PTT Pro and Profile Manager

Workcloud Communication

ZEBRA

AD/ADFS Integration Guide

MN-004591-03EN Rev A

Copyright

2025/05/06

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2025 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: zebra.com/informationpolicy.
COPYRIGHTS: zebra.com/copyright.
PATENTS: ip.zebra.com.
WARRANTY: zebra.com/warranty.
END USER LICENSE AGREEMENT: zebra.com/eula.

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Objectives

A large percentage of the Workcloud Communication deployments already have Active Directory (AD) and Active Directory Federated Services (ADFS) deployed in their enterprise.

The purpose of this document is to inform the administrator of the items necessary for PTT Pro and Profile Manager to provide authentication, provisioning, and transformation using the existing AD database.

Environment Description

The Workcloud Communication solution supports the use of OAuth2 authentication using ADFS to provide the ability for a single device to be used by multiple users. This is often referred to as the Shared Device Model as opposed to the Dedicated Device Model.

The support of a Shared Device model is based on the configuration of a Relying Party Trust in the customer's ADFS server. A fully integrated Workcloud Communication solution includes integrating with an Identity Provider (IdP) solution, like AD/ADFS. The functionality of an IdP connection serves three purposes:

1. User Authentication
   - Granting user access to the system by validating credentials
   - Providing a shared device usage model
2. User Provisioning
   - As associates join and leave the enterprise, they are added to and deleted from the IdP by the customer administrators. The connection to the IdP with Profile Manager and PTT Pro provides the ability to automatically synchronize the user databases with changes made in the IdP.
3. Attribute Transformations
   - Various elements in the IdP database can be evaluated to determine the profile configuration sent to the users.

When the Workcloud Communication solution is fully integrated with AD/ADFS, all three functions are available. Both Profile Manager and PTT Pro rely on the IdP to simplify and automate what would otherwise be a manual process.

The focus of this document is the ADFS configuration and how this determines the configuration of PTT Pro and Profile Manger.

PTT Pro Setup

When PTT Pro is deployed, Zebra requires the following information to configure the and for the customer.

- PTT Pro configuration elements:
  - OAuth URL
  - Access URL
  - OAuth Token

The configuration elements are available through the Well-Known URL.

NOTE:

- To support the shared-device model, the serial numbers of each device must be entered into the .

- The can support a mix of OAuth and Activation Code device users.
- Each device user must have an OAuth user name to authenticate to the Azure AD.

Profile Manager Setup

When the Profile Manager is deployed, Zebra requires the following information to configure the server for the customer.

- Host URL
- Authentication Path
- Token Path
- Client ID
- Client Secret
- Token UserName
- Client Authentication - Header or Body

ADFS Configuration

Workcloud Communication supports the Forms based sign-on process. Some deployments use Windows Integrated Authentication (WIA) which is not compatible.

This guide assumes the following:

1. The customer has an existing 2016 Windows Server with ADFS installed. Additional documentation describes differences when a Windows 2019 and Windows 2022 server.
2. Current Certificates are installed on the ADFS server.
3. The ADFS services are installed on a server that is joined to the Active Directory Domain Controller.
4. Administrative privileges are assigned to the user configuring the system.

If the deployment uses WIA, please review the following information:

- How to configure intranet forms-based authentication for devices that do not support Windows Integrated Authentication (WIA) | Microsoft Docs

ADFS Server Versions

Windows 2016

Windows 2016 Datacenter server supporting SPA (Single Page Authentication)

version 1607 10.0.14393 Build 14393

Server 2019 and 2022

Concepts presented for Server 2016 can be applied to Server 2019 and 2022.

Prerequisite Information

The PTT Pro and Profile Manager servers both require specific URL and Token information to provide a connection to ADFS.

This section describes how to use the Well-Known URL to discover this information

About the Well-Known URL for ADFS

The Well-Known URL concept was created to provide a publicly available central location for server and metadata resources. It provides publicly available information about a site.

A Well-Known URL defines a sign-in flow that enables a client application to authenticate a user and to obtain information (or claims) about that user, such as the user name, email, and so on. It is useful to identify URLs, encryption schemes, and other information useful in establishing server-to-server communications.

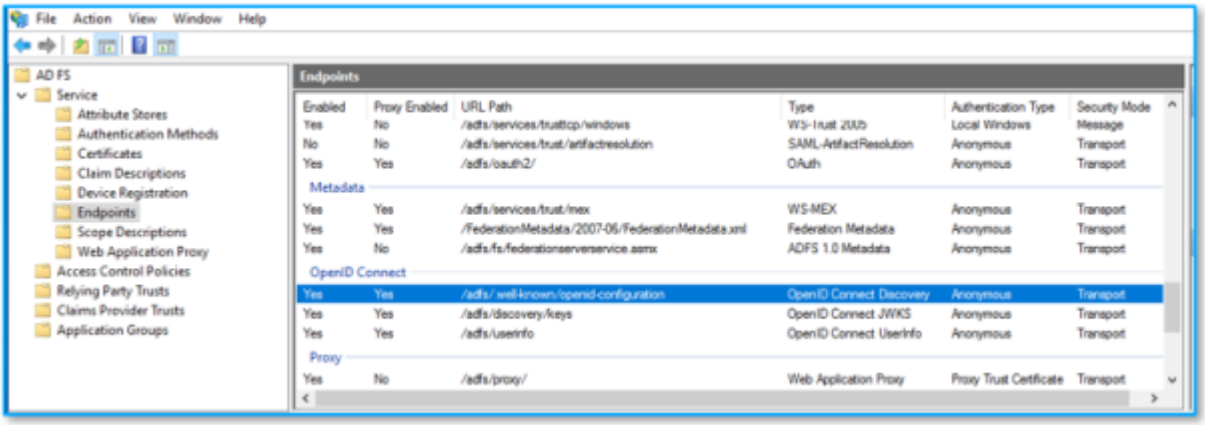
Customers may know their Well-Known URL. The construction of the Well-Known URL is typically `https://<server_DNS>/adfs/.well-known/openid-configuration`, although there can be differences.

How to Find the Well-Known URL

You can determine the Well-Known URL using the ADFS Management Console.

Navigate to **Endpoints**.

The Well-Known URL is constructed by taking the customer host name and appending the value of the OpenID Connect Discovery.



How to Use the Well-Known URL

The Well-Known URL helps configure the Profile Manager and PTT Pro servers to connect to the ADFS environment.

The URL returns a JSON listing of the OpenID/OAuth endpoints, supported scopes and claims, public keys used to sign the tokens, and other details. The clients can use this information to construct a request to the OpenID server.

The Authorization endpoint is shown below. In this example, the Identifier for the Relying Party Trust is Z-Test-Trust. The authorization_endpoint: `"https://pttproadfs.pttpro.zebra.com/adfs/oauth2/authorize/"` becomes `https://pttproadfs.pttpro.zebra.com/adfs/oauth2/authorize?resource=Z-Test-Trust`. Pasting this URL in a browser reveals helpful discovery information.

```
{
    issuer: https://PTTPRO-ADFS.pttpro.zebra.com/adfs,
    authorization_endpoint: https://pttproadfs.pttpro.zebra.com/adfs/oauth2/authorize/,
    token_endpoint: https://pttproadfs.pttpro.zebra.com/adfs/oauth2/token/,
    jwks_uri: https://pttproadfs.pttpro.zebra.com/adfs/discovery/keys,
  ▾ token_endpoint_auth_methods_supported: [
        "client_secret_post",
        "client_secret_basic",
        "private_key_jwt",
        "windows_client_authentication"
    ],
  ▾ response_types_supported: [
        "code",
        "id_token",
        "code id_token",
        "id_token token",
        "code token",
        "code id_token token"
    ],
  ▾ response_modes_supported: [
        "query",
        "fragment",
        "form_post"
    ],
  ▾ grant_types_supported: [
        "authorization_code",
        "refresh_token",
        "client_credentials",
        "urn:ietf:params:oauth:grant-type:jwt-bearer",
        "implicit",
        "password",
        "srv_challenge",
        "urn:ietf:params:oauth:grant-type:device_code",
        "device_code"
    ],
  ▾ subject_types_supported: [
        "pairwise"
    ],
  ▾ scopes_supported: [
        "winhello_cert",
        "vpn_cert",
        "email",
        "allatclaims",
        "profile",
        "logon_cert",
        "user_impersonation",
        "aza",
        "openid"
    ],
  ▾ id_token_signing_alg_values_supported: [
        "RS256"
    ],
  ▾ token_endpoint_auth_signing_alg_values_supported: [
        "RS256"
    ],
```


This information is provided here as an introduction to the Well-Known URL which is helpful in establishing the server connections to ADFS.

More information about using the Well-Known URL is provided in the validation section.

ADFS Setup Flows

The following process describes steps for configuring the ADFS services running on a Windows 2016 server.

ADFS Server Definition

Windows 2016 Datacenter server supporting SPA (Single Page Authentication)

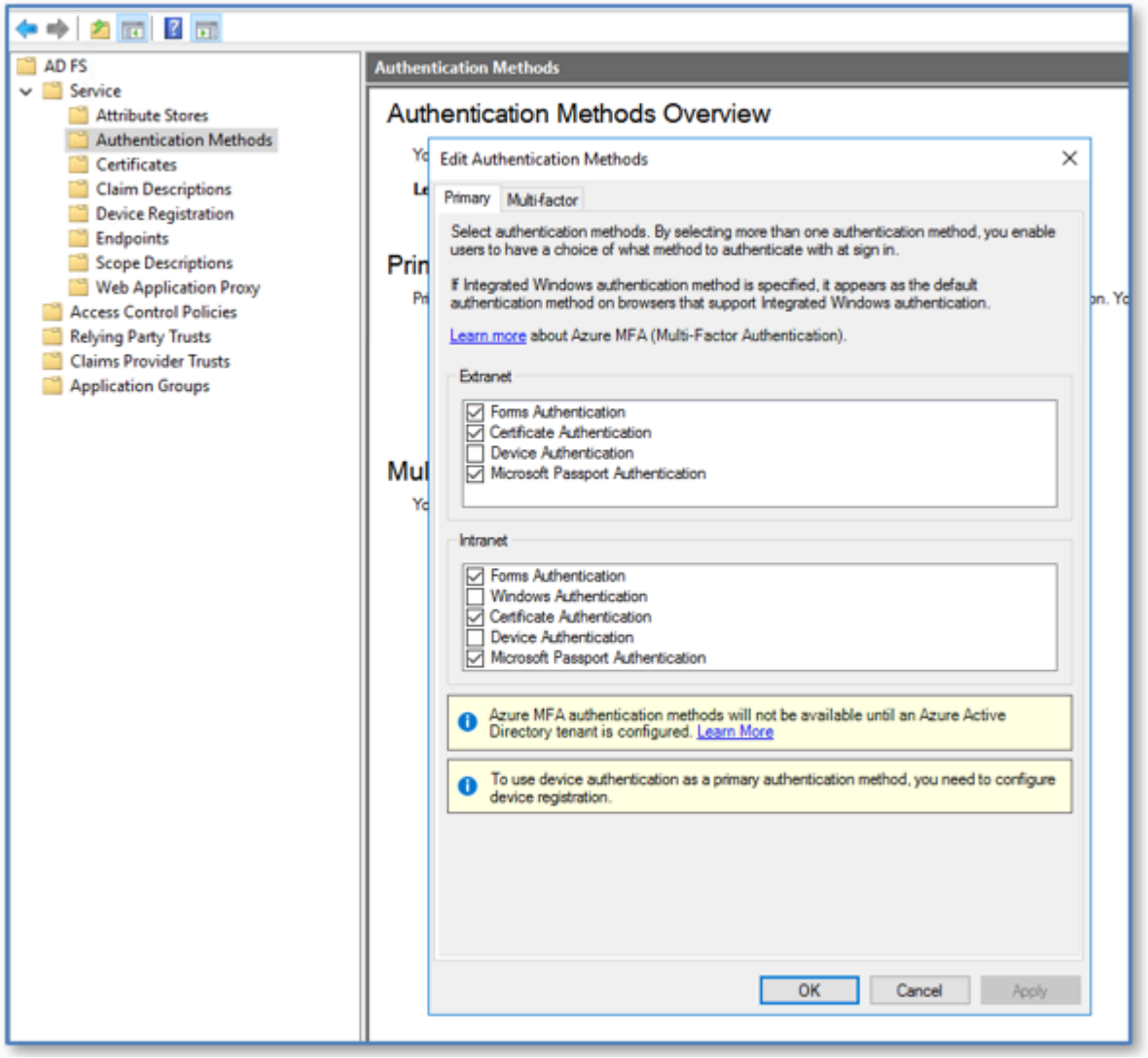
version 10.0.14393 Build 14393

Global Rule

Workcloud Communication depends on Forms-Based authentication.

Open the ADFS Management Console and navigate to **Service** > **Authentication Methods** to open the **Edit Authentication Methods** screen.

Figure 1 Edit Authentication Methods Screen.



The internal and external network methods may be different.

Set **Forms Authentication** as the primary method for the network supporting mobile devices.

NOTE: This is an enterprise-wide setting. Changing the setting may affect existing operations. Check with the System Administrator before modifying.

Configuration Process Overview

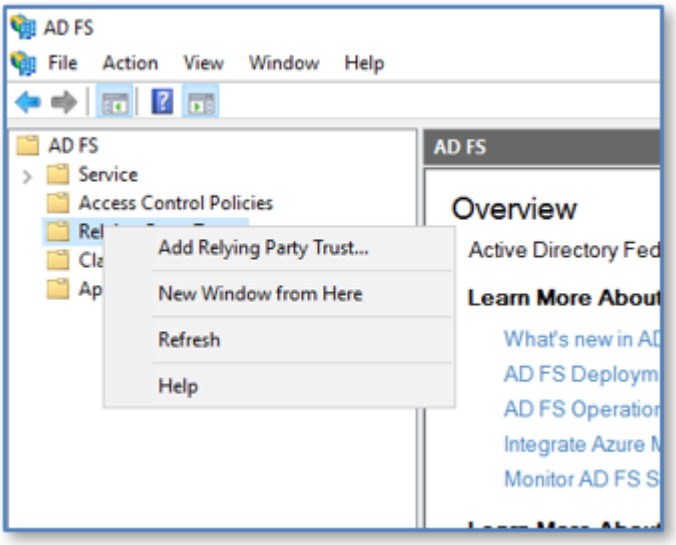
This outline describes the process to properly establish an ADFS configuration to support PTT Pro and Profile Manager OAuth2 authentication. This outline may be helpful for advanced users familiar with ADFS configurations.

