

Radiator

CONFIGURATION GUIDE

Entra ID configuration guide

Date: 2026-11-16

Important notice

Before proceeding with this configuration guide, you must first complete the Microsoft Entra ID application creation process. This includes creating an app registration, configuring API permissions, generating client secrets, and handling the MFA exclusion.

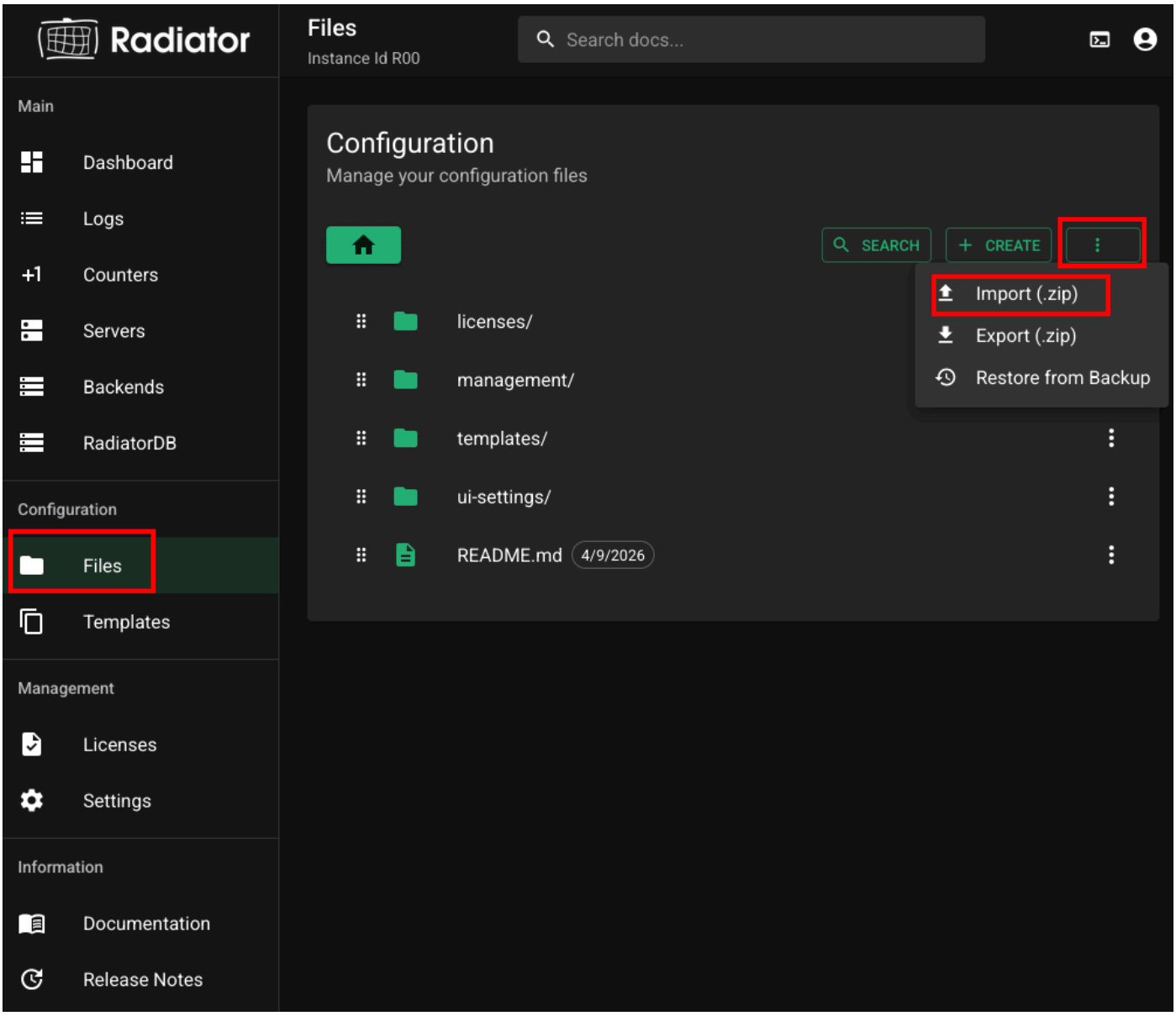
Please refer to the Microsoft Entra ID App creation for Radiator Policy Server authentication before continuing.

