use the next field's drop-down menu and select **Boolean**.

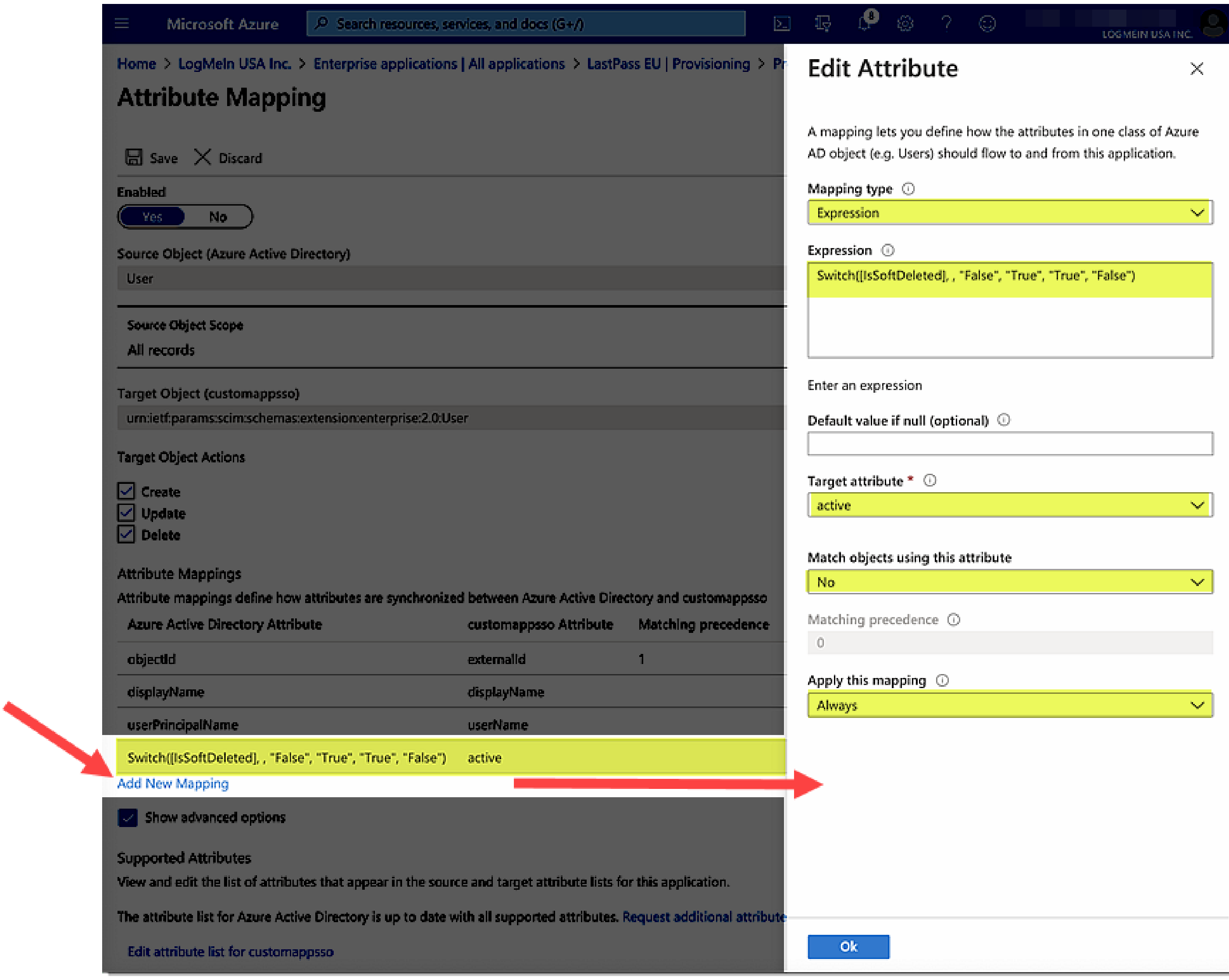
iii. Click **Add Attribute,** then click **Save**.

iv. Back on the Attribute Mapping page, below your existing user attributes, click **Add New Mapping**.

v. On the Edit Attribute card in the right navigation, enter the following information:

- Mapping type = **Expression**
- Expression = **`Switch([IsSoftDeleted], , "False", "True", "True", "False")`**
- Target attribute = **active**
- Match objects using this attribute = **No**
- Apply this mapping = **Always**

vi. Click **OK**.



Home > LogMeIn USA Inc. > Enterprise applications | All applications > LastPass EU | Provisioning > Pr

Attribute Mapping

Enabled: Yes No

Source Object (Azure Active Directory): User

Source Object Scope: All records

Target Object (customappsso): urn:ietf:params:scim:schemas:extension:enterprise:2.0:User

Target Object Actions

☑ Create
☑ Update
☑ Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

Azure Active Directory Attribute	customappsso Attribute	Matching precedence
objectId	externalId	1
displayName	displayName	
userPrincipalName	userName	
Switch([IsSoftDeleted], , "False", "True", "True", "False")	active	

Add New Mapping

☑ Show advanced options

Supported Attributes

View and edit the list of attributes that appear in the source and target attribute lists for this application.

The attribute list for Azure Active Directory is up to date with all supported attributes. Request additional attribute

Edit attribute list for customappsso

Edit Attribute

A mapping lets you define how the attributes in one class of Azure AD object (e.g. Users) should flow to and from this application.