- Add a Relying Party Trust
  - Name the Trust
  - Skip the Cert – not required
  - Skip the URL – not required
  - Configure the Trust Identifier
  - Choose the Access Control Policy
- Configure Claims Issuance Transform Rules
  - Add Rule
  - Select Passthrough Claim Rule Template
  - Create Pass Though for Name
  - Create Pass Through for UPN
- Bind the Token Decrypting Cert to Relying Trust
  - Select the Token Decrypting cert
  - Copy Cert to File -- Name and Save base64 cert file type
  - Bind the cert to the Relying Party Trust in Signature
  - Export the Token Signing Cert to be used in the PTT Pro OAuth configuration
- Create Application Groups
  - Add an Application Group
    - Select Standalone / Native application
  - Capture Client ID
  - Redirect = https://localhost
  - Select Standalone Web API
  - Add identifier
  - Select Access Control Policy
  - Add Application Permissions
  - Add Issuance Transform Rules
  - Add Rules
    - Select passthrough Claim Rule Template
    - Create Pass Through for Name
    - Create Pass Through for UPN
  - Apply Access Control policy
  - Config Application Permissions to include Openid and Profile

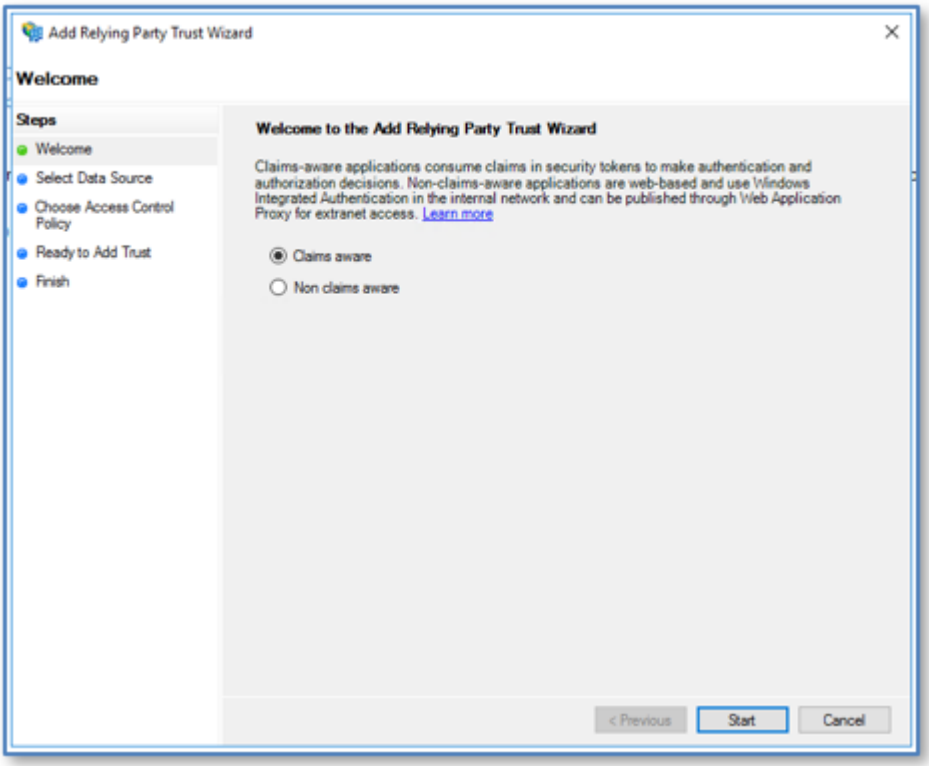
Create a Relying Party Trust

Add a Relying Party Trust using the ADFS Management Console.

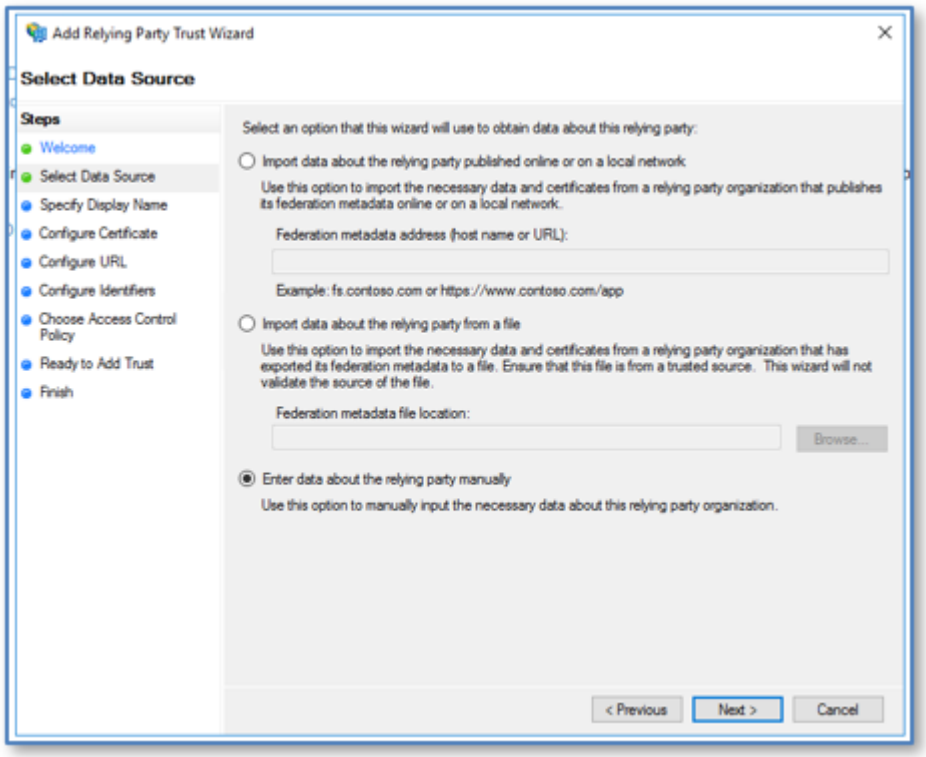
1. Open the **ADFS Management Console**.
2. Select and **Add Relying Party Trust**.



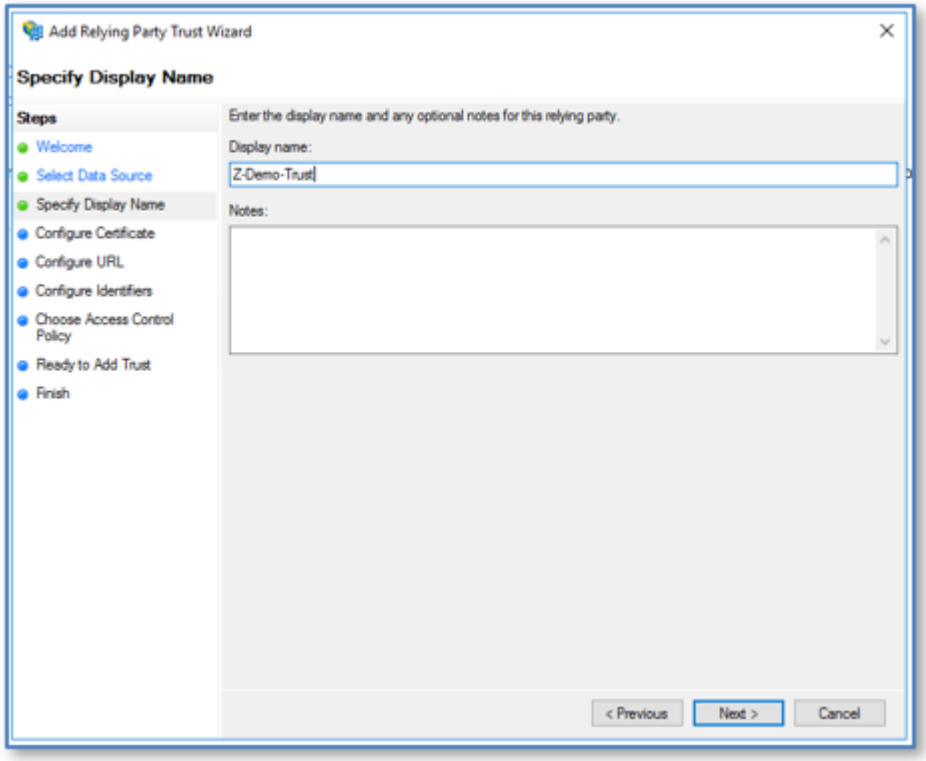
3. Select **Claims aware**.



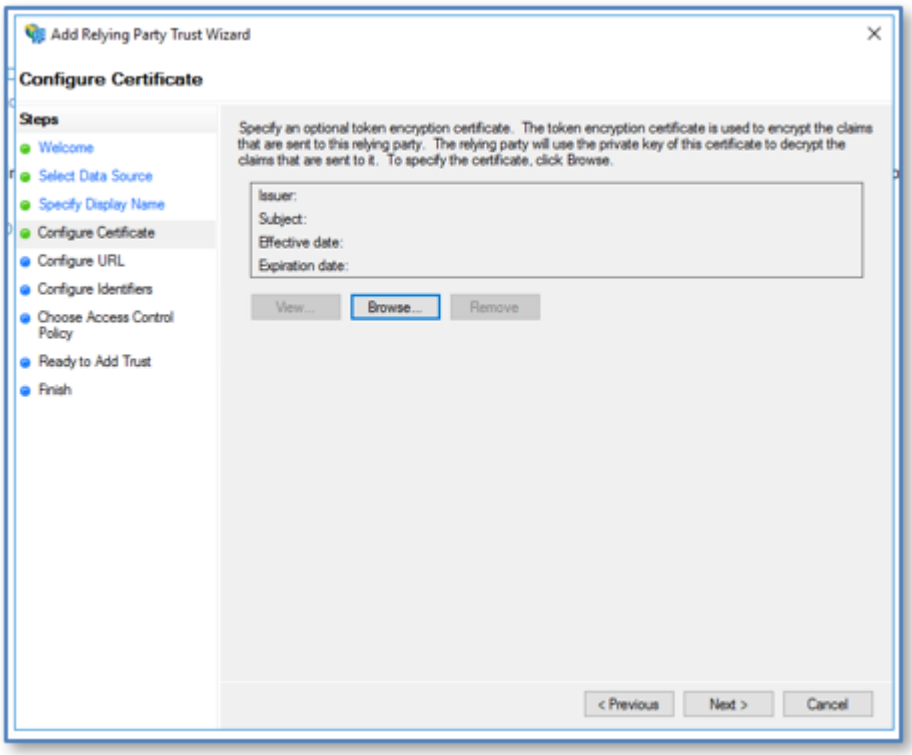
4. Select the option to **Enter the data about the relying party manually**.



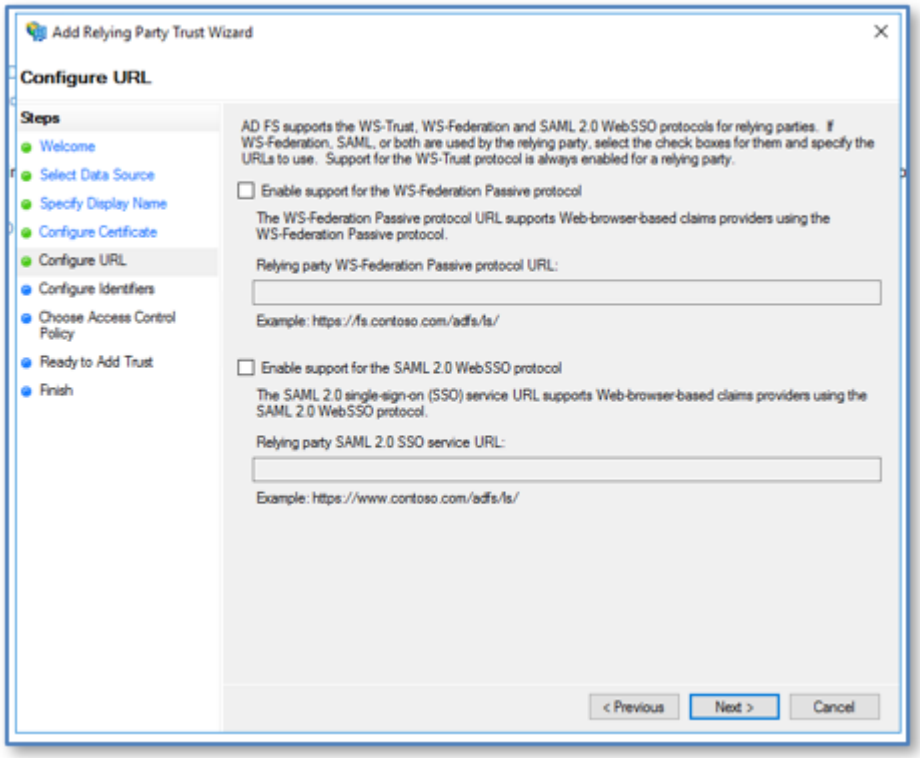
5. Enter a name for the relying party in the **Display name** field.



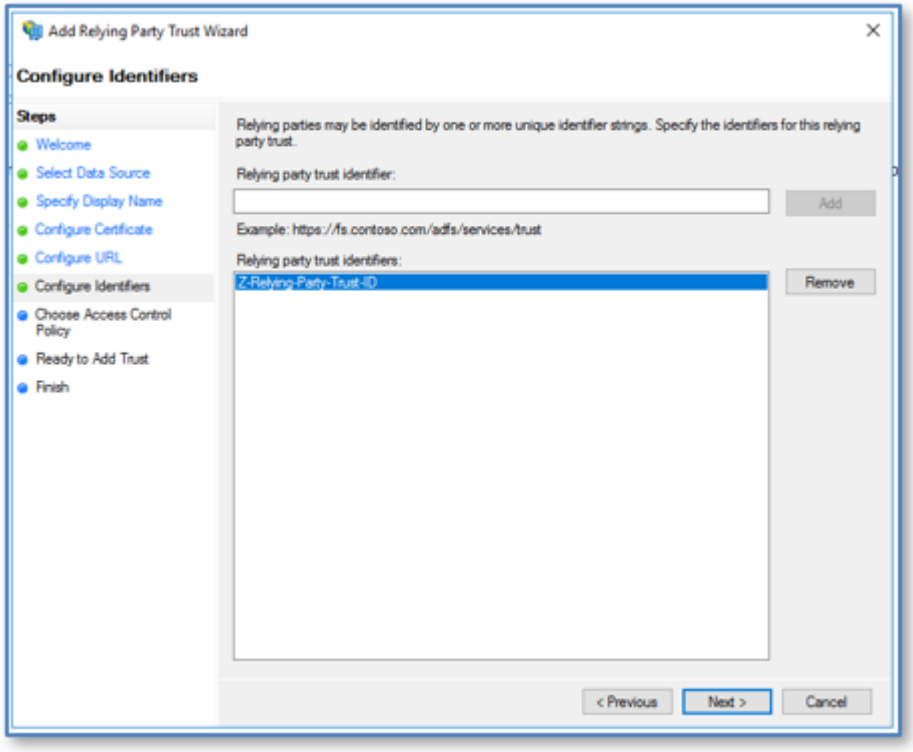
6. Click **Next**.