Example configuration setup

Radiator offers various Entra ID example configurations that demonstrate authentication with Microsoft Entra ID and optionally MFA support in the public radiator-radconfs repository.

A good starting point is the `entra-auth-mfa` configuration, which authenticates against Microsoft Entra ID with MFA support, or `entra-ad`, which does not include MFA support.

To get a working configuration, download the configuration zip from the radiator-radconfs repository and import it from the GUI by selecting **Files** > **Import (.zip)**



Please note that the imported configuration contains placeholder values that must be replaced with your actual Entra ID application information (covered in the following sections).

Service principal creation for MFA usage

Microsoft Entra ID requires a service principal with appropriate permissions to handle MFA notifications. With the PowerShell script below, you can create a new client secret for Microsoft's Entra ID MFA Notification Client. Make a note of this secret; it must be added to the Radiator configuration so that Radiator is allowed to use the MFA notification service (covered in the next section).

Run this PowerShell script to create the required secret (running this script requires global admin rights):

```
 Connect-MgGraph -Scopes 'Application.ReadWrite.All'
# Get ID of 'Entra ID MFA Notification Client' Service Principal
$servicePrincipalId = (Get-MgServicePrincipal -Filter "appid eq '981f26a1-7f43-403b-a875-
f8b09b8cd720'").Id

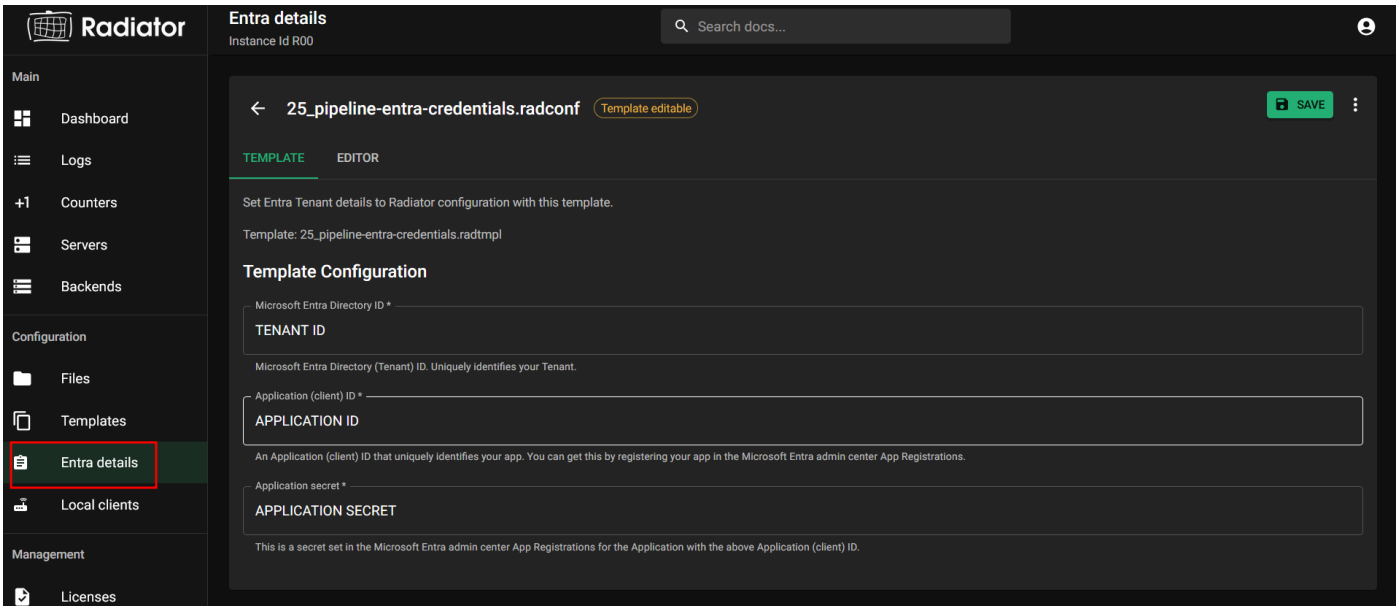
$params = @{
        passwordCredential = @{
                displayName = "My Application MFA"
        }
}

# Create Client Secret onto client
$secret = Add-MgServicePrincipalPassword -ServicePrincipalId $servicePrincipalId -BodyParameter
$params
Write-Output $secret
```