Mapping type: Expression

Expression: Switch([IsSoftDeleted], , "False", "True", "True", "False")

Enter an expression

Default value if null (optional):

Target attribute *: active

Match objects using this attribute: No

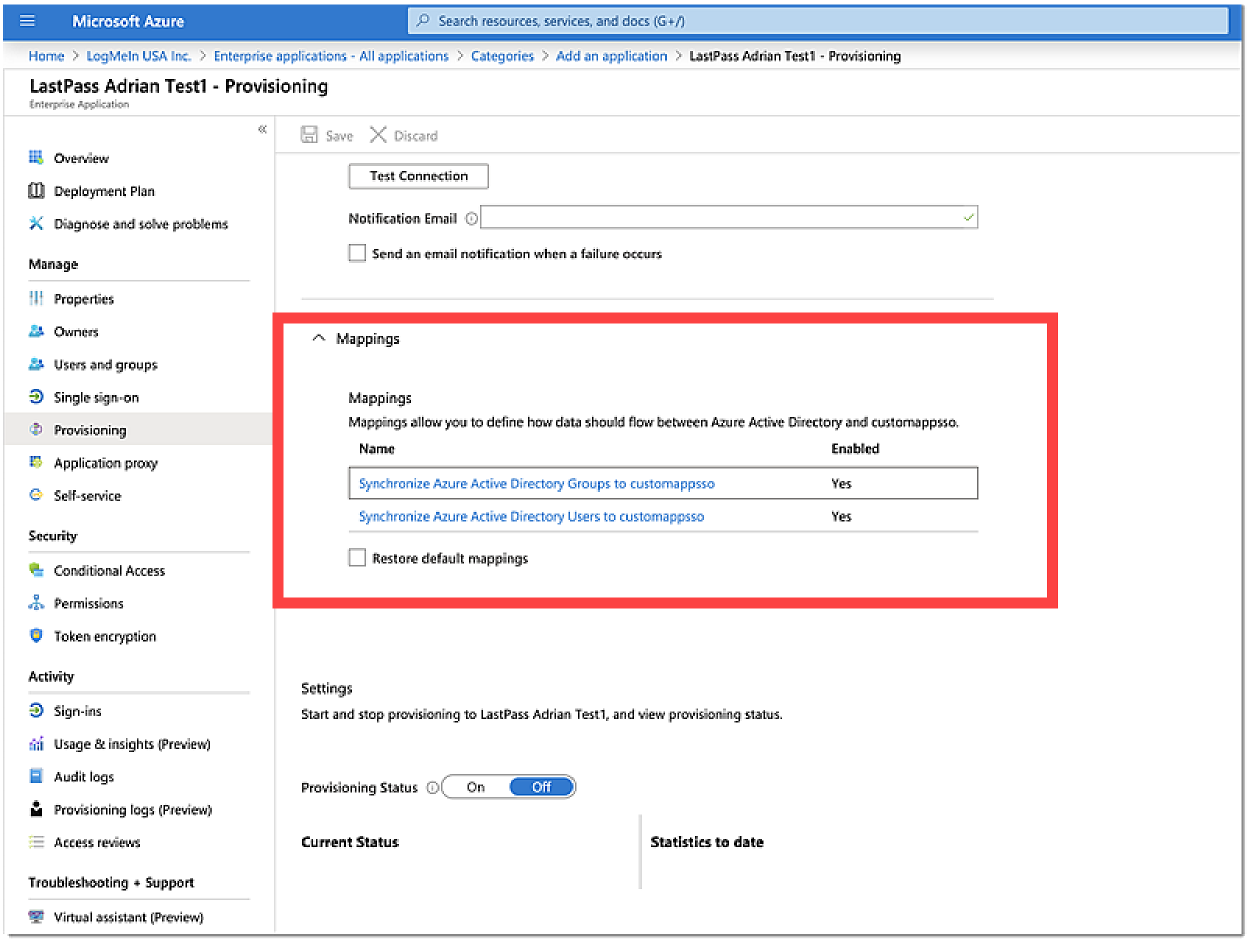
Matching precedence: 0

Apply this mapping: Always

16. Under the "Attribute Mapping" section, Azure may have created mappings already, but those can be modified and/or deleted.

*Important: Only the 4 mappings should be present after editing, and must be configured correctly. You **MUST** delete all extra mappings except for the ones listed below, otherwise you will encounter synchronization issues.*

17. Click **Save**, then return to the Provisioning settings and select **Mappings** (from **Step #10** above).
18. Click **Synchronize Azure Active Directory Groups to <app name>** to modify group object mappings.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > LogMeIn USA Inc. > Enterprise applications - All applications > Categories > Add an application > LastPass Adrian Test1 - Provisioning
Overview
Deployment Plan
Diagnose and solve problems
Manage
Properties
Owners
Users and groups
Single sign-on
Provisioning
Application proxy
Self-service
Security
Conditional Access
Permissions
Token encryption
Activity
Sign-ins
Usage & insights (Preview)
Audit logs
Provisioning logs (Preview)
Access reviews
Troubleshooting + Support
Virtual assistant (Preview)
Save
Discard
Test Connection

LastPass Adrian Test1 - Provisioning
Enterprise Application

Notification Email

□ Send an email notification when a failure occurs

Mappings

Mappings

Mappings allow you to define how data should flow between Azure Active Directory and customappsso.

Name	Enabled
Synchronize Azure Active Directory Groups to customappsso	Yes
Synchronize Azure Active Directory Users to customappsso	Yes

□ Restore default mappings

Settings

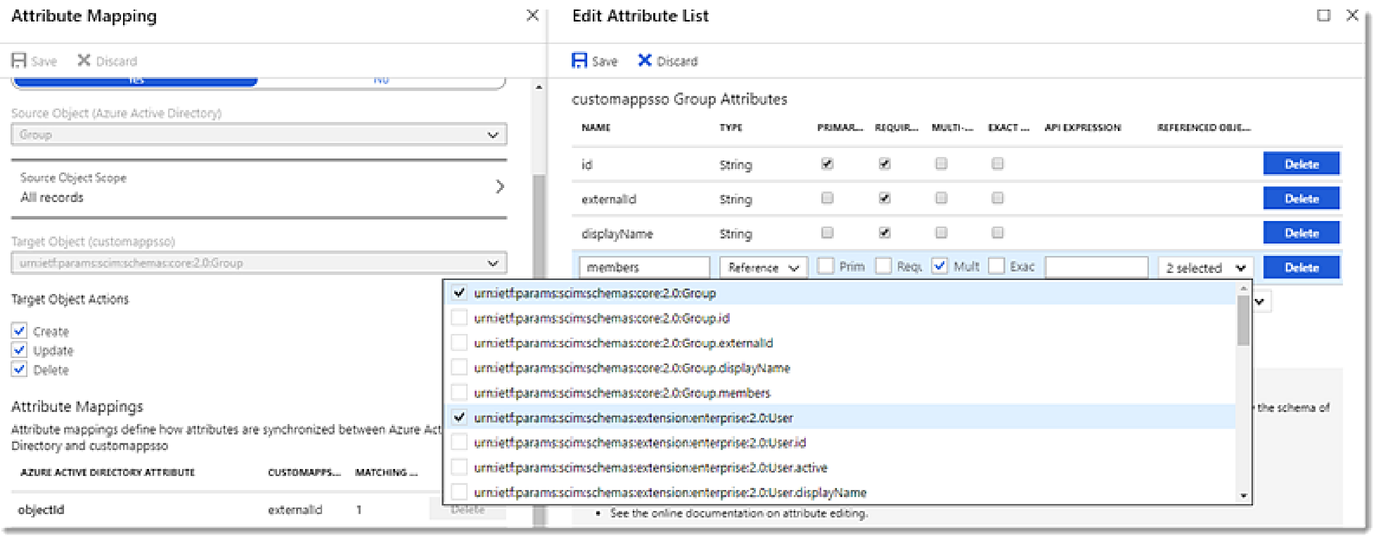
Start and stop provisioning to LastPass Adrian Test1, and view provisioning status.