7. No certificate is required. Click **Next**.

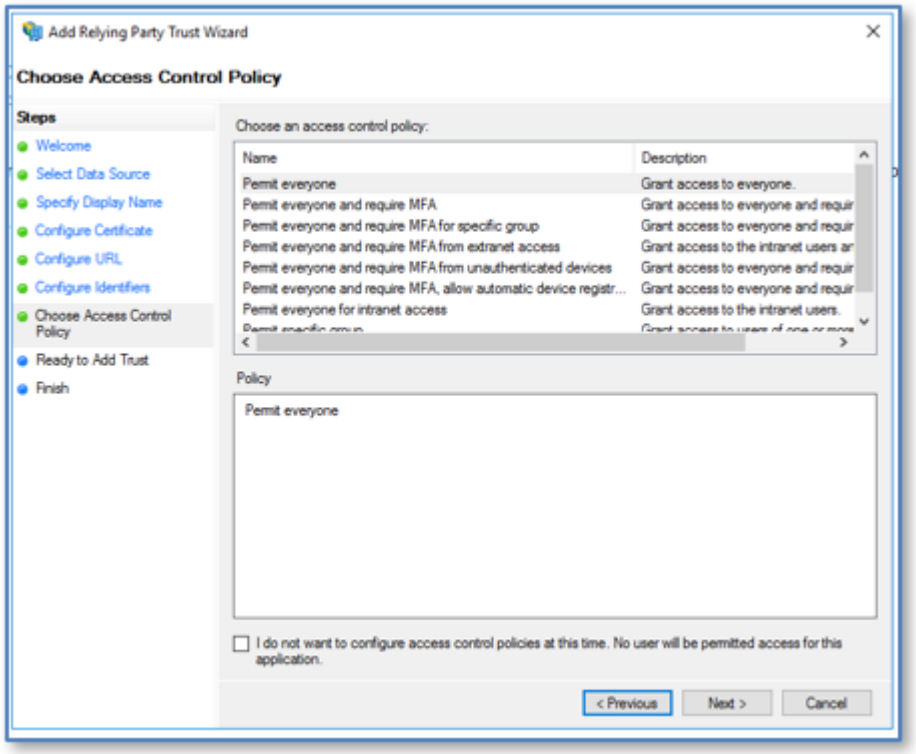


8. No URL is required. Click **Next**.
9. In the **Relying party trust identifier** field, enter a meaningful name for the relying trust, and click **Add**. In this example, the identifier Z-Relying-Party-Trust-ID is used.



10. Click **Next**.

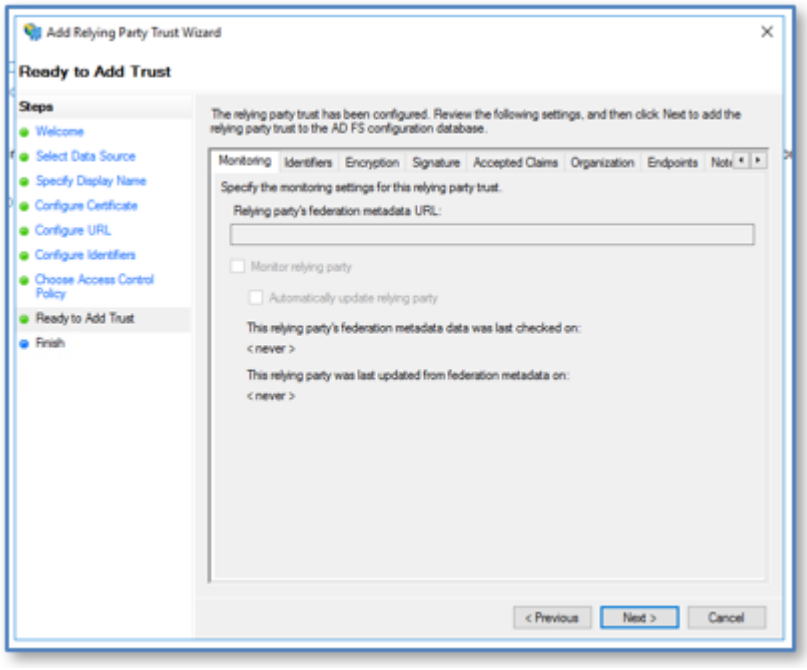
    The wizard advances to **Choose Access Control Policy** screen.



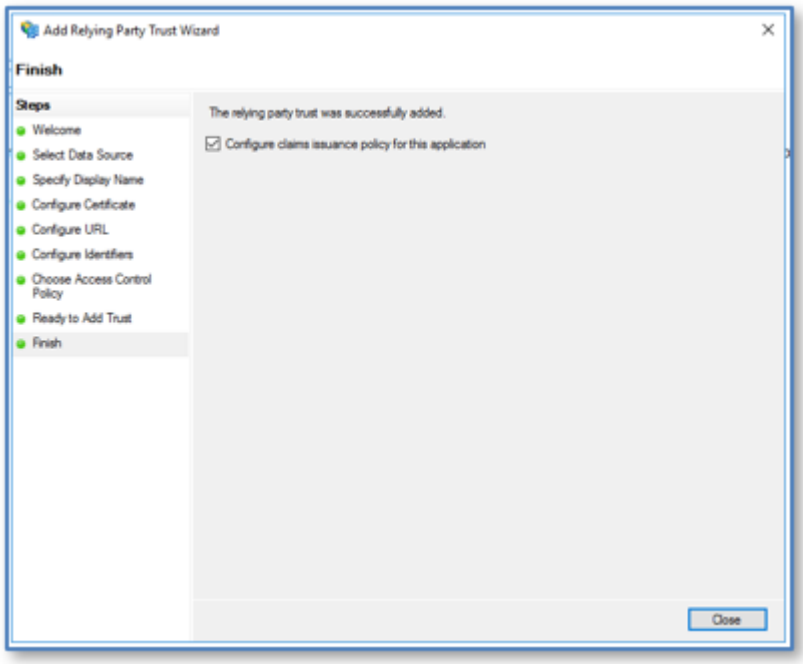
11. In the **Choose an access control policy** field, choose **Permit everyone**.
12. Leave the Access Control Policy option unchecked.

13. Click **Next**.

    The wizard advances to **Ready to Add Trust**. The summary page provides the ability to review all selections made.

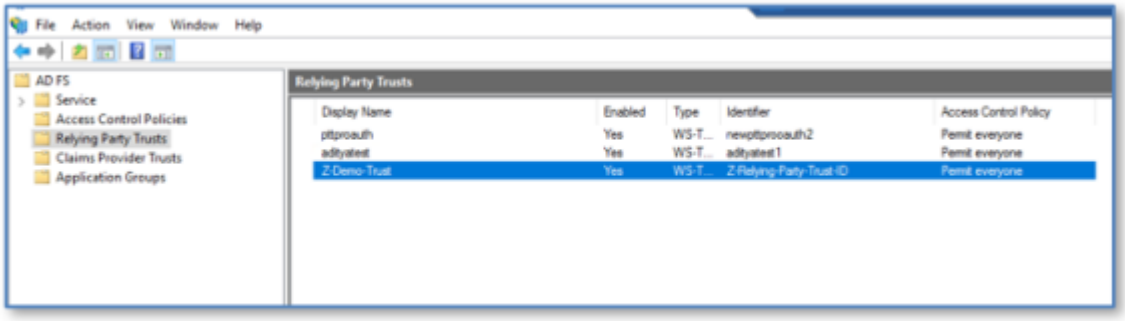


14. Click **Next**.



15. Ensure the **Configure claims issuance policy for this application** box is checked.

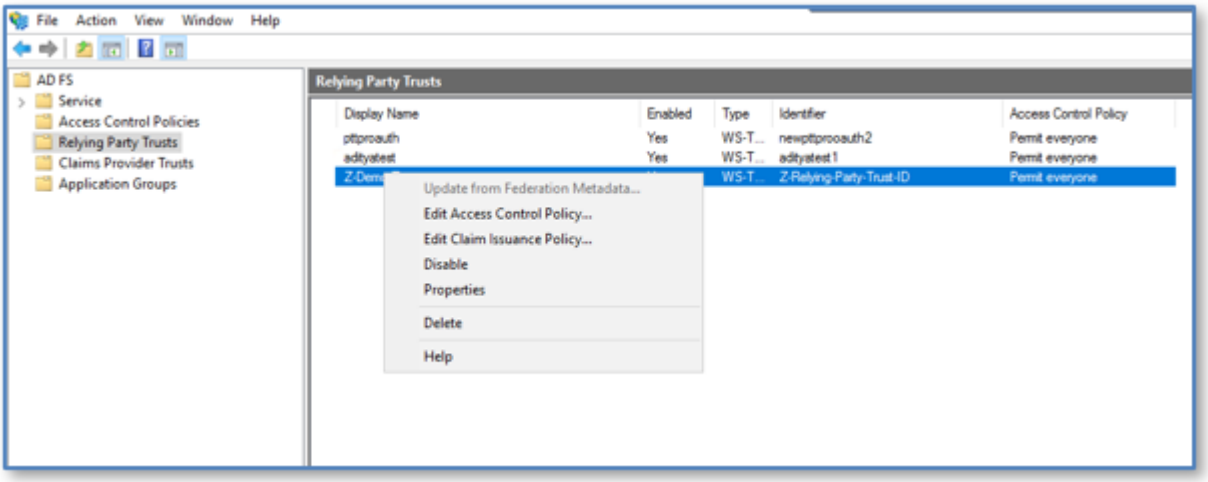
16. Click **Close** to finish. The console returns to the Relying Party Trusts summary page.



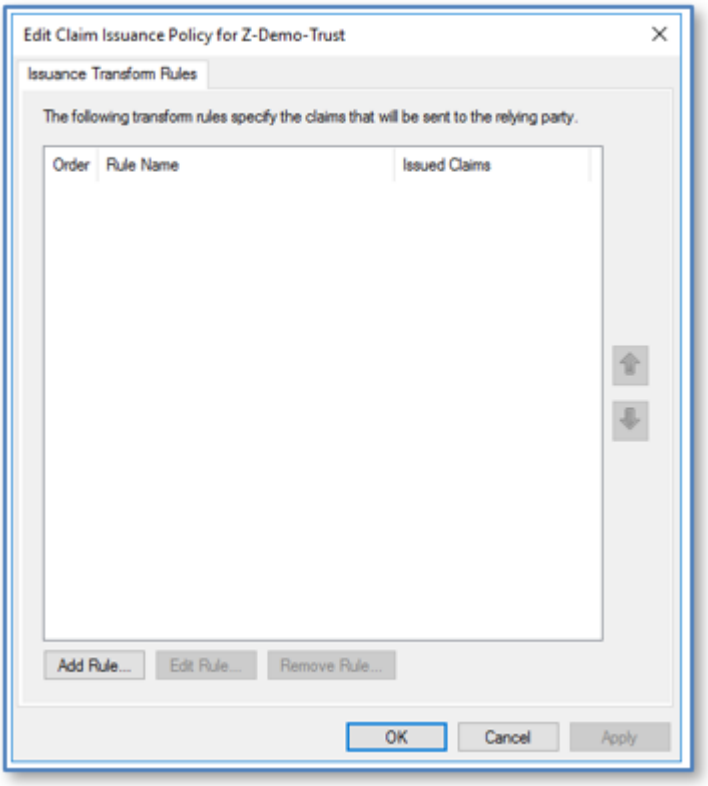
Configure Relying Trust for Claim Issuance Policy

Edit the Claim Issuance Policy of the Relying Trust policy.

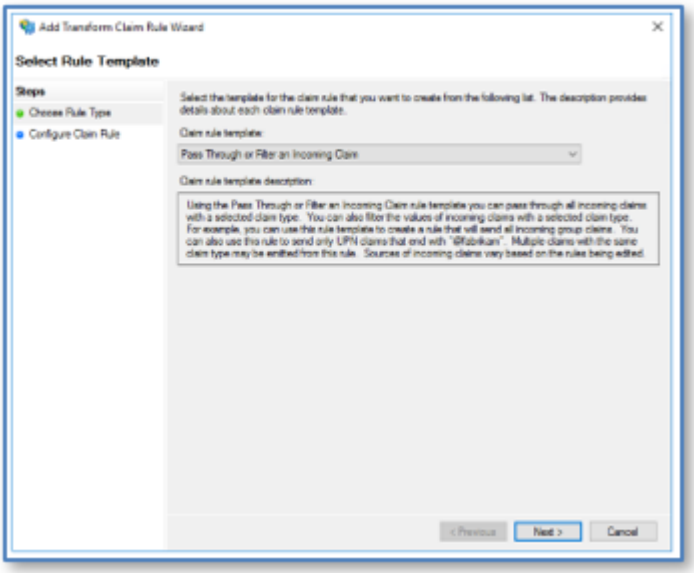
1. Right-click on the new Relying Trust policy and select **Edit Claim Issuance Policy**.



