

How-to: Azure Conditional Access and Jamf Connect

Change log:

14JUL2022 - Added note about ROPG scopes, added note about Discovery URL no longer needs to be defined in Jamf Connect 2.13 or greater.

Minimum version requirements: Jamf Connect 2.11 or greater

Summary:

What to do if administrators find "failed" login attempts in Azure sign-in logs when using Jamf Connect

In this blog post, you'll learn:

- How the ROPG workflow and Jamf Connect communicate
- How to make an app registration in Microsoft Azure Active Directory that allows for Conditional Access policies
- How to make Conditional Access policies NOT apply to the ROPG workflow

Problem

Administrators may observe failed login attempts in the log for the enterprise application created in Microsoft Azure Active Directory when using Jamf Connect and a Conditional Access policy that requires Multi-Factor Authentication (MFA) for the target of "All cloud apps." While this is expected behavior of the Resource Owner Password Grant (ROPG) workflow, it may trigger a user appearing in the Risky Sign-Ins in Azure Active Directory security reports.

What is happening

The target of "All cloud apps" applies policies far beyond the logins to specific cloud services and applies policies to non-interactive workflows like those with ROPG.

Specifically, the "All cloud apps" appears to apply to any application requesting a login with the scope of any of the following: `openid profile email`

The Open ID Connect 2.0 specification uses these default scopes to obtain an access or identity token for a client application. Consequently, in its default configuration, Jamf Connect login uses the `openid profile email` scope, and the only way to apply a CA policy in this default behavior is to apply the policy to "All cloud apps" with NO exceptions applied or the CA policy will break.

Administrators have multiple options for enforcing MFA on the Jamf Connect login screen:

- **Simplest, but most impact on user logins:**
Set hard requirements for MFA via the older method of Azure Multi-Factor Authentication which applies an MFA requirement to ALL logins to ANY service for a specific user. Ignore failed logins in the sign-in logs for ROPG checks of the password. (Additional information on how to determine if a failed login is due to Jamf Connect menu bar agent doing an ROPG request is below.)
- **Simple, but may affect other services:**
Apply a Conditional Access policy applied to "All cloud apps" requiring multi-factor authentication for login. Do NOT use an exception to the policy as that appears to break functionality of the CA rule as of testing done 10DEC2021. Ignore failed logins in the sign-in logs for ROPG checks of the password.
- **Complex, but exacting:**
Follow the instructions below to create a "private/public" app combination. Verify that no policies are created that apply to "All cloud apps" as to not affect the ROPG workflow. CA policy will be applied as expected to the Jamf Connect login application and ROPG check will appear as a successful login in sign-in logs.

Azure Multi-factor Authentication vs. Conditional Access

Administrators can enable multi-factor authentication requirements for a user account in two ways:

- Multi-factor Authentication which is reachable via the "All services" list in the Azure portal
 - Conditional Access which is reachable via Azure Active Directory under Security
- Multi-factor Authentication is a system-wide, all-login-attempts, master switch

system for enforcing MFA at authentication. While IP address ranges can be exempted, the rules apply to all authentications.

Conditional Access allows for fine grain details to apply for when MFA is required including exempting MFA for web applications.

Resource Owner Password Grant workflow

Jamf Connect uses a Resource Owner Password Grant (ROPG) workflow to synchronize the user's password in the identity provider with the password on the user's client machine. The user name and the password are sent to the identity provider in a "non-interactive" login to receive a response. This means that the user is not prompted for any sort of user name or password when logging in; Jamf Connect is using the information securely stored in the user's keychain for this event.

For Azure, the responses are one of the following:

- **Success, no MFA requirements:**

An access, refresh, and ID token encoded in HS256

- **Success, MFA required through a policy:**

An error response like:

```
AADSTS50076: Due to a configuration change made by your administrator, or because you moved to a new location, you must use multi-factor authentication to access [application UUID]
```

- **Failure, bad password or user name:**

An error response like:

```
AADSTS50126: Error validating credentials due to invalid username or password.
```

As long as the user password is correct, the ROPG flow has succeeded - the password has been validated to be correct. Whereas Jamf Connect has no need for the access, refresh, and ID token to keep the local password in sync with the identity provider, an appropriate error response is interpreted as a successful password check.