Provisioning Status: On [✓ Off]

Current Status

Statistics to date

19. Next, modify group object mappings as follows:
    a) Check the box for **Show advanced options** at the bottom of Attribute Mapping.
    b) Click **Edit attribute list for <app name>**.
    c) In the Edit Attribute List, make the following selections:
        - Name = id, Type = String – Check the boxes for **Primary** and **Required**
        - Name = externalID, Type = String – Check the box for **Required**
        - Name = displayName, Type = String – Check the box for **Required**
        - Name = members, Type = Reference – Check the box for **Multi-Valued**, then set referenced objects for:
            - **urn:ietf:params:scim:schemas:core:2.0:Group**
            - **urn:ietf:params:scim:schemas:extension:enterprise:2.0:User**
    d) Click **Save** and return to Attribute Mapping.



Source Object (Azure Active Directory): Group

Source Object Scope: All records

Target Object (customappsso): urn:ietf:params:scim:schemas:core:2.0:Group

Target Object Actions

- ☑ Create
- ☑ Update
- ☑ Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Act Directory and customappsso

AZURE ACTIVE DIRECTORY ATTRIBUTE	CUSTOMAPPS...	MATCHING ...	
objectId	externalId	1	Delete

customappsso Group Attributes

NAME	TYPE	PRIMAR...	REQUIR...	MULTI-...	EXACT ...	API EXPRESSION	REFERENCED OBJE...	
id	String	☑	☑	□	□			Delete
externalId	String	□	☑	□	□			Delete
displayName	String	□	☑	□	□			Delete
members	Reference	□ Prim	□ Requ	☑ Mult	□ Exac		2 selected	Delete

- ☑ urn:ietf:params:scim:schemas:core:2.0:Group
- □ urn:ietf:params:scim:schemas:core:2.0:Group.id
- □ urn:ietf:params:scim:schemas:core:2.0:Group.externalId
- □ urn:ietf:params:scim:schemas:core:2.0:Group.displayName
- □ urn:ietf:params:scim:schemas:core:2.0:Group.members
- ☑ urn:ietf:params:scim:schemas:extension:enterprise:2.0:User
- □ urn:ietf:params:scim:schemas:extension:enterprise:2.0:User.id
- □ urn:ietf:params:scim:schemas:extension:enterprise:2.0:User.active
- □ urn:ietf:params:scim:schemas:extension:enterprise:2.0:User.displayName

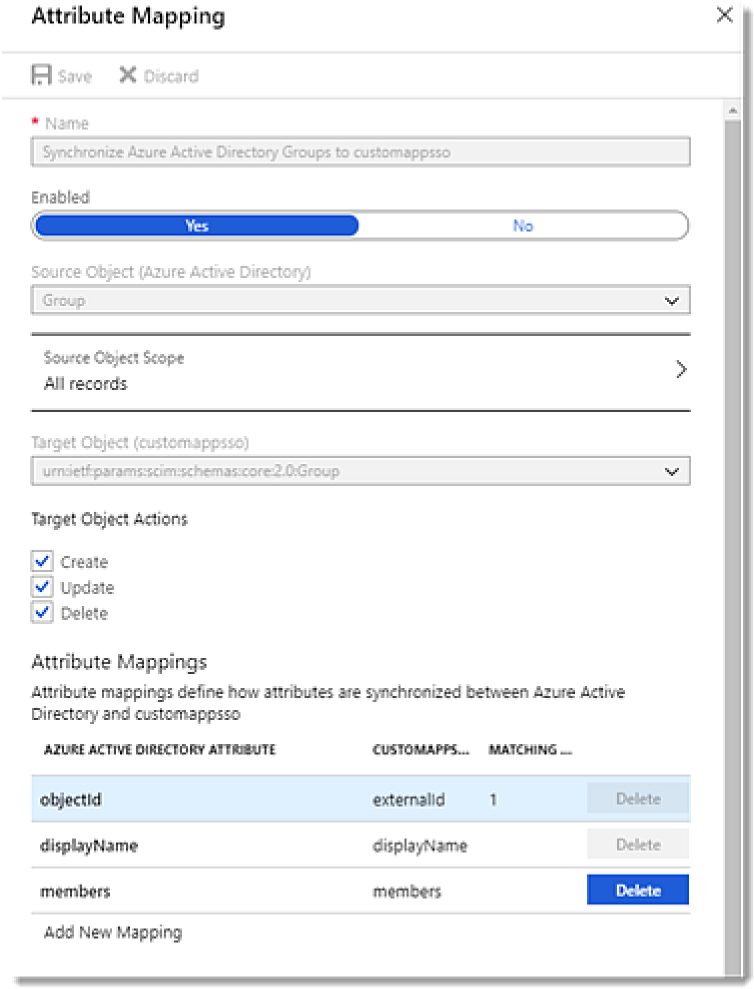
the schema of

- See the online documentation on attribute editing.

20. Under the ”Attribute Mapping” section, Azure may have created mappings already, but those can be modified and/or deleted.