2. Click **Add Rule**.

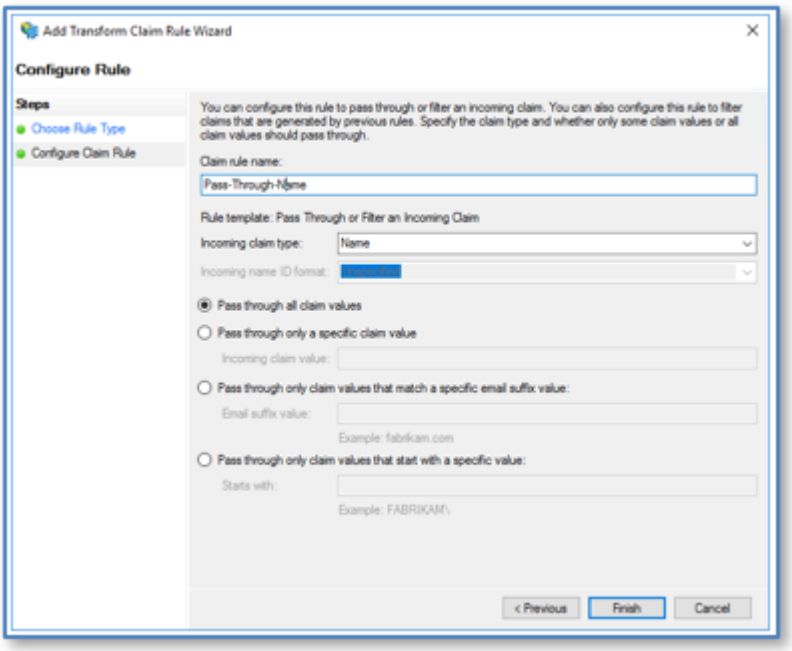


3. In the **Open rule template** drop-down, select **Pass Through or Filter an Incoming Claim**.



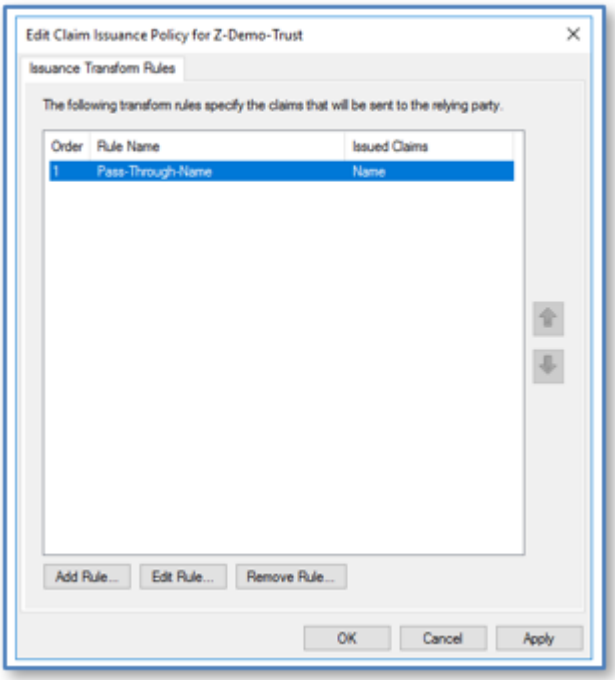
4. Click **Next**.
5. Enter a name in the **Claim rule name** field.
6. Select **Name** from the **Incoming claim type** menu.

7. Select **Pass through all claim values**.

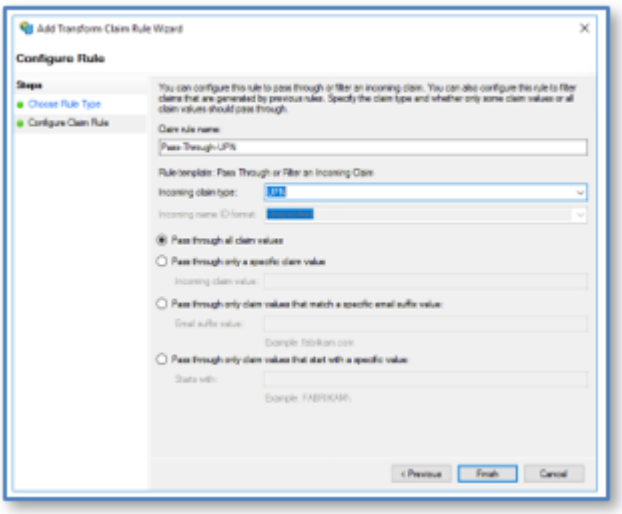


8. Click **Finish**.

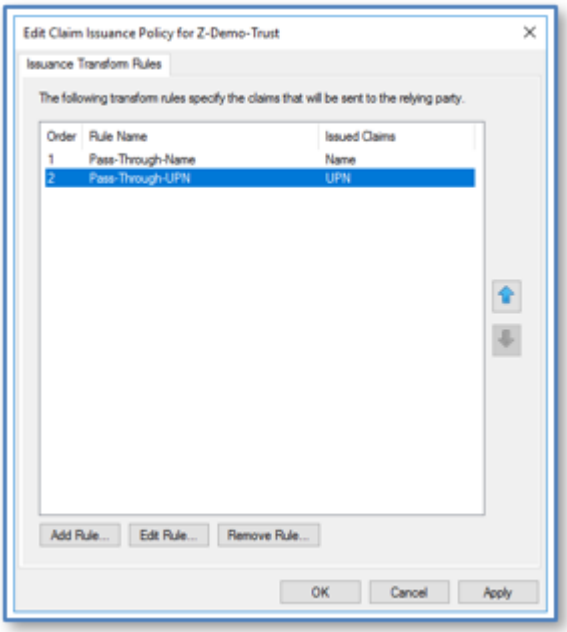
   The console returns to the Insurance Transform Rules screen.



9. Click **Add Rule** to add another Rule for UPN.



10. Click **Finish** when UPN is complete.

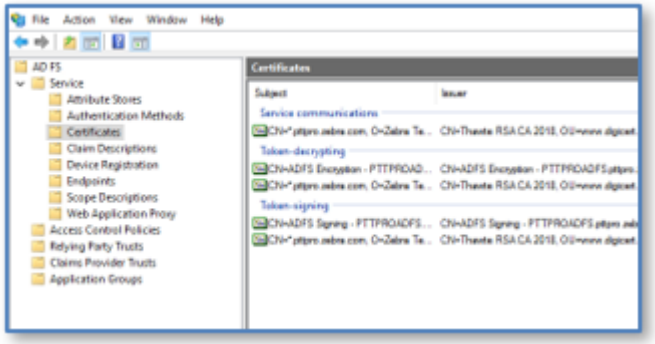


11. Click **Apply** to return to the Relying Trusts listing.

Export the Token Decrypting Certificate

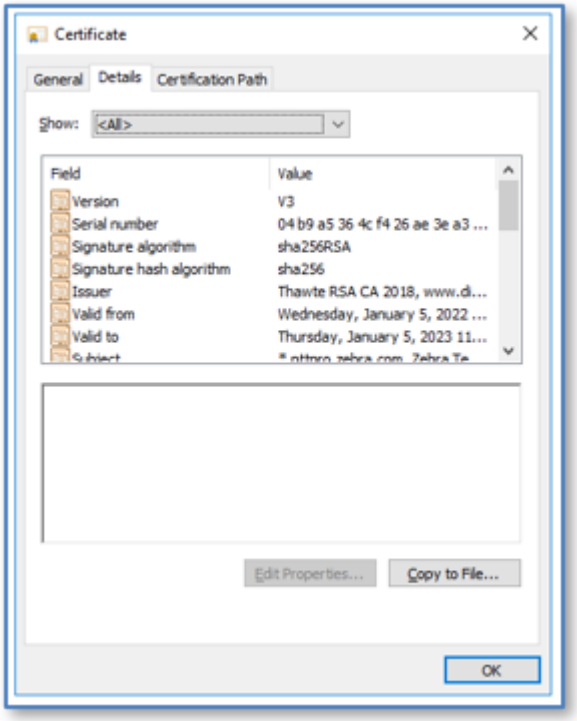
Select the certificate for decrypting the token and export it for later use.

1. Select **ADFS** > **Service** > **Certificates** from the ADFS Management Console view.



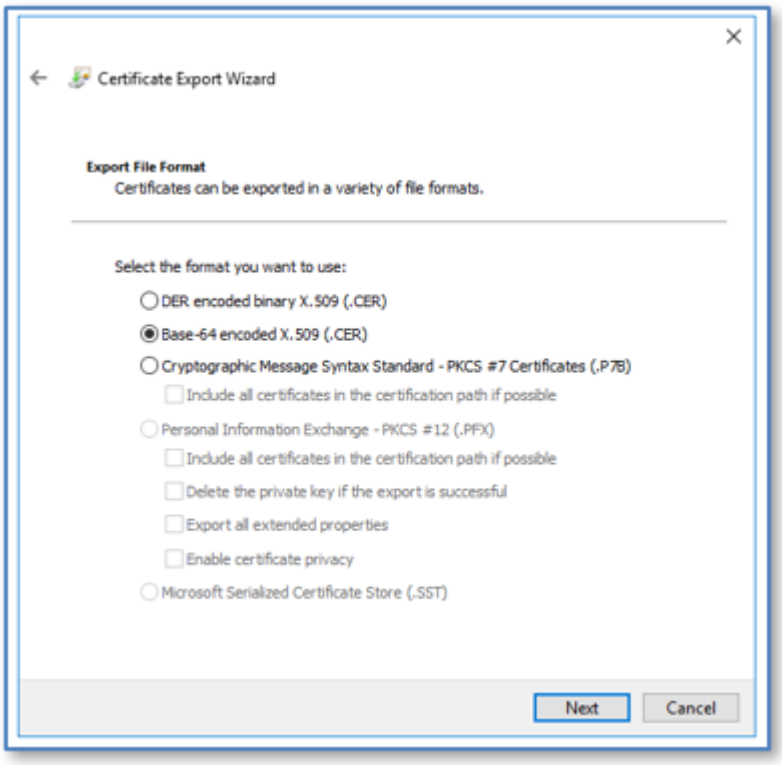
   The **Certificate** window appears.

2. Select the appropriate token-decrypting certificate from the **Certificates** window.



3. Right-click on the desired certificate, click the **Details** tab and copy the certificate to a file on your computer.

4. Use the **Certificate Export Wizard** to export the Token Encryption Certificate to a Base-64 .CER file type.

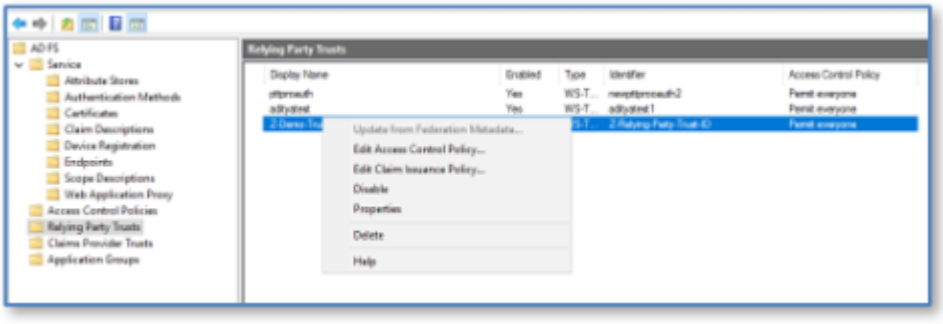


5. Name the exported cert and save it to a location on your computer.
6. Click **Save** to finish.

Bind the Token Decrypting Certificate to the Relying Party Trust

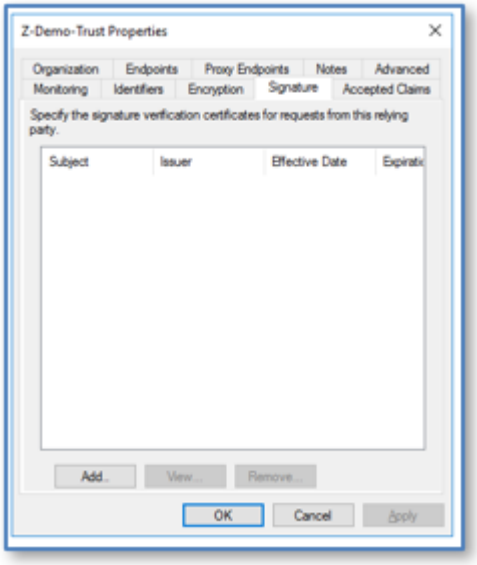
Select the Token-Decrypting Certificate you created earlier and bind it to the Relying Party Trust.

1. Select **Relying Party Trusts** to display the list **Replying Party Trusts**.

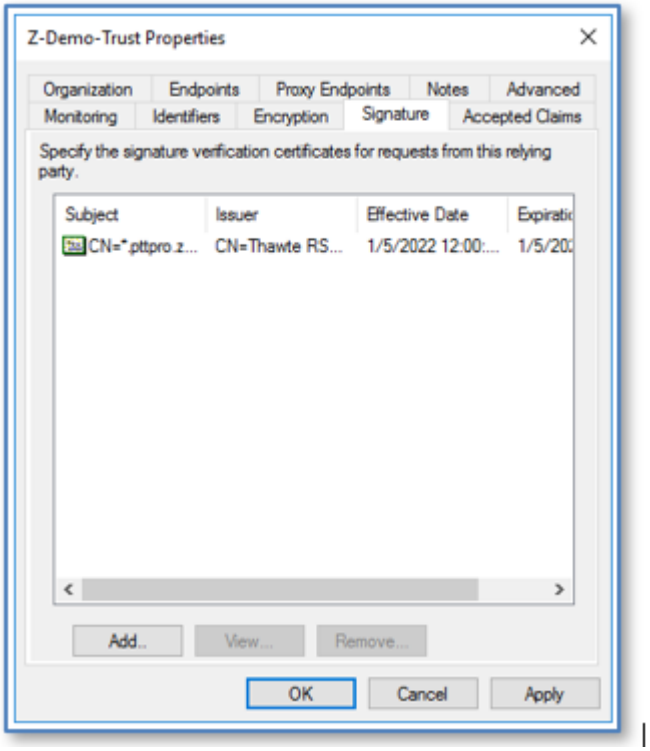


2. Right-click on the trust and select **Properties**.

3. Select the **Signature** tab and click **Add**.



4. Browse for and select the exported Token Encryption Certificate and click **Open** to add the certificate.

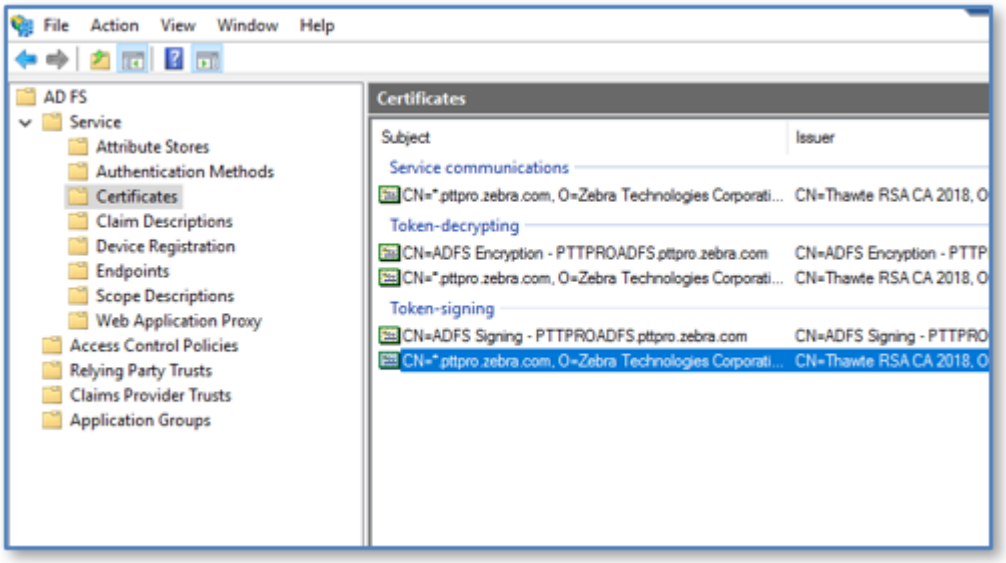


5. Click **Apply** and **OK** to finish.

Export Token Signing Certificate

Export the Token-Signing Certificate using the same method as the Token Decrypting Certificate to a Base-64 .CER file type.