Reference: <https://docs.jamf.com/jamf-connect/administrator->

[guide/Authentication_Protocols.html](#)

Diagnosing MFA vs. failed password in Azure logs

Navigate to Azure Active Directory → Enterprise Applications and select the name of your Jamf Connect application in Azure. Navigate to Activity → Sign-ins to open user usage logs.

Date	Request ID	User	Application	Status	IP address	Location	Conditional access	Authentication requi...
3/2/2021, 1:37:02 PM	d8fe2ab1-5896-401a-8c9...	Sean Rabbitt	Jamf Connect - jamfse.io ...	Failure	206.214.236.79	San Jose, California, US	Not Applied	Multi-factor authentication
3/2/2021, 1:36:55 PM	7bdf11f6-89ab-4ca8-8a5...	Sean Rabbitt	Jamf Connect - jamfse.io ...	Failure	206.214.236.79	San Jose, California, US	Not Applied	Single-factor authenticati...

Example logs from a sample Microsoft Azure instance

Shown above are two logins which appear to be failures. Under the “Authentication required” column, the first login says “Multi-factor authentication”. Clicking on the row will pull up additional details about the login attempt.

Details						
Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	Additional Details
Authentication Policies Applied						
Per-user MFA						
Date	Authentication method	Authentication method...	Succeeded		Result detail	Requirement
3/2/2021, 1:37:02 PM	Password	Password Hash Sync	false		User did not pass the MF...	User
3/2/2021, 1:37:02 PM			false		MFA required in Azure AD	User

Details on a non-interactive login from Jamf Connect via ROPG

Under Authentication Details, the “Result detail” will let an administrator determine if the login was successful or a failure. In this example, the login was a success - the Result detail shows that the “User did not pass the MFA challenge (non interactive).” This login can be interpreted in that the user was required to use MFA by either a Conditional Access policy or through Azure Multi-factor authentication. In the second example, a user with MFA required failed to enter their correct password:

Details

Basic info

Location

Device info

Authentication Details

Conditional Access

Report-only

Additional Details

Date

Authentication met...

Authentication met...

Succeeded

Result detail

3/2/2021, 1:36:55 PM

Password

Password Hash Sync

false

Invalid username or password or Invalid on-premise username or passwo

Details on a failed non-interactive login due to an incorrect password

The Authentication required column shows “Single-factor authentication” and the Authentication Details show “Invalid username or password or Invalid on-premise username or password.” While the user is required to use Multi-factor authentication, the user failed the first, single factor and thus was never prompted for MFA.

Note on ROPG at the login screen: As of Jamf Connect 2.13, the ROPG scopes sent to Azure will be the same scope sent to OIDC when defined with the `OIDCScope` key. Administrators may see a single failure with error code 50037 immediately after a successful Multi-factor authentication when logging into Azure. Error code 50037 is interpreted by the Jamf Connect login mechanism as a successful validation of the password. This behavior is expected.

Creating a Custom Scope for Jamf Connect and Conditional Access policies

Workflow overview:

- Create a “private endpoint” application registration with a custom API
 - With API permissions for “User.read”
 - With “Expose an API” scope created
 - Define roles like “Admin” and “Standard” for elevating macOS account permissions
- Create a “public endpoint” application registration for OIDC to call that custom API
 - Remove API permissions for “User.read”
 - Add API permission for “My APIs” for the name of the application created in first step and the scope created in first step
- Create an Azure Conditional Access policy to require multifactor authentication
 - Apply to application created in first step
- Remove any CA policy applied to “All cloud apps” that would require MFA
- Create a Jamf Connect Login configuration profile
 - Azure as Identity Provider
 - Define a custom scope
 - Define the Discovery URL for OIDC and ROPG
 - Test with Jamf Connect Configuration

H3 Step One: Create an application registration with a custom API

Navigate to portal.azure.com → Azure Active Directory → App Registrations. Create a new app registration. Name the application something like “Jamf Connect - Conditional Access Policy API”. Select the supported account types to “Accounts in this organizational directory only”. Leave Redirect URI section blank. Register the application.