    Important: *Only the required 3 mappings should be present after editing, and must be configured correctly. You* ***MUST*** *delete all extra mappings except for the ones listed below, otherwise you will encounter synchronization issues.*

21. Modify the Group Attribute Mapping rules as follows:
    a) **ExternalID** – Use the objectID attribute from Azure AD and set this as a matching attribute with Precedence set as **1**. This should be the only mapping with any Precedence set.
    b) **DisplayName** – Use any attribute for group name.
    c) **Members** – User members from Azure AD.



Attribute Mapping

Save Discard

* Name

Synchronize Azure Active Directory Groups to customappsso

Enabled

Yes No

Source Object (Azure Active Directory)

Group

Source Object Scope

All records

Target Object (customappsso)

urn:ietf:params:scim:schemas:core:2.0:Group

Target Object Actions

☑ Create

☑ Update

☑ Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

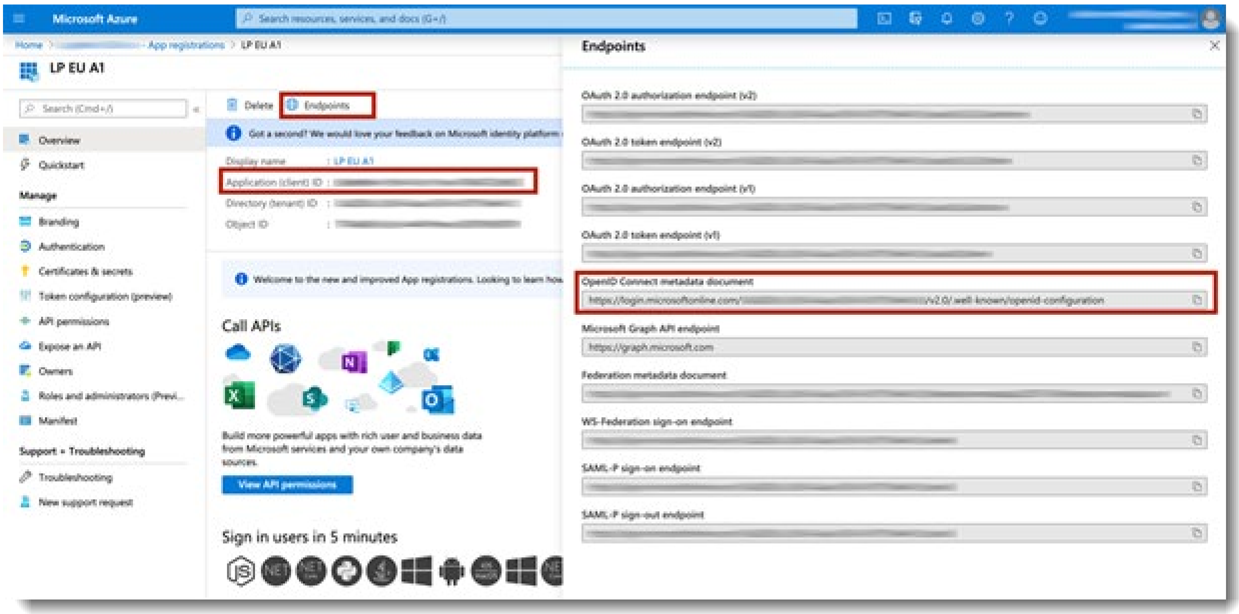
AZURE ACTIVE DIRECTORY ATTRIBUTE	CUSTOMAPPS...	MATCHING ...	
objectId	externalId	1	Delete
displayName	displayName		Delete
members	members		Delete

Add New Mapping

22. Click **Save**, then return to the Provisioning settings.
23. Under Settings, the Scopefield defines which users and or groups are synchronized. Selecting **Sync only assigned users and groups** (recommended) will only sync users and/or groups assigned in the Users and groups tab.
    - ***Note:*** *If syncing only assigned users and groups (recommended), be sure to select the* ***Users and groups*** *tab and assign the users and/or groups you wish to sync.*
24. Once your configuration is complete, enable the Provisioning Status by clicking **On**.
25. Click **Save** to start the Azure AD provisioning service.

Step #3: Capture the Application ID and OpenID Connect from Azure AD

1. In the Azure AD portal, navigate to your directory.
2. Select **App registrations** in the left navigation, then select **the name of your app**. You will then be redirected to the **Overview** page.
3. With **Overview** selected in the left navigation, copy the *Application (client) ID* field contents and paste it into your open text editor.
4. Click **Endpoints** in the top navigation to expand the menu on the right.
5. Copy *OpenID Connect metadata document* field contents and paste it into your open text editor.
6. Proceed to the next step below where these items will be used.



Got a second? We would love your feedback on Microsoft identity platform

Display name : LP EU A1

Application (client) ID : [redacted]

Directory (tenant) ID : [redacted]

Object ID : [redacted]

Welcome to the new and improved App registrations. Looking to learn how

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

OAuth 2.0 authorization endpoint (v2): [redacted]

OAuth 2.0 token endpoint (v2): [redacted]

OAuth 2.0 authorization endpoint (v1): [redacted]

OAuth 2.0 token endpoint (v1): [redacted]

OpenID Connect metadata document: https://login.microsoftonline.com/[redacted]/v2.0/.well-known/openid-configuration

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

Federation metadata document: [redacted]