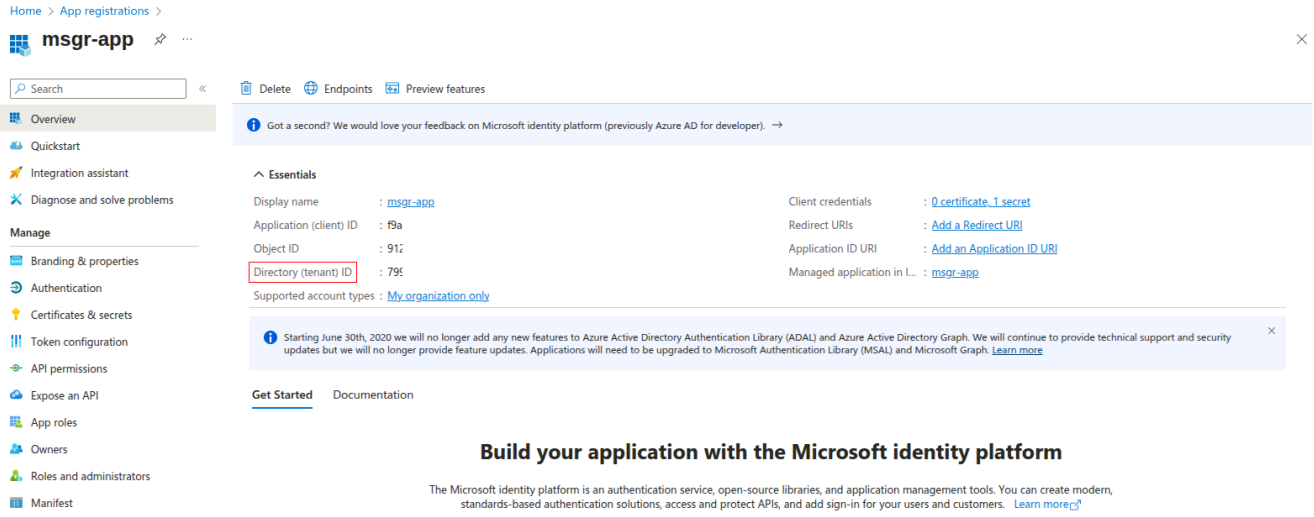
Record the secret's value for use in Radiator Policy Server configuration.