Microsoft Azure

Search resources, services, and docs (G+)

[Home](#) > [jamfse.io](#) >

Register an application

*

Name

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

Jamf Connect - Conditional Access Policy API

Supported account types

Who can use this application or access this API?

☒

Accounts in this organizational directory only (jamfse.io 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)☐[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.

Web

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

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

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

Application Registration Screen (as of 06DEC2021)

Navigate to API permissions on the left hand navigation bar. Grant admin consent

for the organization.

Jamf Connect - Conditional Access Policy API | API permissions

Search (Cmd+/) Refresh Got feedback?

Overview
Quickstart
Integration assistant

Manage

Branding
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators | Preview
Manifest

Support + Troubleshooting

Troubleshooting
New support request

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect th

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 [redacted]

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

To view and manage permissions and user consent, try [Enterprise applications](#).

Using the left hand navigation bar, select "Expose an API". Set the Application ID URI. A default entry will be created based on the pattern of `api://[application ID]`. This may be modified if desired but default entry is acceptable.

Jamf Connect - Conditional Access Policy API | Expose an API

Search (Cmd+/) Got feedback?

Overview
Quickstart
Integration assistant

Manage

Branding
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators | Preview
Manifest

Support + Troubleshooting

Troubleshooting
New support request

Set the App ID URI

Application ID URI
`api://66b2a554-0863-44be-a8e6-303cd3645b3c`

Save Discard

Adding a scope here creates only delegated permissions. If you are looking to create application-only scopes, use 'App roles' and define app roles assignable to application type. [Go to App roles](#).

+ Add a scope

Scopes	Who can consent	Admin consent display ...	User consent display na...	State
No scopes have been defined				

Authorized client applications

Authorizing a client application indicates that this API trusts the application and users should not be asked to consent when the client calls this API.

+ Add a client application

Client Id	Scopes
No client applications have been authorized	

Select the option for "Add a scope"

Conditional Access Policy API

Conditional Access Policy API | Expose an API

Got feedback?

Application ID URI `api://66b2a554-0863-44be-a8e6-303cd3645b3c`

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protected by the API. An application that requires access to parts of this API can request that a user or admin consent to one or more of these.

Adding a scope here creates only delegated permissions. If you are looking to create application-only scopes, use 'App roles' and define app roles assignable to application type. [Go to App roles.](#)

+ Add a scope

Scopes	Who can consent	Admin consent display name	User consent display name	State
No scopes have been defined				

Authorized client applications

Authorizing a client application indicates that this API trusts the application and users should not be asked to consent when the client calls this API.

+ Add a client application

Client Id	Scopes
No client applications have been authorized	

Add a scope

Scope name *

 api://66b2a554-0863-44be-a8e6-303cd3645b3c/

Who can consent?

☐ Admins and users ☒ Admins only

Admin consent display name *

e.g. Read user files

Admin consent description *

e.g. Allows the app to read the signed-in user's files.

User consent display name *

e.g. Read your files

User consent description *

e.g. Allows the app to read your files.

State

☒ Enabled ☐ Disabled

Add scope

Cancel

Set the scope name to `jamfconnect`. Set "Who can consent" to "Admins". Enter information into the Admin consent display name and Admin consent description. Any text is acceptable - this will be accepted by the admin in the next step. Press "Add scope" to save.

Add a scope

Scope name *

`jamfconnect``api://66b2a554-0863-44be-a8e6-303cd3645b3c/jamfconnect`

Who can consent?

☒ Admins and users ☐ Admins only

Admin consent display name *

Read user information

Admin consent description *

Allows Jamf Connect to read user information like user name, email address, real name, role, and group membership if required.

User consent display name ⓘ

Read user information ✓

User consent description ⓘ

Users should never see this description unless an administrator has failed to grant consent for the organization.

State ⓘ

Enabled

Disabled

Add scope

Cancel

Copy the scope with the Copy button and save it for later. This will be used as the `OIDCScopes` later in Jamf Connect Configuration.

H3 Step Two: Create an application registration using this new API permission

Return to Azure Active Directory → App Registrations. Create a new app registration. Name the app "Jamf Connect - OIDC Endpoint". Set Supported account types to "Accounts in this organizational directory only". Set Redirect URI to "Public client/native (mobile & desktop)" with the value `https://127.0.0.1/jamfconnect`. Register the application.