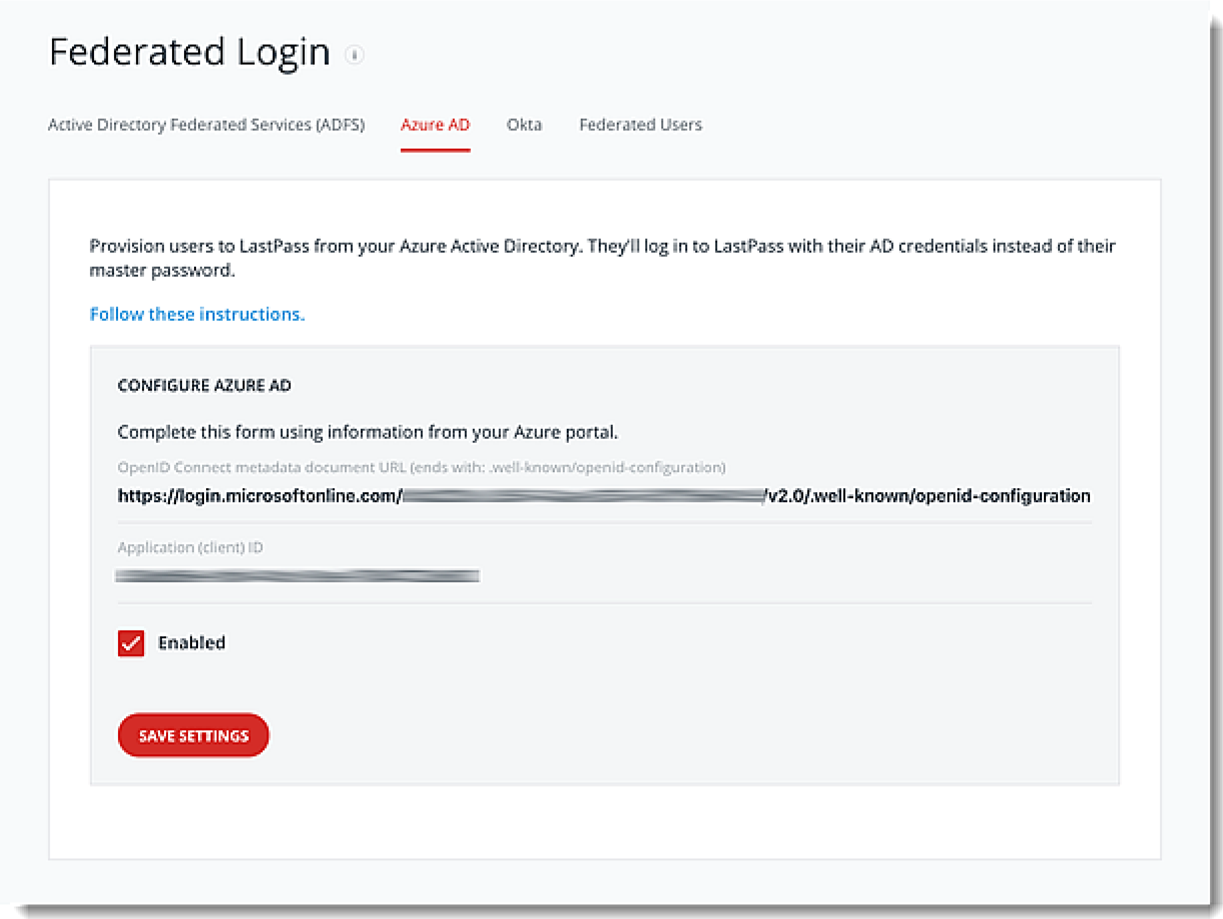
WS-Federation sign-on endpoint: [redacted]

SAML-P sign-on endpoint: [redacted]

SAML-P sign-out endpoint: [redacted]

Step #4: Configure Federated login settings in LastPass

1. Go back to the LastPass Admin Console, then select **Settings** > **Federated login** in the left navigation.
2. Select the **Azure AD** tab, then enter the following:
    - In the "Directory (tenant) ID" field, paste the **OpenID Connect metadata document** from **Step #3** (in the previous section).
    - In the "Application (client) ID" field, paste the **Application (client) ID** from **Step #3** (in the previous section).
3. Check the box for **Enabled**.
4. Click **Save Settings** when finished.



Federated Login

Provision users to LastPass from your Azure Active Directory. They'll log in to LastPass with their AD credentials instead of their master password.

Follow these instructions.

CONFIGURE AZURE AD

Complete this form using information from your Azure portal.

OpenID Connect metadata document URL (ends with: .well-known/openid-configuration)

https://login.microsoftonline.com/[redacted]/v2.0/.well-known/openid-configuration

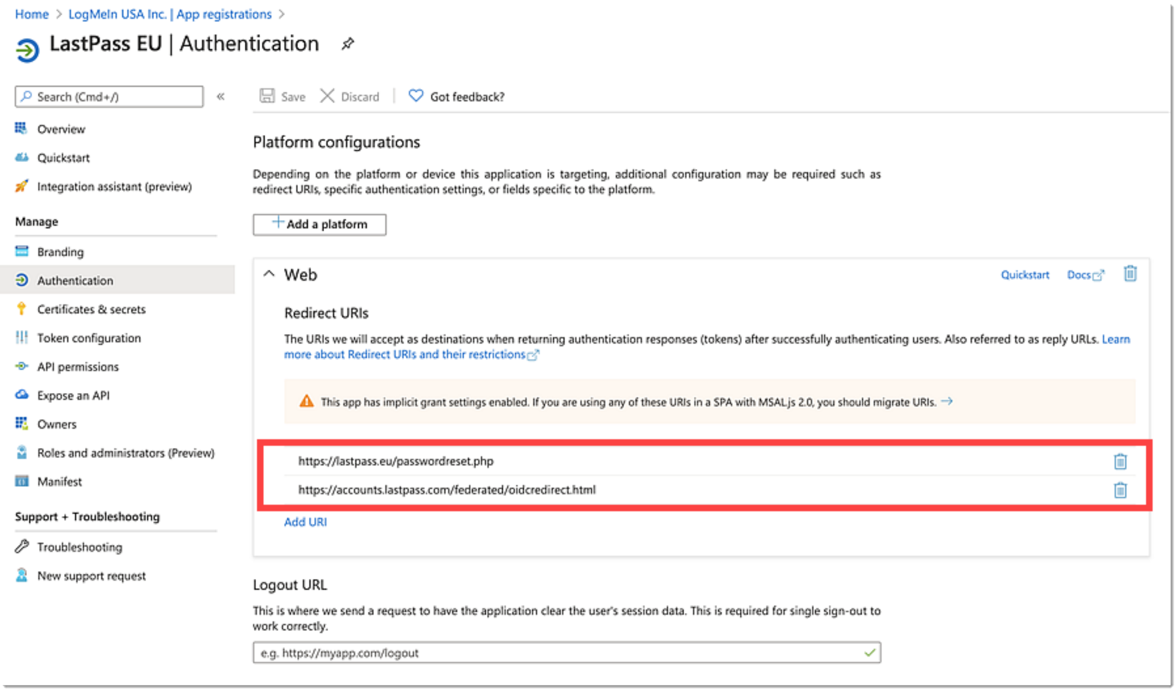
Application (client) ID

[redacted]