1. Return to **Certificates** and select a valid Token Signing Certificate.



2. Export the selected certificate in that same fashion as the Token Decryption Certificate.

   Save the certificate. You will use the Token Signing Certificate to configure OAuth2 in the PTT Pro Server.

Create the Application Group

When you define the Application Group, the template selection determines whether a Client Secret is used. The process flow below describes two flows. The first process flow steps through a configuration without the Client Secret, and the second process flow shows a configuration with a Client Secret. Workcloud Communication supports both configurations.

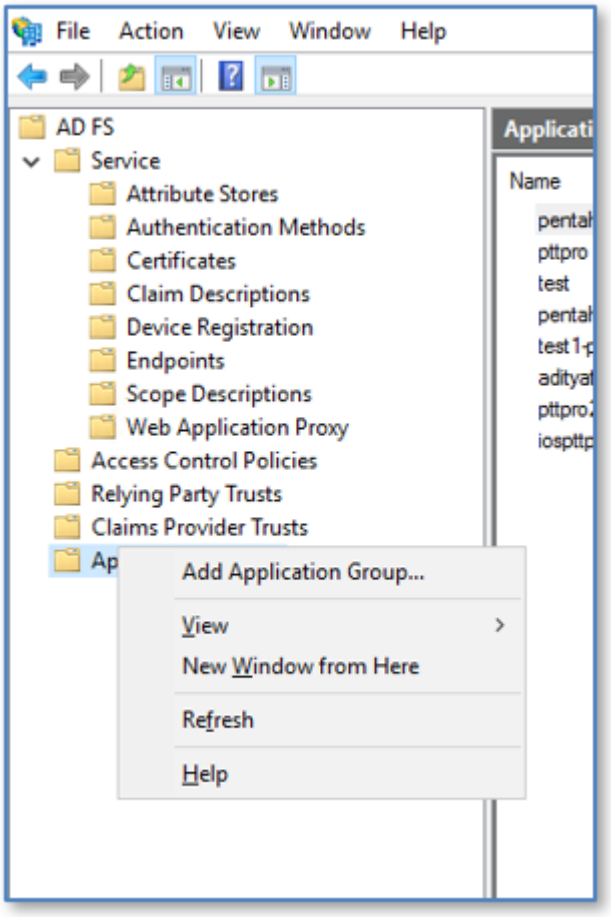
The use of a Client Secret is up to the Customer. If the customer wants to use a Client Secret:

- The Client Secret must be defined in the Profile Manager.
  - If the secret is not populated or is incorrect for the customer in Profile Manager, the user is prompted for credentials but the Profile Client displays an “Unknown Error” message when the credentials are entered.
- The Client Secret must be included in the PTT Pro Client configuration.
  - If the Client Secret is not included in the PTT Pro client JSON configuration file, the client displays a security error when the connection is activated.

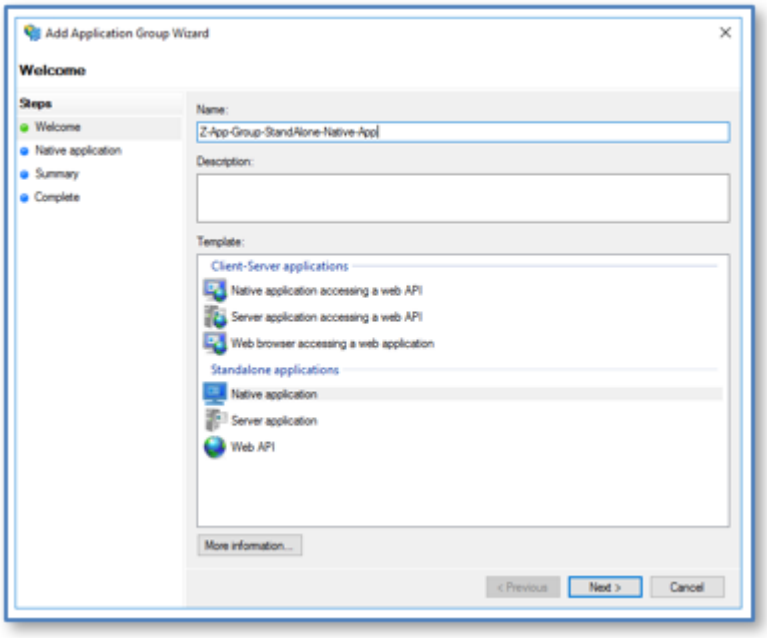
Create a Standalone Application Group

After you create the application group, you need to create a Web API Application Group. If the customer requires a client secret, choose the procedure referenced in the post-requisites.

1. In the ADFS Management Console, right-click on **Application Groups** and select **Add Application Groups**.



2. In the **Templates** list, select **Native Application** from the **Stand Alone Application** list.

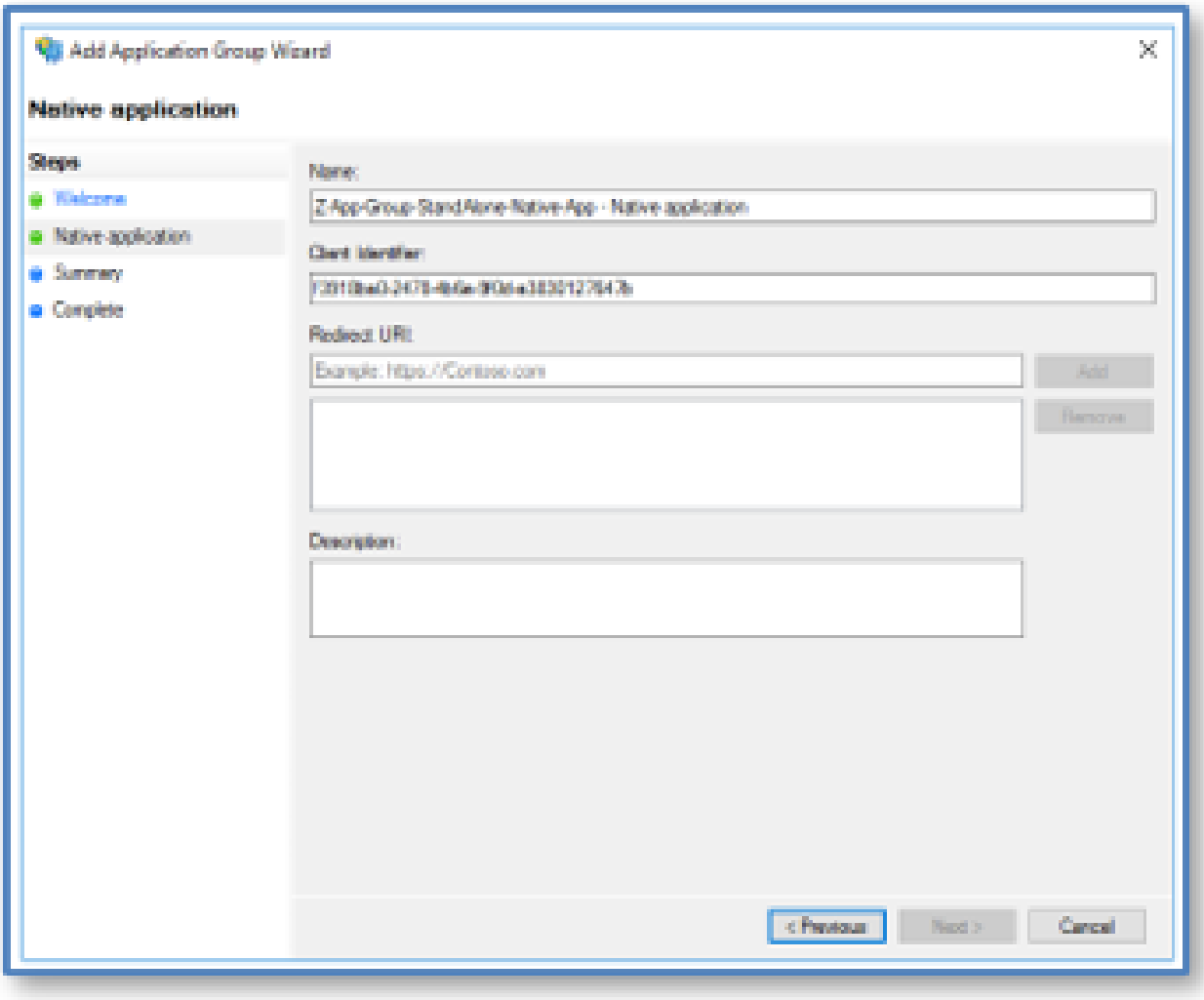


3. Enter a meaningful name and click **Next**.

   The Client Identifier is displayed in the **Add Application Group Wizard**.

   a) Copy the Client Identifier so that you can paste it into a validation application such as Postman.

   b) Enter `https://localhost` in the **Redirect URL** field.



4. Click **Next** and **Close**.

The next depends on whether the customer requires a client secret:

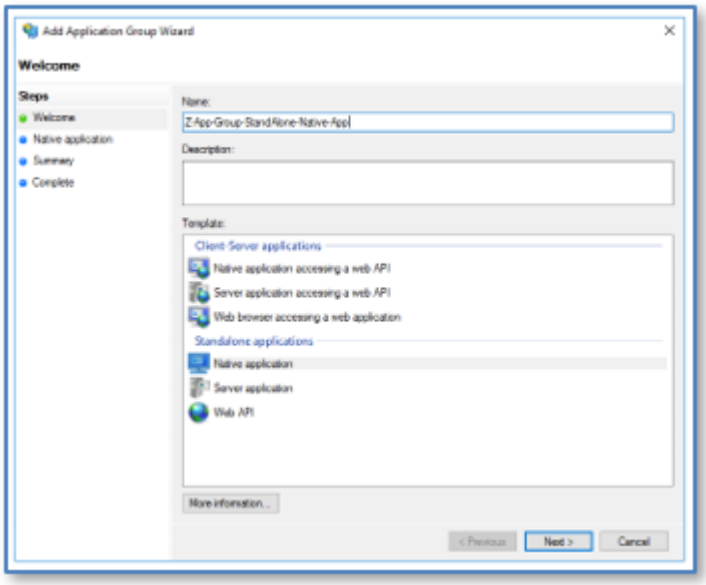
- If a client secret is not required, proceed to Create Standalone Web API Application Group on page 26.
- If a client secret is required, proceed to Create a Web API Application Group with a Client Secret on page 29.

Create Standalone Web API Application Group

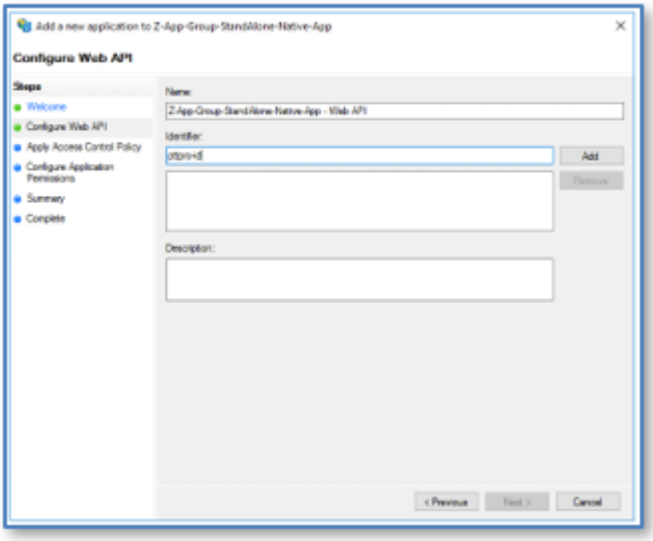
Use this task to create an application group that does not use a shared secret.

1. Open the ADFS Management Console

2. Right-click the **Application Group** and select **Add Application Group**.
3. In the **Templates** list, select **Web API** from the **Stand Alone Application** list.



4. Configure the Web API.



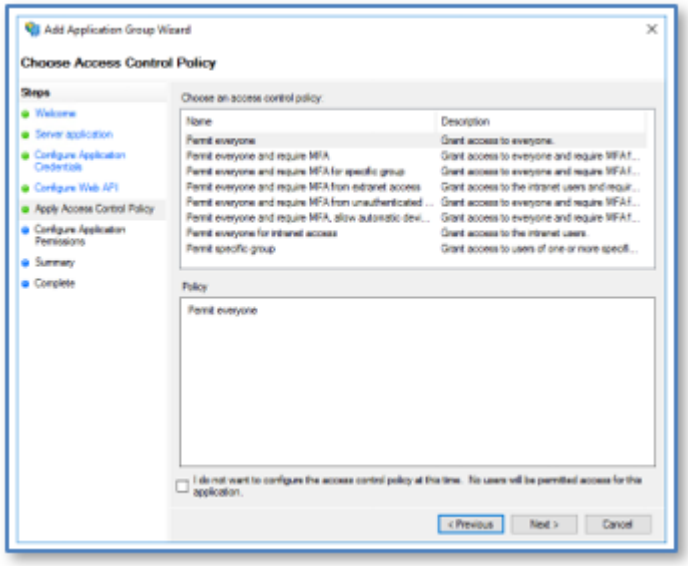
a) Enter a name in the **Name** field.