Register an application ...

* Name

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

Jamf Connect - OIDC Endpoint ✓

Supported account types

Who can use this application or access this API?

- ☒ Accounts in this organizational directory only (jamfse.io 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

[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 ... ▼ <https://127.0.0.1/jamfconnect> ✓

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

By proceeding, you agree to the [Microsoft Platform Policies](#) ↗

Register

Navigate to API permissions. By default, the Microsoft Graph → User.Read permission is added. Use the ellipses to the right of Status to remove this permission from the application. Next, select "+ 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 jamfse.io

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Microsoft Graph (1) ...				
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for jamfse.io

Remove permission
Revoke admin consent

To view and manage permissions and user consent, try [Enterprise applications](#).

Select the “My APIs” tab. Select the name of the application you created in step 1.

Request API permissions

Select an API

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

Applications that expose permissions are shown below

Name	Application (client) ID
Jamf Setup - Retail	3c272147- 
Jamf Connect - Conditional Access Policy API	66b2a554- 
Jamf Connect - INFOSEC ONLY ACCESS	b92961e0- 

Select the option for “Delegated permissions” and check the box for “jamfconnect” - the only permission listed in the application. Use the “Add permissions” button to close the window.

Request API permissions

[← All APIs](#)**Jamf Connect - Conditional Access Policy API**

api://66b2a554-0863-44be-a8e6-303cd3645b3c

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.

Select permissions

[expand all](#)

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)



Permission

Admin consent required

▼ Permissions (1)



jamfconnect ⓘ

Read user information

No

[Add permissions](#)[Discard](#)

Use the "Grant admin consent for [domain]" to grant permission to access the API on behalf of users.

Optional: Use the "App roles" option to add a role for "Administrator" and "Standard". This will allow you to define users or groups of users directly in Azure who should have administrator rights on macOS client machines. "App roles" is located on the left hand navigation tool bar in the App registration - refer to

https://docs.jamf.com/jamf-connect/2.8.0/documentation/Login_Window_Preferences.html#ID-00007186 for more details on the `OIDCAdminAttribute` and `OIDCAdmin` settings for Jamf Connect.

Navigate to Overview. Record the Application (client) ID and the Directory (tenant) ID for later use with Jamf Connect Configuration.

^ Essentials

Display name : [Jamf Connect - OIDC Conditional Access](#)

Application (client) ID : baf44d07-

Object ID : c520d014-

Directory (tenant) ID : f83fb0da-

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

Navigate to Azure Active Directory → Enterprise Applications. Find the Jamf Connect - OIDC Endpoint application you created and assign users and roles to the application.

H3: Step Three: Create an Azure Conditional Access policy

Navigate to portal.azure.com → Azure Conditional Access. Create a new policy.

[Home](#) > [Conditional Access](#)

Conditional Access | Policies

Azure Active Directory

[+ New policy](#) ▾[What If](#)[Refresh](#)[Overview \(Preview\)](#)[Policies](#)[Insights and reporting](#)[Diagnose and solve problems](#)[Create new policy](#)[Create new policy from templates \(Preview\)](#)[Filter](#)

To improve the resilience of Azure AD, we are an

Name the policy as desired. The sample will name the policy “Jamf Connect - Require Multifactor Authentication”

[Home](#) > [Conditional Access](#) >

New ...

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies.
[Learn more](#)

Name *

Jamf Connect - Require Multifactor Auth... ✓

Assignments

Users or workload identities ⓘ

0 users or workload identities selected

Cloud apps or actions ⓘ

No cloud apps, actions, or authentication contexts selected

Conditions ⓘ

0 conditions selected

Access controls

Grant ⓘ

0 controls selected

Session ⓘ


0 controls selected

Enable policy

Report-only

On

Off

 Do not block yourself out! This policy impacts the Azure portal and other clients that do not support CAE today.

Create

Select “Users or workload identities”. Select a test user to test your conditional access policy before applying to all users.

[Home](#) > [Conditional Access](#) >

New ...

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies.