☑ Enabled

Step #5: Configure a Redirect URI in Azure AD

1. In the Azure AD portal, select **App Registrations**, then select your **<app name>.**
2. Click **Authentication** in the left navigation.
3. Click **Add a platform**.
4. Select **Web**.
5. Add the first Redirect URI, as follows:
   - For the Type column, use the drop-down menu and select **Web**
   - For the Redirect URI column, enter either of the following:
     i. For accounts using **US data centers**: https://lastpass.com/passwordreset.php
     ii. For accounts using **EU data centers**: https://lastpass.eu/passwordreset.php
6. Click **Configure**.
7. Under the Web section, enter the second Redirect URI as follows:
   - https://accounts.lastpass.com/federated/oidcredirect.html
8. Click anywhere outside of the field to add the second URI.
9. Under the Implicit Grant settings, check the boxes to enable the following settings:
   - **Access tokens**
   - **ID tokens**
10. Click **Save** when finished.



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.

Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating users. Also referred to as reply URLs. Learn more about Redirect URIs and their restrictions

⚠ This app has implicit grant settings enabled. If you are using any of these URIs in a SPA with MSAL.js 2.0, you should migrate URIs. →

https://lastpass.eu/passwordreset.php

https://accounts.lastpass.com/federated/oidcredirect.html

Logout URL

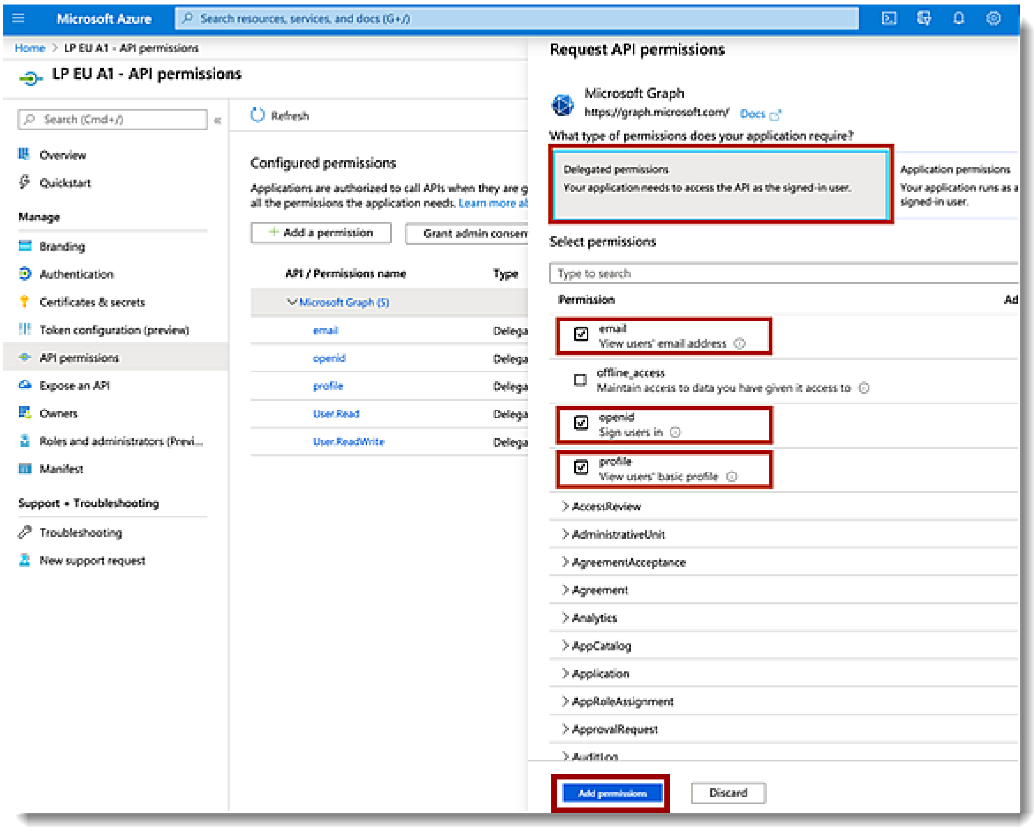
This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

e.g. https://myapp.com/logout ✓

Step #6: Configure API permissions in Azure AD

1. In the Azure AD portal, select **API permissions** in the left navigation.
2. Click the **Add a permission** button, then select **Microsoft Graph**.
3. In the right navigation, select **Delegated permissions**.
4. Under the Permission menu, check the boxes to enable the following permission settings:

- **email**
- **openid**
- **profile**



Applications are authorized to call APIs when they are g all the permissions the application needs. Learn more ab

API / Permissions name	Type
Microsoft Graph (5)	
email	Delega
openid	Delega
profile	Delega
User.Read	Delega
User.ReadWrite	Delega