The name is appended to the Access URL to access the ADFS Application Group, as in `https://<server_name>/adfs/oauth2/authorize?resource=pttpro-id` in this example.

b) Click **Add**.

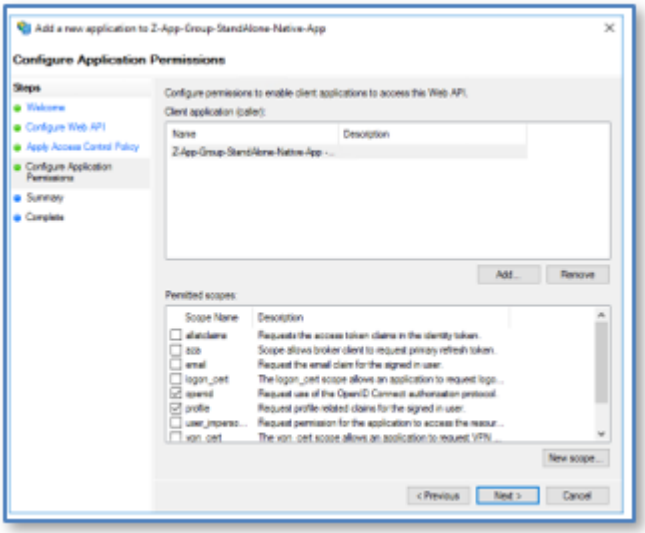
5. Click **Next** to advance.

   The Access Control Policy defaults to Permit Everyone.



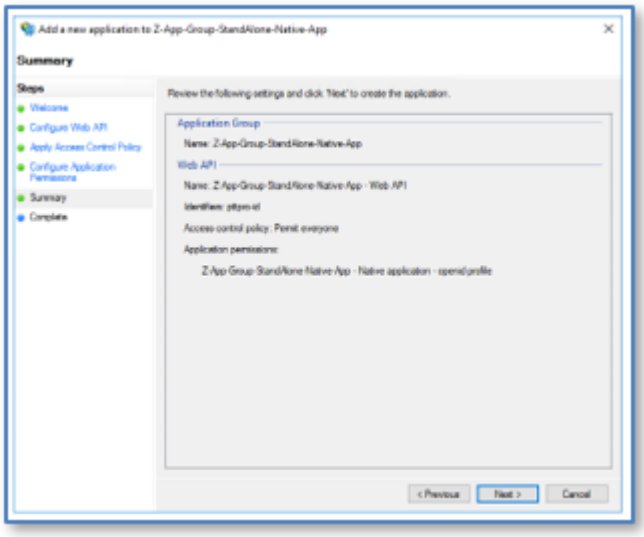
6. Click **Next** to advance.

   Under **Permitted Scopes**, the application permissions default is **openid**. Also ensure **profile** is checked.



7. Click **Next** to advance.

   A summary is shown.



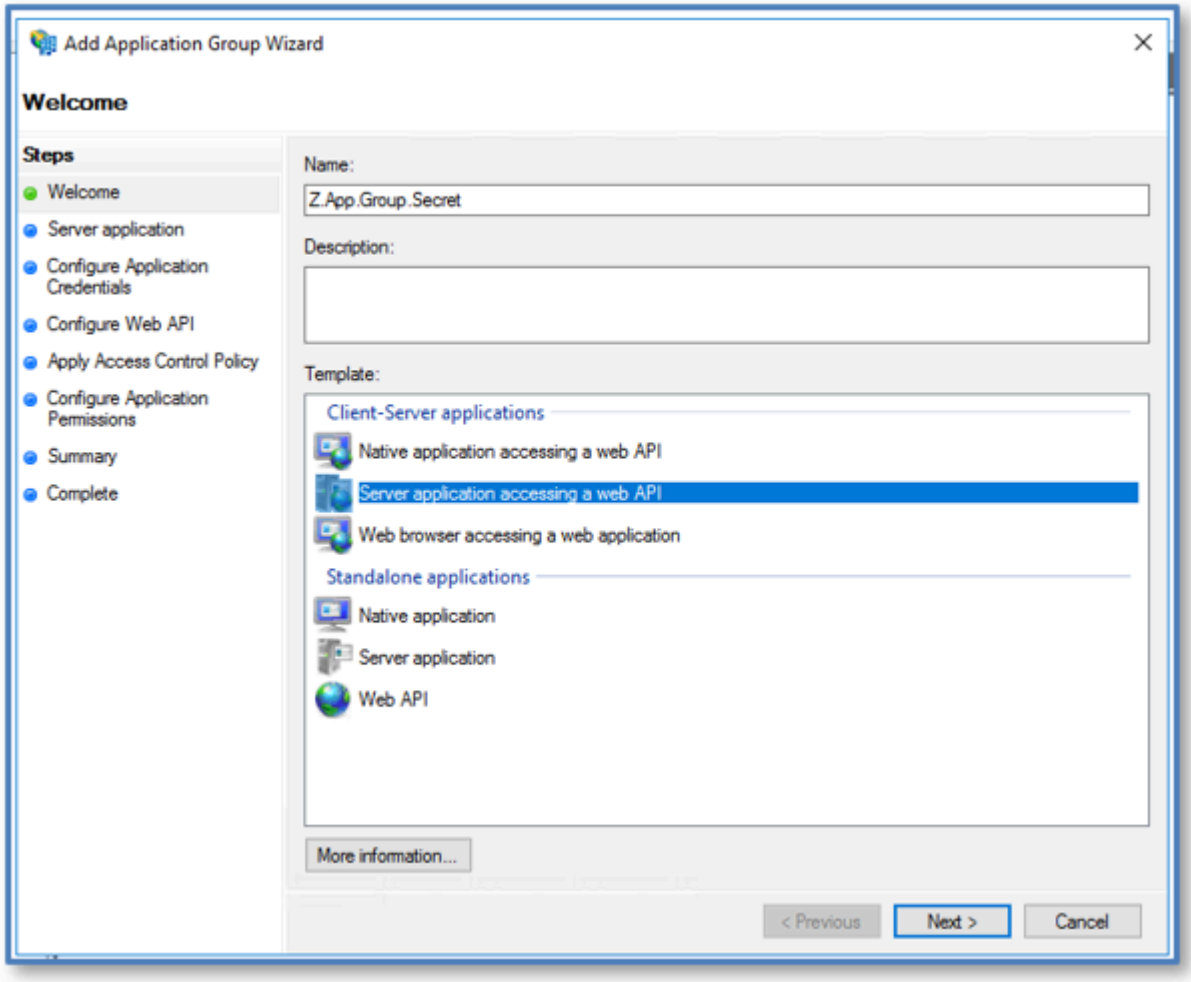
8. Click **Next** to close and finish.

Create a Web API Application Group with a Client Secret

Use this task to create an application group that uses a shared secret.

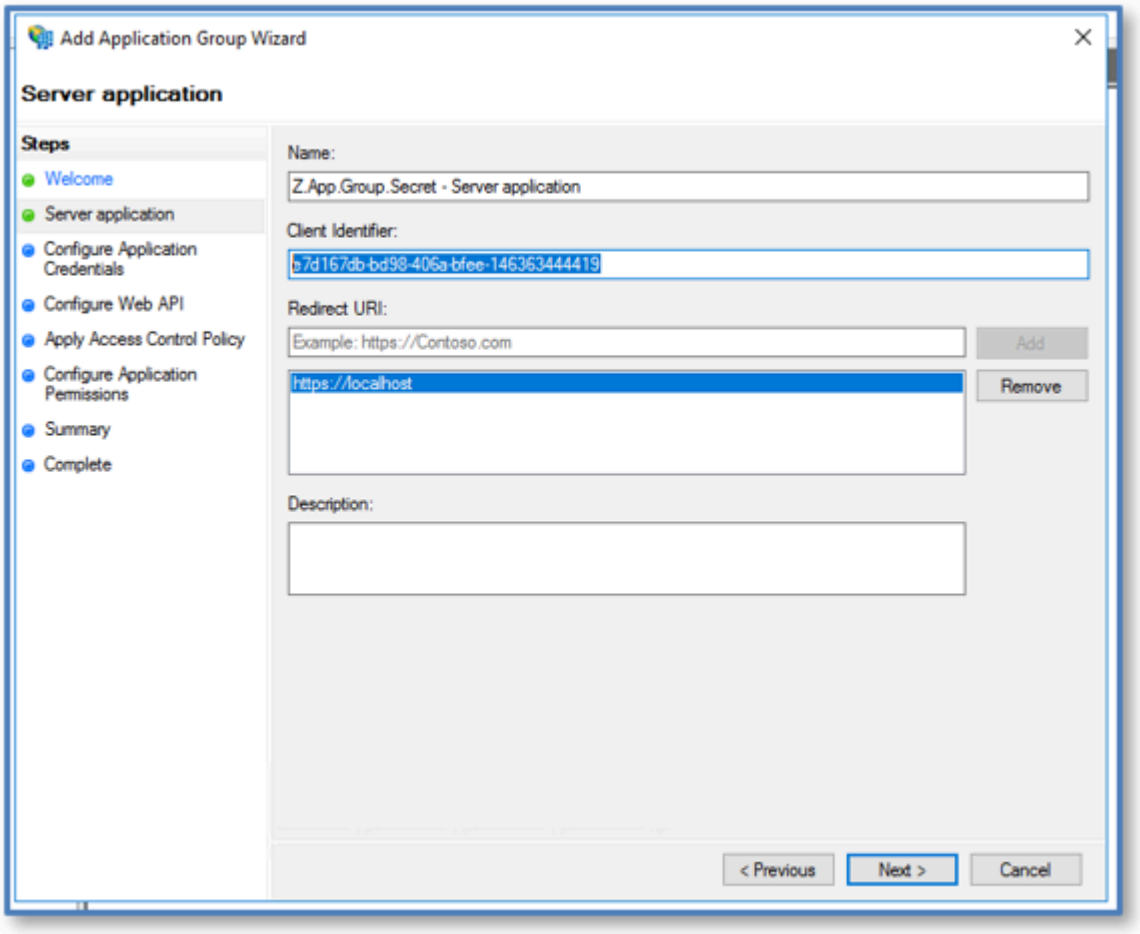
If the customer uses a Client Secret, the Application Group configuration is different.

1. Select **Server Application accessing a web API** from **Client-Server Applications** instead of **Native Application** from **Standalone applications**.



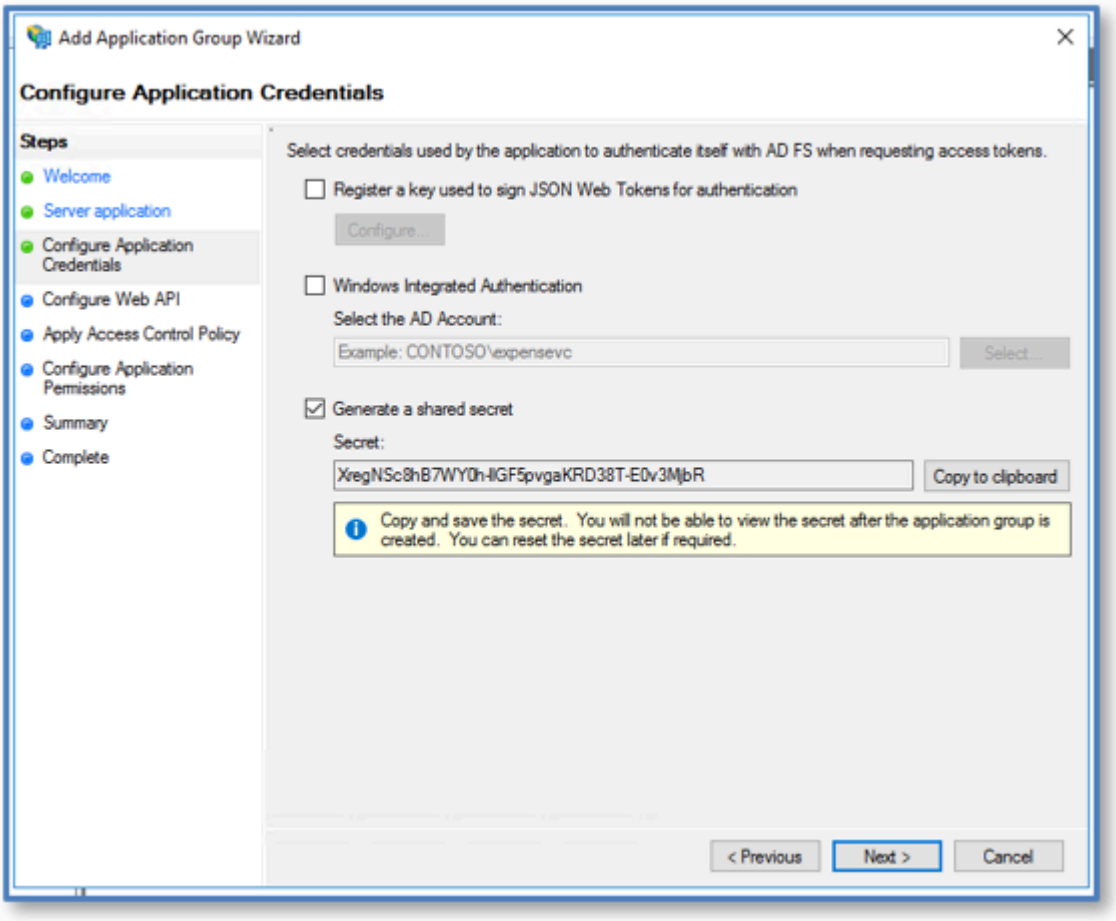
2. Enter a meaningful name in the **Name** field.

   The Client Identifier is automatically generated. Copy the Client Identifier for later use.