[Learn more](#)

Name *

Jamf Connect - Require Multifactor Auth... ✓

Assignments

Users or workload identities ⓘ

[Specific users included](#)

✗ "Select users and groups" must be configured

Cloud apps or actions ⓘ

[No cloud apps, actions, or authentication contexts selected](#)

Conditions ⓘ

[0 conditions selected](#)

Access controls

Grant ⓘ

Control access based on who the policy will apply to, such as users and groups, workload identities, directory roles, or external guests.

[Learn more](#)

What does this policy apply to?

Users and groups ▼

Include Exclude

- ☐ None
- ☐ All users
- ☒ Select users and groups

☐ All guest and external users ⓘ

☐ Directory roles ⓘ

☒ Users and groups

Select

[0 users and groups selected](#)

✗ Select at least one user or group

Select “Cloud apps or actions”. Select the Jamf Connect - Conditional Access Policy API you created in step one.

[Home](#) > [Conditional Access](#) >

New

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies. [Learn more](#)

Name *

Jamf Connect - Require Multifactor Auth... ✓

Assignments

Users or workload identities ⓘ

[Specific users included](#)

Cloud apps or actions ⓘ

No cloud apps, actions, or authentication contexts selected

✗ "Select apps" must be configured

Conditions ⓘ

Control access based on all or specific cloud apps or actions. [Learn more](#)

Select what this policy applies to

Cloud apps ▼

Include Exclude

- ☐ None
- ☐ All cloud apps
- ☒ Select apps

Select

None

✗ Select at least one app.

Select

Cloud apps



JC

Jamf Connect - Conditional Access Policy API



JC



JC

Selected items

Select "Grant". Check the option for "Require multi-factor authentication". Set Enable policy to "On" and "Create" to save the policy.

[Home](#) > [Conditional Access](#) >

New

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies. [Learn more](#)

Name *

Jamf Connect - Require Multifactor Auth... ✓

Assignments

Users or workload identities ⓘ

[Specific users included](#)

Cloud apps or actions ⓘ

[1 app included](#)

Conditions ⓘ

[0 conditions selected](#)

Access controls

Grant ⓘ

[0 controls selected](#)

Session ⓘ

[0 controls selected](#)

Enable policy

[Report-only](#) On Off

⚠ Do not block yourself out! This policy impacts the Azu today.

[Create](#)

Grant



Control access enforcement to block or grant access. [Learn more](#)

☐ Block access☒ Grant access☒ Require multi-factor authentication ⓘ☐ Require device to be marked as compliant ⓘ☐ Require Hybrid Azure AD joined device ⓘ☐ Require approved client app ⓘ
[See list of approved client apps](#)☐ Require app protection policy ⓘ
[See list of policy protected client apps](#)☐ Require password change ⓘ☐ RequireDuoMfa

For multiple controls

☒ Require all the selected controls☐ Require one of the selected controls[Select](#)

(H3) Step Four: Remove any Conditional Access policies applied to All cloud apps

Navigate to portal.azure.com → Azure Conditional Access. Examine any application applied to the scope of “All cloud apps”. Either set “Enable policy” to “Off” for any application that has a Grant of “Require multi-factor authentication” or modify the “Cloud apps or actions” to specifically list resources that should have MFA applied.

Applying a policy to require MFA for “All cloud apps” will cause the ROPG application in the next step to inaccurately show failed logins in the Azure sign-in logs.

Undocumented behavior: If you wish to keep “All cloud apps” as a definition, but you still want the policy to not be applied to the `openid` scope, create a bogus Enterprise app registration for an unused SAML application, and then use that bogus app registration as an “Exclude” to the “All cloud apps” policy:

New

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies. [Learn more](#)

Name *

Example: 'Device compliance app policy'

Assignments

Users or workload identities ⓘ

0 users or workload identities selected

Cloud apps or actions ⓘ

All cloud apps included and 1 app excluded

Conditions ⓘ

0 conditions selected

Control access based on all or specific cloud apps or actions. [Learn more](#)

Select what this policy applies to

Cloud apps

Include Exclude

Select the cloud apps to exempt from the policy

Select excluded cloud apps

[Microsoft Planner](#)



Microsoft Planner

09abbdfd-ed23-44ee-a2d9-a627aa1c90f3



This is undocumented behavior, but the application of an exclusion to the policy will break how the policy is applied to `openid` scope. As this is undocumented, the behavior may be unexpected and may change by Microsoft without notice.