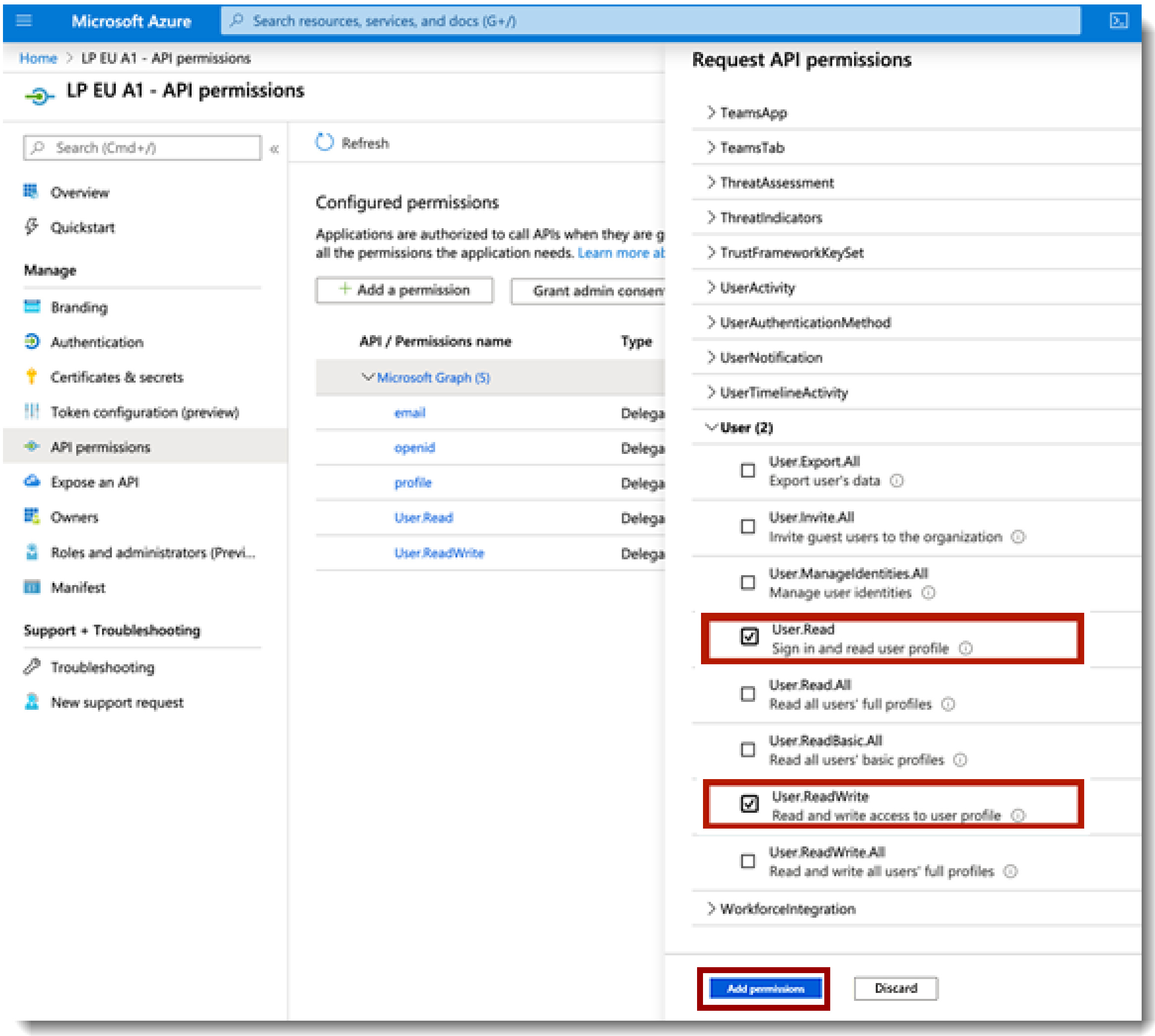
What type of permissions does your application require?

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

Application permissions
Your application runs as a signed-in user.

Permission	Ad
☑ email View users' email address	
□ offline_access Maintain access to data you have given it access to	
☑ openid Sign users in	
☑ profile View users' basic profile	
> AccessReview	
> AdministrativeUnit	
> AgreementAcceptance	
> Agreement	
> Analytics	
> AppCatalog	
> Application	
> AppRoleAssignment	
> ApprovalRequest	
> AuditLog	

5. Under the User menu, check the boxes to enable the following user settings:
   - **User.Read**
   - **User.ReadWrite**
6. When finished, click **Add permissions**.



Configured permissions

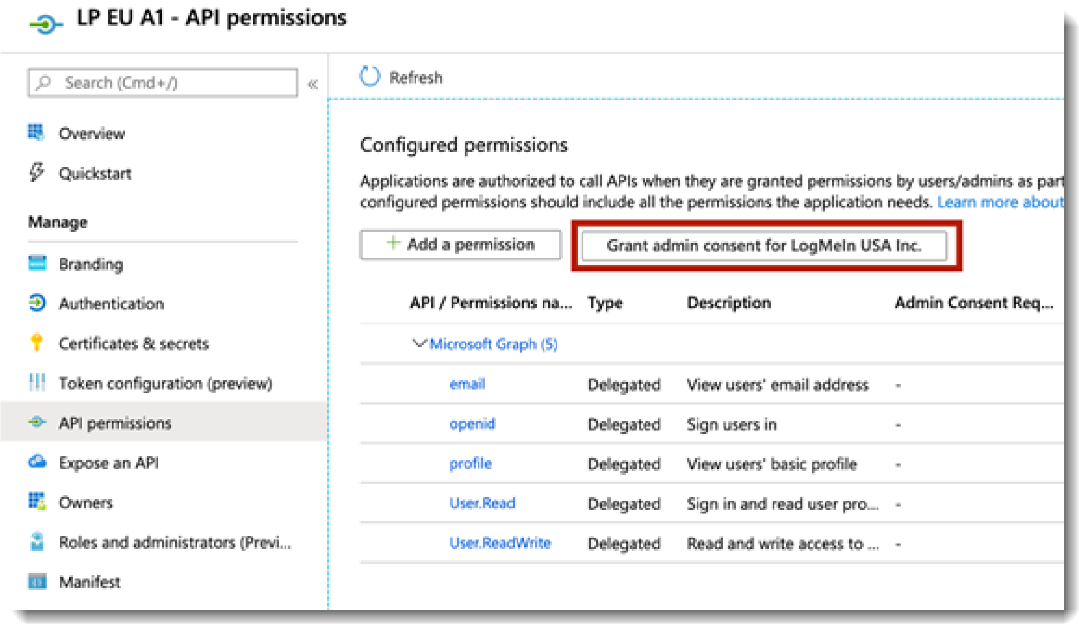
Applications are authorized to call APIs when they are g all the permissions the application needs. Learn more ab

API / Permissions name	Type
Microsoft Graph (5)	
email	Delega
openid	Delega
profile	Delega
User.Read	Delega
User.ReadWrite	Delega

Request API permissions

- > TeamsApp
- > TeamsTab
- > ThreatAssessment
- > ThreatIndicators
- > TrustFrameworkKeySet
- > UserActivity
- > UserAuthenticationMethod
- > UserNotification
- > UserTimelineActivity
- User (2)
  - □ User.Export.All — Export user's data
  - □ User.Invite.All — Invite guest users to the organization
  - □ User.ManageIdentities.All — Manage user identities
  - ☑ User.Read — Sign in and read user profile
  - □ User.Read.All — Read all users' full profiles
  - □ User.ReadBasic.All — Read all users' basic profiles
  - ☑ User.ReadWrite — Read and write access to user profile
  - □ User.ReadWrite.All — Read and write all users' full profiles
- > WorkforceIntegration

7. Under Configured Permissions, click **Grant <your LastPass application name>** to finish configuring API permissions for your LastPass app.