Updating the example configuration

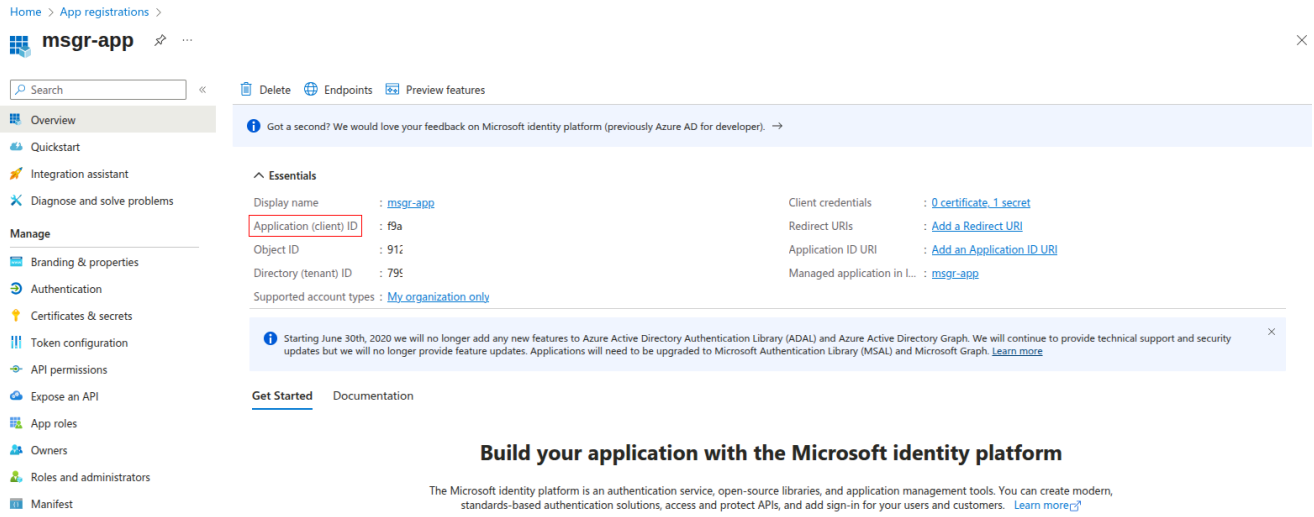
After completing the previous steps as well as the Microsoft Entra ID App creation for Radiator Policy Server authentication guide, you will need to modify the example configuration from the first step to match your specific Entra ID application details. This can be done from the GUI by selecting **Entra details** on the left side panel and filling in the fields (see below). The MFA specific field is only shown if the starting point was `entra-auth-mfa` configuration.



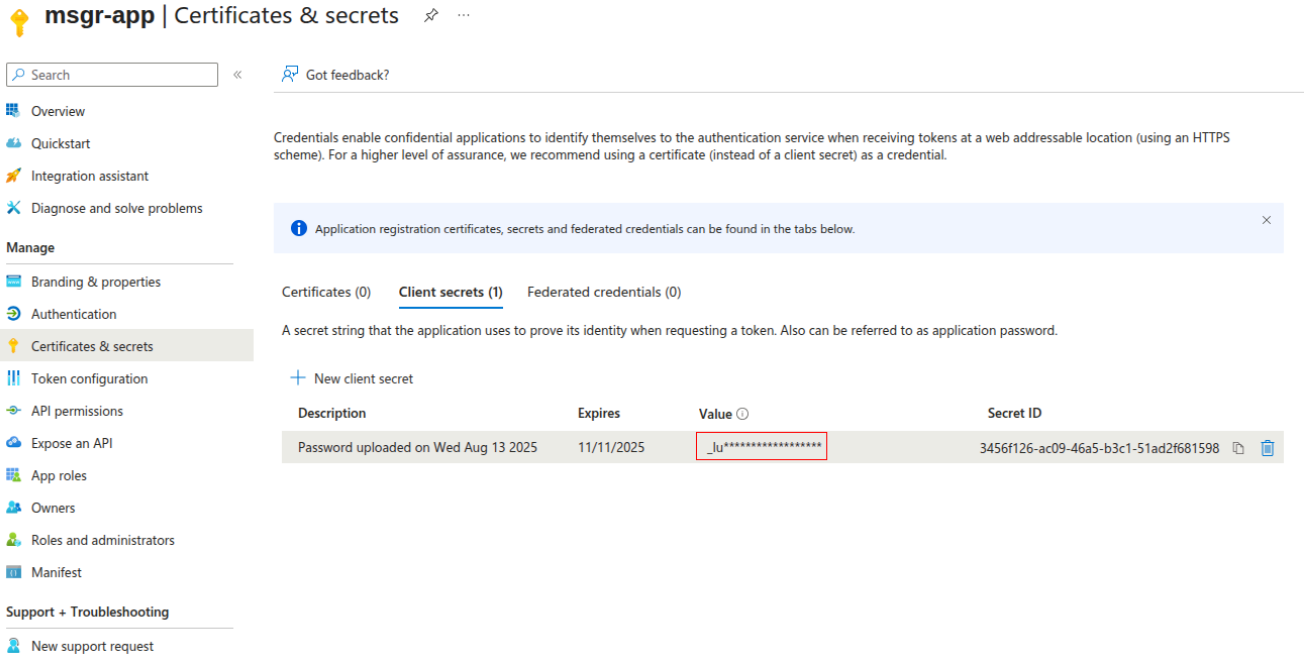
1. Replace `TENANT ID` with the Directory (tenant) ID. The Directory (tenant) ID is shown on the application Overview page:



2. Replace `APPLICATION ID` with the Application (client) ID. The Application (client) ID is shown on the application Overview page:



3. Replace `APPLICATION SECRET` with the secret. The secret is shown on the application Certificates & secrets page right after creation:



4. If shown, replace `MFA PROVISIONING SECRET` with the secret you created in the service principal creation section.
5. Save the configuration file by pressing **Save**

When doing changes on the GUI, new configuration is automatically taken into use.

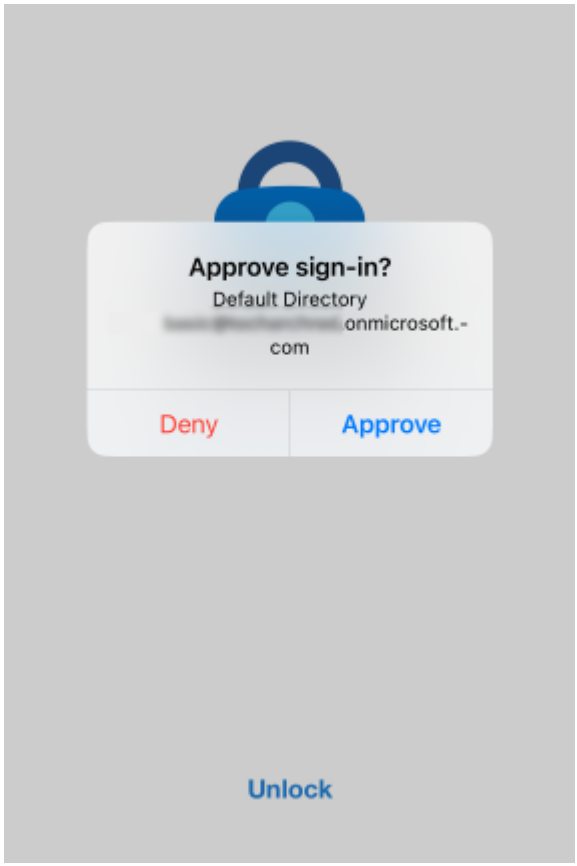
Testing

You can test the Entra ID authentication using the Radiator test client on the CLI with this command:

```
/opt/radiator/server/bin/radiator-client --server 127.0.0.1 --secret mysecret --user <user@entra-domain.com> --password <password> --timeout 120
```

Note: Replace `mysecret` with your configured RADIUS client secret, and use actual user credentials from your Entra ID tenant.

After running the test command, you should receive an MFA notification on your authenticator app:



MFA response options:

- **Accept the MFA** - Access will be granted and authentication succeeds
- **Reject the MFA** - Access will be denied and authentication fails
- **No response** - The request will time out after the specified duration

The `--timeout 120` parameter allows sufficient time for users to respond to MFA prompts on their mobile devices.