3. Enter `https://localhost` in the **Redirect URI** field.
4. Click **Next** to advance.

5. Select **Generate a shared secret** in the **Configure Applications Credentials** screen.

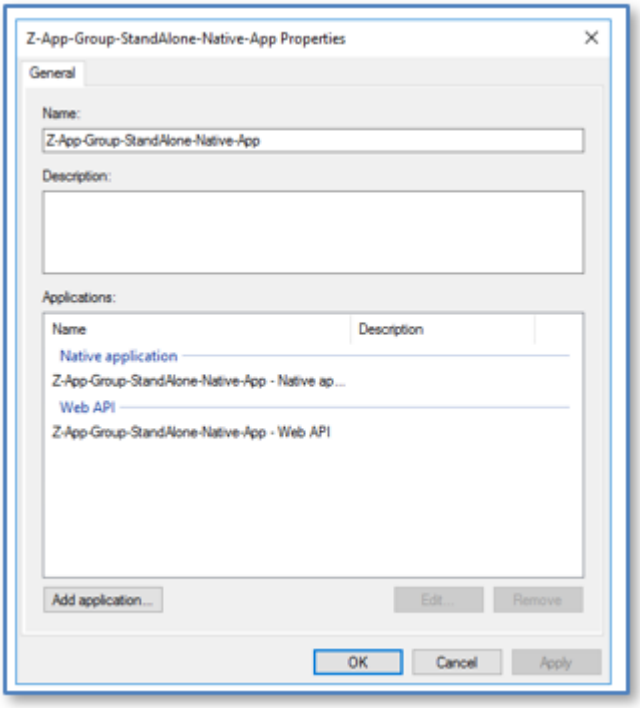


6. Copy and save the secret that is generated. The shared secret will not be shown again.
7. Complete the creation of the application group as previously described Create a Standalone Application Group on page 24.

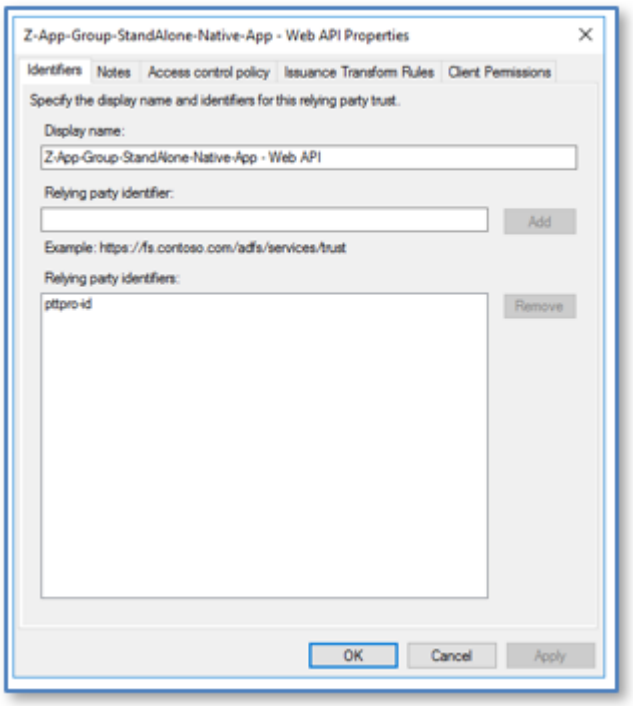
Add Claims to the Application Group

After creating the Application Group, the Claims Issuance may not be automatically prompted for and must be added.

1. Right-click on the Application Group and select **Properties**.

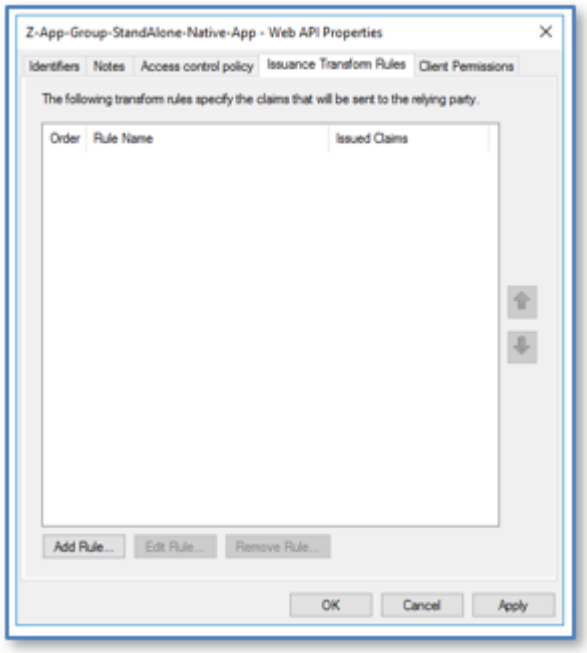


2. Select the Web API application and click **Edit**.



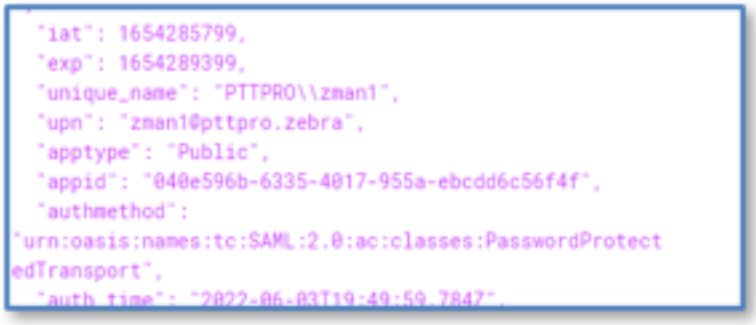
3. Select the **Issuance Transform Rules** tab.

4. Click **Add Rule** and enter two rules, one for Name and one for UPN.

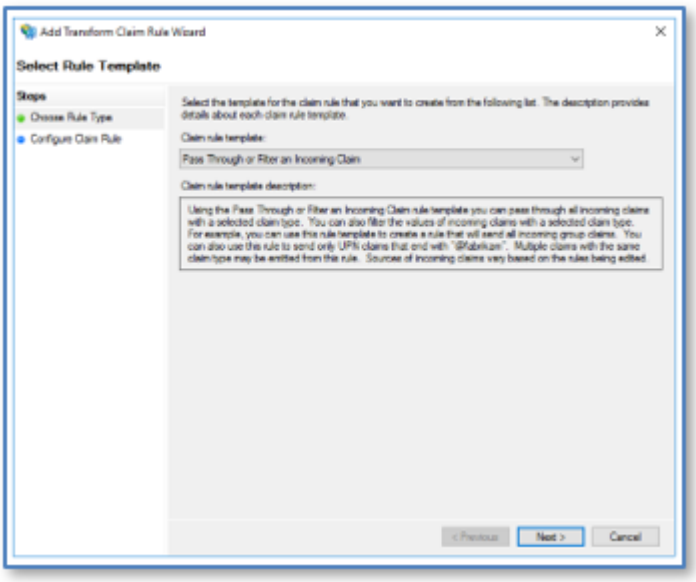


Using two rules provides flexibility. The required Transform Rule is based on the content of the Access Token.

In this snippet, we see both the UPN and the Name (unique_name) in the token. Refer toExamine the Returned Access Token on page 43 for more details.

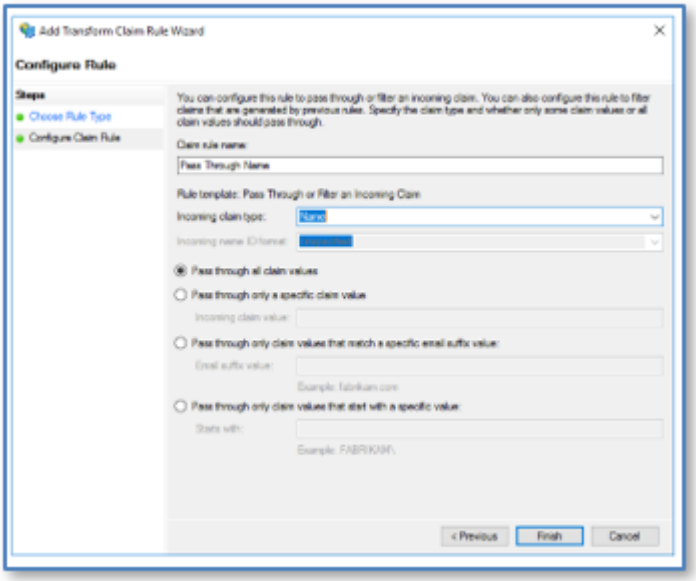


5. Select **Pass Through or Filter an Incoming Claim** from the **Open rules template** drop-down menu.



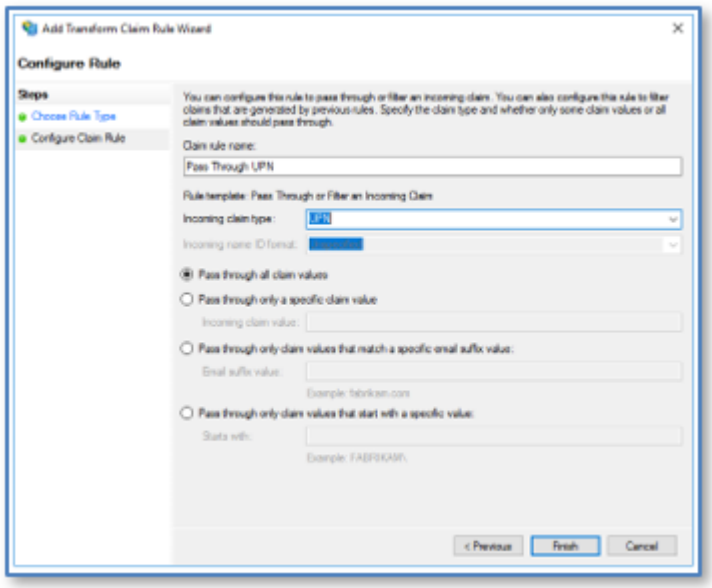
6. Click **Next**.
7. Enter a name in the **Claim rule name** field and select **Name** from the **Incoming claim type** drop-down menu.

   Ensure that **Pass through all claim values** is selected.

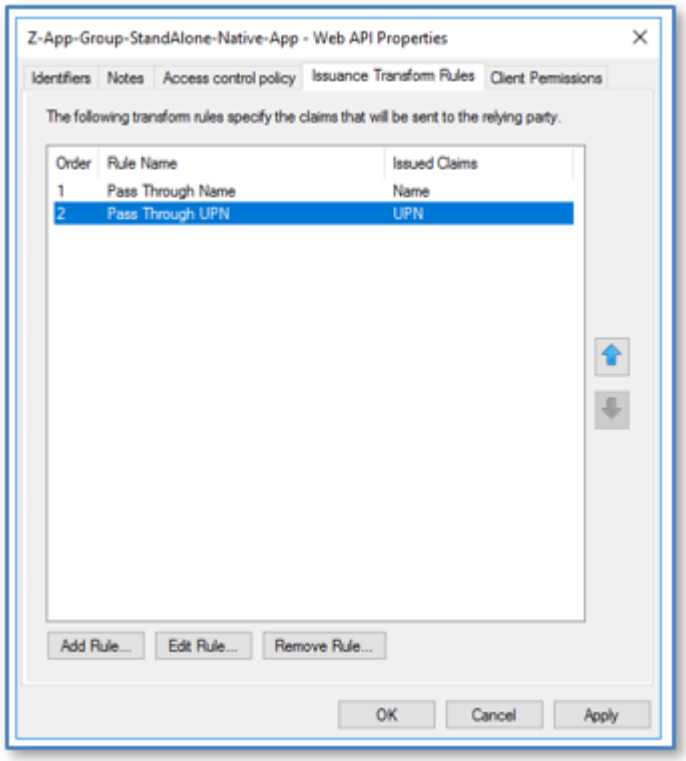


8. Click on **Finish** to return to the **Add Transform Claim Rule Wizard**.

9. Repeat these steps to add a Pass Through UPN claim rule.



10. Click **Finish** and **Apply** to display a list of the rules.



11. Click on **Apply** and **OK** to finish.

If the procedures under the ADFS Setup Flows were followed, the ADFS server support for the PTT Pro server and Profile Manager should now be established.

The validation procedures provide tools and methods to help validate the configuration settings.

Updating the Access and Refresh Token Lifespans

You can update the lifespan for access tokens and refresh tokens according to the requirements of your business. The commands listed below require administrator privileges and must be run in Powershell.

NOTE: The access token lifespan must be less than the refresh token lifespan.

1. Run the following command to issue refresh tokens. If refresh tokens are issued, you can skip this step.

```
set-AdfsRelyingPartyTrust -TargetName "RelyingPartyTrust_name" -
IssueOAuthRefreshTokensTo AllDevices
```

   Replace *RelyingPartyTrust_name* with the address of the ADFS server.

2. Set the access token lifespan. This value is at the Relying Party Trust level.

```
set-AdfsWebApiApplication -TargetIdentifier Identifier_name -
tokenlifetime timeInMin
```

```
set-AdfsWebApplicationProxyRelyingPartyTrust -TokenLifetime timeInMin
```

```
set-AdfsRelyingPartyTrust -TargetName RelyingPartyTrust_name -
TokenLifeTime timeInMin
```

   Replace *Identifier_name*, *timeInMin*, and *RelyingPartyTrust_name* with values appropriate for your environment. The value of *timeInMin* is in minutes.

3. Set the refresh token lifespan. This is a global setting.

```
set-AdfsProperties -ssolifetime timeInMin
```

   Replace the value of *timeInMin* with the refresh token lifespan. The value is in minutes.

Validating the Configuration

After the ADFS server is configured, validate the configuration to ensure the PTT Pro and Profile Manager servers operate as expected.

The following tools can be used to validate the configuration.

Postman
Postman is an API platform for building and using APIs. https://www.postman.com/downloads/

JWT Token validator
JSON Web Tokens are an open, industry standard RFC 7519 method used to represent claims securely between two parties. JWT.IO allows you to decode, verify, and generate a JSON Web Token. https://jwt.io

JSONLint
Validates the structure of a JSON file. JSONLint validates and reformats JSON, a lightweight data-interchange format. Copy and paste, directly type, or input a URL in the editor and let JSONLint tidy and validate your messy JSON code. https://jsonlint.com

KeyCDN Cert Checker
KeyCDN is a web-based certificate checker. tools.keycdn.com

The Well-Known URL

When the Well-Known URL is pasted into a browser, it returns a JSON listing of the OpenID/OAuth endpoints, supported scopes and claims, public keys used to sign the tokens, and other details.

Clients can use this information to construct a request to the OpenID server. The example below is the response returned from the ADFS server for the Well-Known URL.