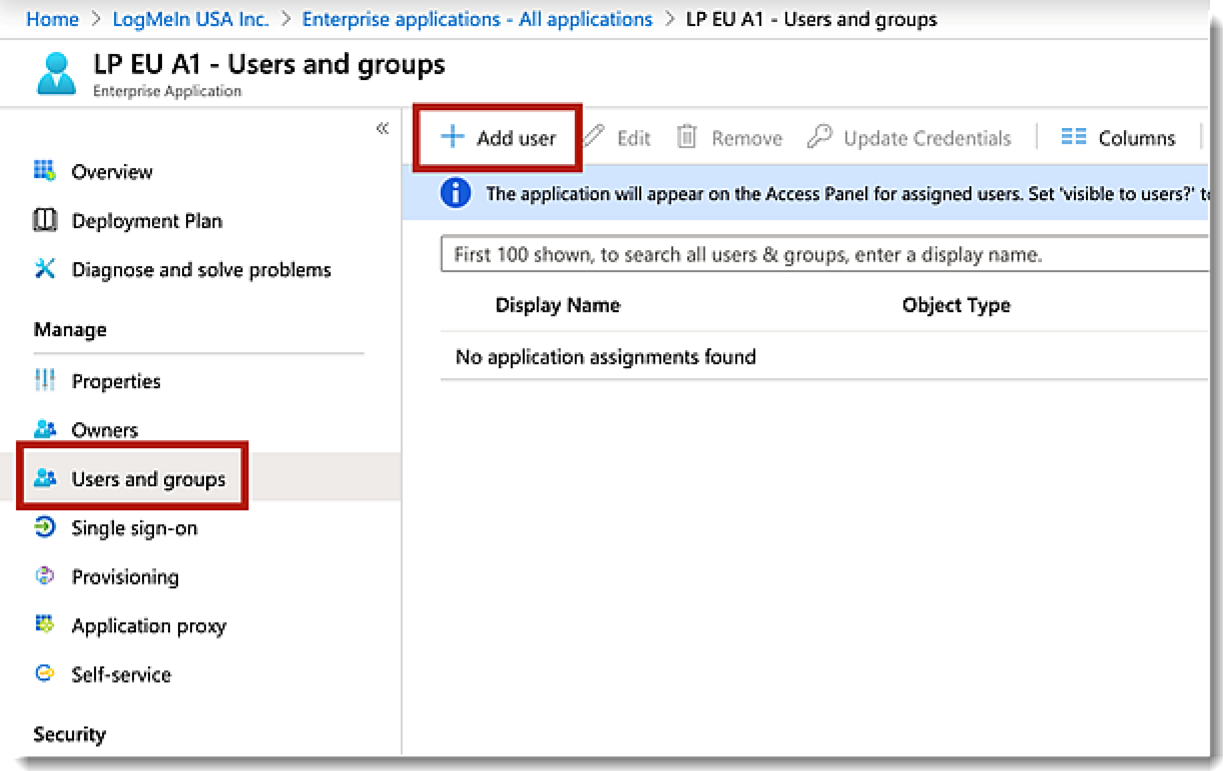
Configured permissions

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

API / Permissions na...	Type	Description	Admin Consent Req...
Microsoft Graph (5)			
email	Delegated	View users' email address	-
openid	Delegated	Sign users in	-
profile	Delegated	View users' basic profile	-
User.Read	Delegated	Sign in and read user pro...	-
User.ReadWrite	Delegated	Read and write access to ...	-

Step #7: Add users to the LastPass app in Azure AD

1. In the Azure AD portal, with your LastPass application selected, go to **Overview** > **Enterprise applications** in the left navigation.
2. Select your newly created LastPass application.
3. Select **Users and groups** in the left navigation.
4. Click **Add user**.
5. Locate each of the users and/or groups in the list, then click **Select** to grant access to the LastPass app.



Home > LogMeIn USA Inc. > Enterprise applications - All applications > LP EU A1 - Users and groups

LP EU A1 - Users and groups
Enterprise Application

The application will appear on the Access Panel for assigned users. Set 'visible to users?' t

First 100 shown, to search all users & groups, enter a display name.

Display Name	Object Type
No application assignments found	

Step #8: Set up Multifactor Authentication on Azure AD (optional)

If desired, you can set up Multifactor Authentication at the Azure AD (Identity Provider) level.

You’re all set!

You have successfully set up your LastPass Enterprise or LastPass Identity account to use federated login with your Azure Active Directory. All of your newly populated federated users will receive a Welcome email informing them that they can now log in to use LastPass. Please note that your LastPass users must log in using the LastPass web browser extension in order to use federated login for their Azure AD account with LastPass.

- To learn more about deploying the LastPass web browser extension to your organization, please see Install LastPass Software Using the Admin Console.
- To see your end users' experience, please see Federated Login Experience for LastPass Users.
- If your end users have linked personal accounts associated with their federated login account, please see How do I verify my linked personal account?
- To convert a non-federated user to a federated user, please see How do I convert an existing LastPass user to a federated (Azure AD) user?

Troubleshooting & Tips

- It is **required** that you enable the “Permit super admins to reset Master Passwords” policy for at least 1 LastPass admin (who is also a non-federated admin) in the LastPass Admin Console. This ensures that all LastPass user accounts can still be recovered (via Master Password reset) if a critical setting is misconfigured or changed for federated login after setup is complete.

Contact Us

If you have not started a LastPass Enterprise or LastPass Identity trial, please contact our Sales team at lastpass.com/contact-sales for more information.

For additional help, please see Set Up Federated Login for LastPass Using Azure Active Directory, and if desired, select a contact option at the bottom of the article.