(H3) Step Five: Create a Jamf Connect Configuration Profile

Use the Jamf Connect Configuration app included in the Jamf Connect software distribution disk image which you can download from account.jamf.com with your Jamf Nation credentials.

On the Identity Provider tab, set:

- Identity Provider: Azure
- OIDC Client ID: The application ID of the PUBLIC application you created in Step Two
- ROPG Client ID: The same application ID
- Scopes: Combine the scope you saved in Step One with `+openid+profile+email` to look similar to: `api://[RANDOM UUID STRING]/jamfconnect+openid+profile+email`
- Tenant: Enter the UUID of the tenant of your Azure instance. This can be found under the "Overview" tab of either of the App registrations made in Step One or Step Two.
- OIDC Redirect URI: (optional) Set to `https://127.0.0.1/jamfconnect`
- Discovery URL: **This step is not required for Jamf Connect version 2.13 or greater.** This can be found under the "Overview" tab of either of the App registrations from Step One or Step Two under the "Endpoints" option. You can also manually create it by using the UUID of the tenant of your Azure instance in a format like the following:
`https://login.microsoftonline.com/[TENANT UUID]/v2.0/.well-known/openid-configuration`

The screenshot shows the Jamf Connect configuration interface with two tabs: 'Identity Provider' and 'Connect'. The 'Identity Provider' tab is active and contains the following fields:

- Required:**
 - Identity Provider: Azure (with a dropdown arrow)
 - OIDC Client ID: baf44d07-PUBLIC_APP_REGISTRATION_UUID
 - ROPG Client ID: baf44d07-PUBLIC_APP_REGISTRATION_UUID
- Advanced OIDC:**
 - Scopes: api://[RANDOM UUID STRING]/jamfconnect+openid+profile+email
 - Token Caching: ☐ Ignore cookies
 - Client Secret: JCCyfvL7YWtP6gudLjBZRZV_N0dW4f3xETilxqtokEAZ6FAsBtgylq0MpU1uQ7Jid
 - ROPG Tenant: f83fb0da-TENANT_ID
 - OIDC Redirect URI: https://127.0.0.1/jamfconnect
 - Discovery URL: https://login.microsoftonline.com/f83fb0da-TENANT_ID/v2.0/.well-known/openid-configuration (highlighted with a blue border)

At the bottom, there is a 'Choose License...' button and a message: 'Jamf Connect operates in trial mode without a license'.

OPTIONAL: If you want to define a role for users to be made administrators on a macOS client device, on the Login tab, set:

- User Creation → Admin Roles: The value of the administrator App role you created in Step Two
- User Creation → Admin Attribute: `roles`

On the Connect tab, set:

- Authentication
 - ROPG Client ID: This should auto populate from your entry on the Identity Provider screen
 - ROPG Tenant: The UUID of the Azure tenant
 - ROPG Scopes: **This step is only available in Jamf Connect version 2.11 or greater.** Set value to `openid+email+profile`
 - Discovery URL: **This step is not required for Jamf Connect version 2.13 or greater.** This format is DIFFERENT from the Discovery URL from the Identity Provider tab. This can be found under the "Overview" tab of either of the App registrations from Step One or Step Two under the "Endpoints" option for the V1 endpoint. You can also manually create it by using the UUID of the tenant of your Azure instance in a format like the following:

`https://login.microsoftonline.com/[TENANT UUID]/.well-known/openid-`

configuration

(note how the /v2.0 is missing from the URL.

Test your configuration with the test user via OIDC. Make sure MFA was required.

Test your configuration with the test user via ROPG. Validate in the Azure portal under Azure Active Directory → Sign-in logs that the authentication was successful. Look for the Authentication Requirement to be “Single-factor authentication”. The Basic tab will show something like:

Basic info	Location	Device info	Authentication Details	Conditio
Date		1/13/2022, 1:00:07 PM		
Request ID		1d801759-1eea-416e-a079-1a862b495d00		
Correlation ID		1a51280a-b717-4dd9-ba49-39823d0ce55f		
Authentication requirement		Single-factor authentication		
Status		Success		

The Conditional Access tab should show that no policy was applied to the login.