Figure 2 Response for Well-Known URL

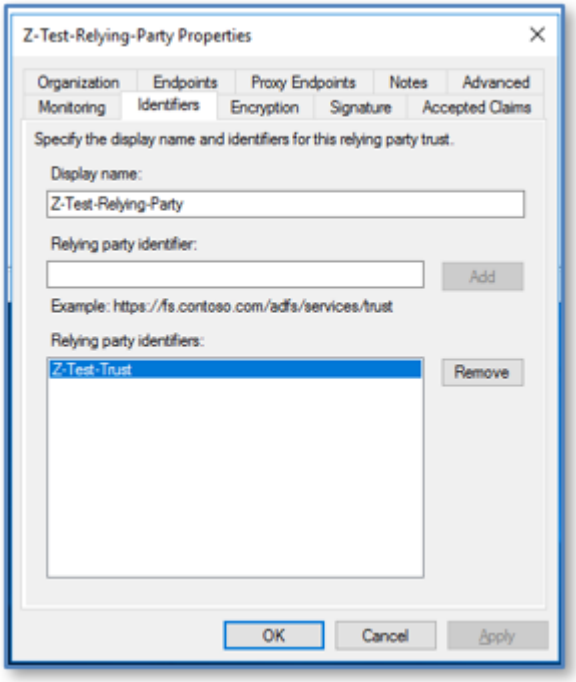
```
{
    issuer: https://PTTPRO-ADFS.pttpro.zebra.com/adfs,
    authorization_endpoint: https://pttproadfs.pttpro.zebra.com/adfs/oauth2/authorize/,
    token_endpoint: https://pttproadfs.pttpro.zebra.com/adfs/oauth2/token/,
    jwks_uri: https://pttproadfs.pttpro.zebra.com/adfs/discovery/keys,
  ▾ token_endpoint_auth_methods_supported: [
        "client_secret_post",
        "client_secret_basic",
        "private_key_jwt",
        "windows_client_authentication"
    ],
  ▾ response_types_supported: [
        "code",
        "id_token",
        "code id_token",
        "id_token token",
        "code token",
        "code id_token token"
    ],
  ▾ response_modes_supported: [
        "query",
        "fragment",
        "form_post"
    ],
  ▾ grant_types_supported: [
        "authorization_code",
        "refresh_token",
        "client_credentials",
        "urn:ietf:params:oauth:grant-type:jwt-bearer",
        "implicit",
        "password",
        "srv_challenge",
        "urn:ietf:params:oauth:grant-type:device_code",
        "device_code"
    ],
  ▾ subject_types_supported: [
        "pairwise"
    ],
  ▾ scopes_supported: [
        "winhello_cert",
        "vpn_cert",
        "email",
        "allatclaims",
        "profile",
        "logon_cert",
        "user_impersonation",
        "aza",
        "openid"
    ],
  ▾ id_token_signing_alg_values_supported: [
        "RS256"
    ],
  ▾ token_endpoint_auth_signing_alg_values_supported: [
        "RS256"
    ],
```


The response includes the two URLs required by the PTT Pro server to complete the OAuth configuration: the Access URL and the Token URL. Examine the Well-Known response to find the Authorization Endpoint and the Token Endpoint.

- Authorization endpoint: `https://pttproadfs.pttpro.zebra.com/adfs/oauth2/authorize`

- Token endpoint: `https://pttproadfs.pttpro.zebra.com/adfs/oauth2/token`

Once the Authorize URL has been identified, resources will need to be added to the URL.



In our example, the Identifier for the Relying Party Trust is `Z-Test-Trust`, and the PTT Pro server authorization URL becomes:

```
https://pttproadfs.pttpro.zebra.com/adfs/oauth2/authorize?resource=Z-Test-Trust
```

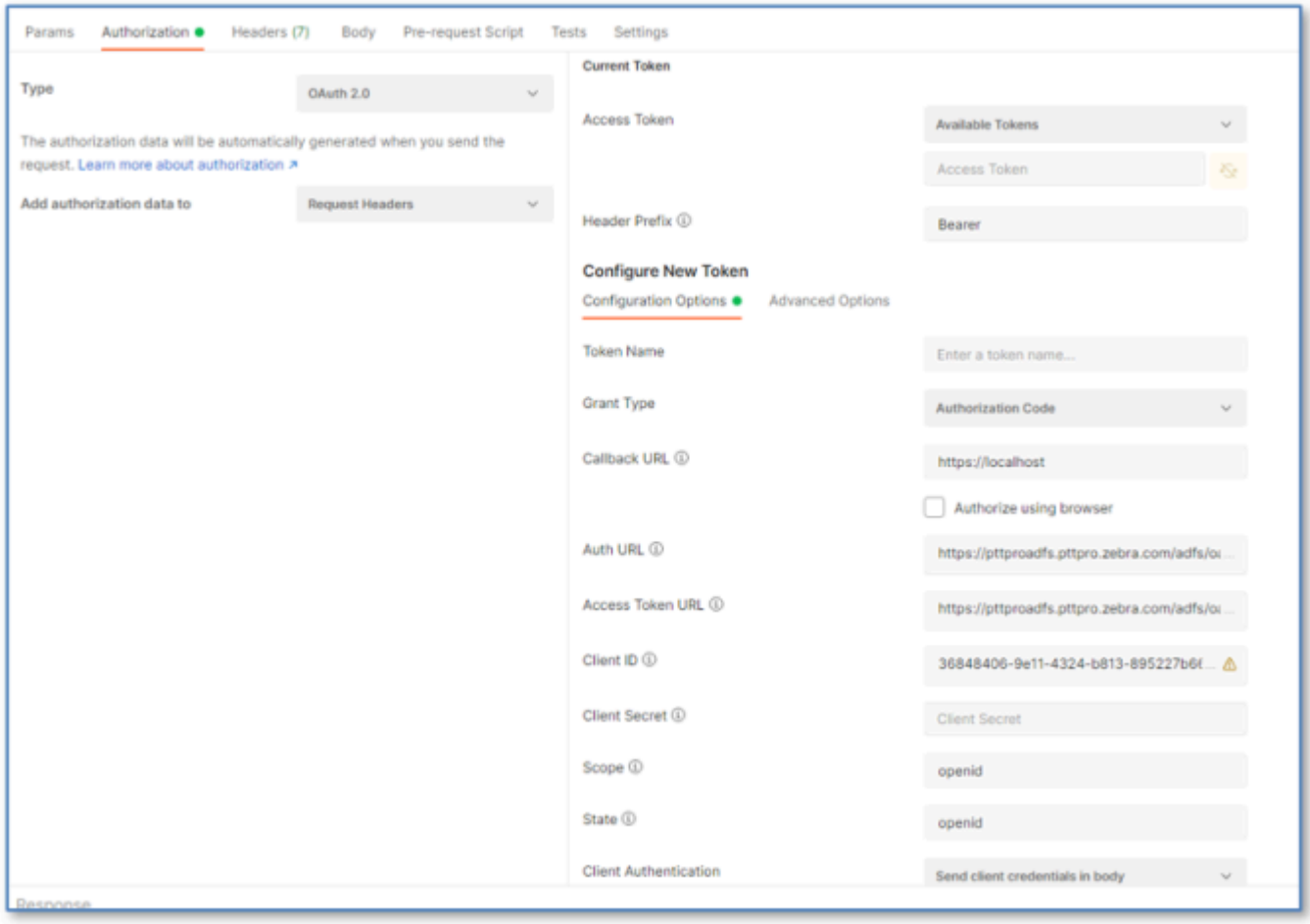
Using Postman

Postman is a graphical tool used to test APIs and construct HTTP requests. You can also use Postman to validate the Authorization and Token URLs, the Client ID, and the Client Secret you will need to access the ADFS server from PTT Pro and Profile Manager.

NOTE: Postman can be freely downloaded from the web.

You can use a tool like Postman to validate expected connection attributes and retrieve an Access Token from the connection.

Figure 3 Requesting an Access Token



Use the information collected from the customer environment and enter the following information into Postman:

1. Enter the Authorization URL in the **Auth URL** field.

   The Authorization URL is found in the response from the Well-Known URL and then adding the identifier from the Web API Application Group with `?resource=`.

2. Enter the client identifier in the **Client ID** field.

   The client identifier is found in the Native Application Group.

3. Enter openid in the **Scope** field.

   The openid is specified in Web API Application Group when creating the application permissions.

4. Enter openid in the **State** field.

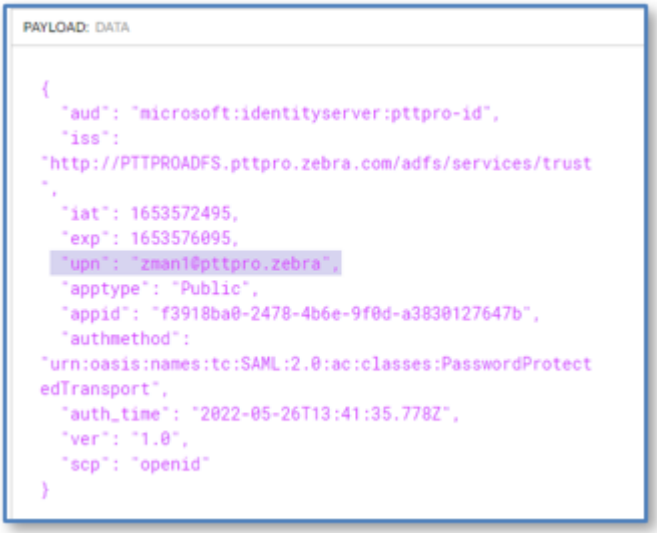
After you enter the required information, clear any cookies that may be present and initiate the Get New Access Token. If the settings are accurate, the console will prompt you for AD credentials. Enter valid user credentials to retrieve the Access Token.

Examine the Returned Access Token

After you sign on using Postman, the next step is to examine the returned JSON Web Token Access Token (JWT) using the JWT Token Analyzer.

Copy the Access Token returned from Postman and paste the token into the analyzer at https://jwt.io.

Review the Payload portion of the token. For example:



- Valid encryption is expected, for example, RS256.
- The **aud** field contains the correct identifier.
- The identifier in the token must match the Relying Party Identifier stated in the Standalone Web API in the Application Group.
- The Claims Issuance examines the token contents, so the Claims can use the UPN. The Name is not included in the Token so is not used or needed in this definition. If other elements are available, the Claims can be refined with the token elements.

This process is also helpful in the Profile Manager environment to identify token elements used to determine User Profile assignment.

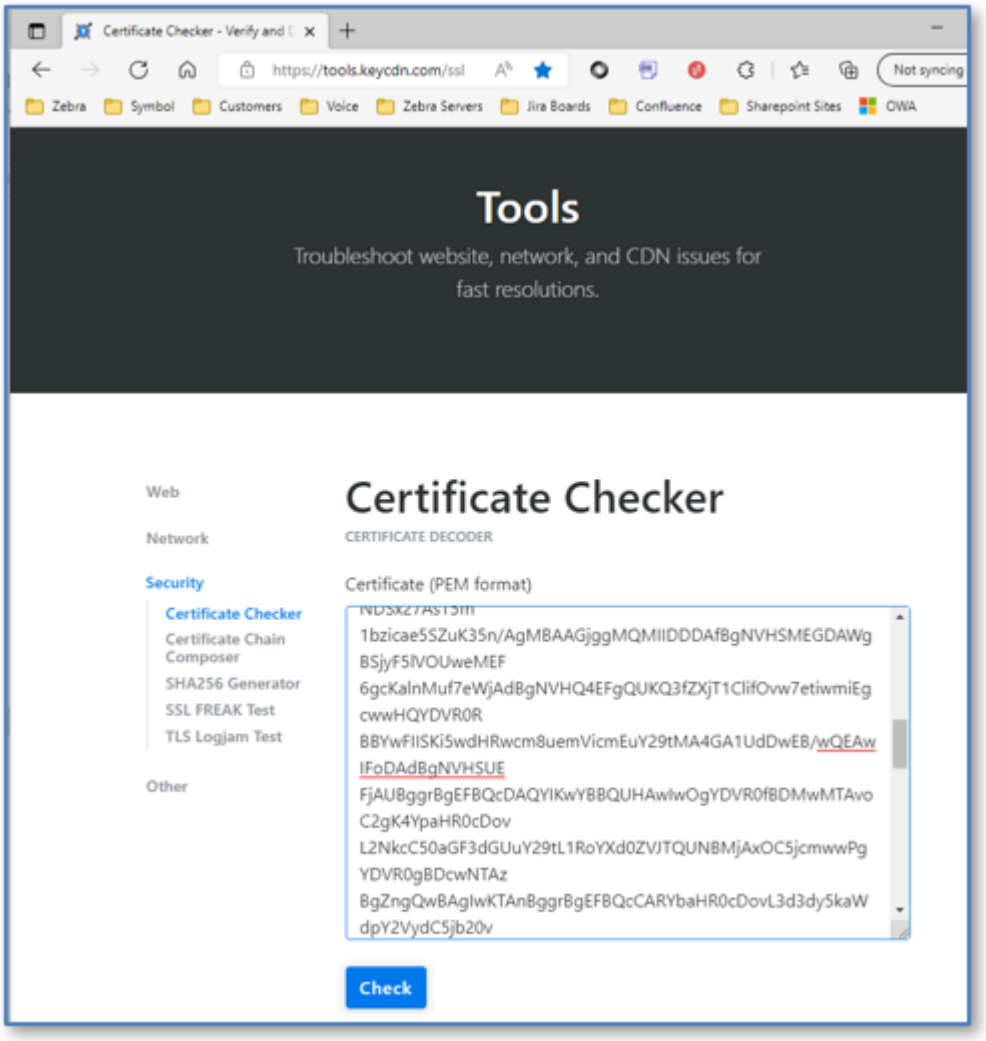
Validate the Signing Certificate

A valid signing certificate is required for the PTT Pro server OAuth2 configuration. In a previous step, you exported the Signing Certificate for later use. This process reveals if the Signing Certificate is valid and can be used in the PTT Pro server runtime configuration.

Open the Signing Certificate in Notepad and copy the certificate into the clipboard.

Verify the integrity of the certificate using a validation tool such as Certificate Checker. Pasting the token into the tool enables you to verify that the token details are as expected.

Figure 4 Validating the Signing Certificate



The Certificate Checker produces JSON output, which is not shown here for security purposes. The Certificate Checker provides information including:

- Validity of the certificate
- Expiration timing
- The Signature RSA encoding

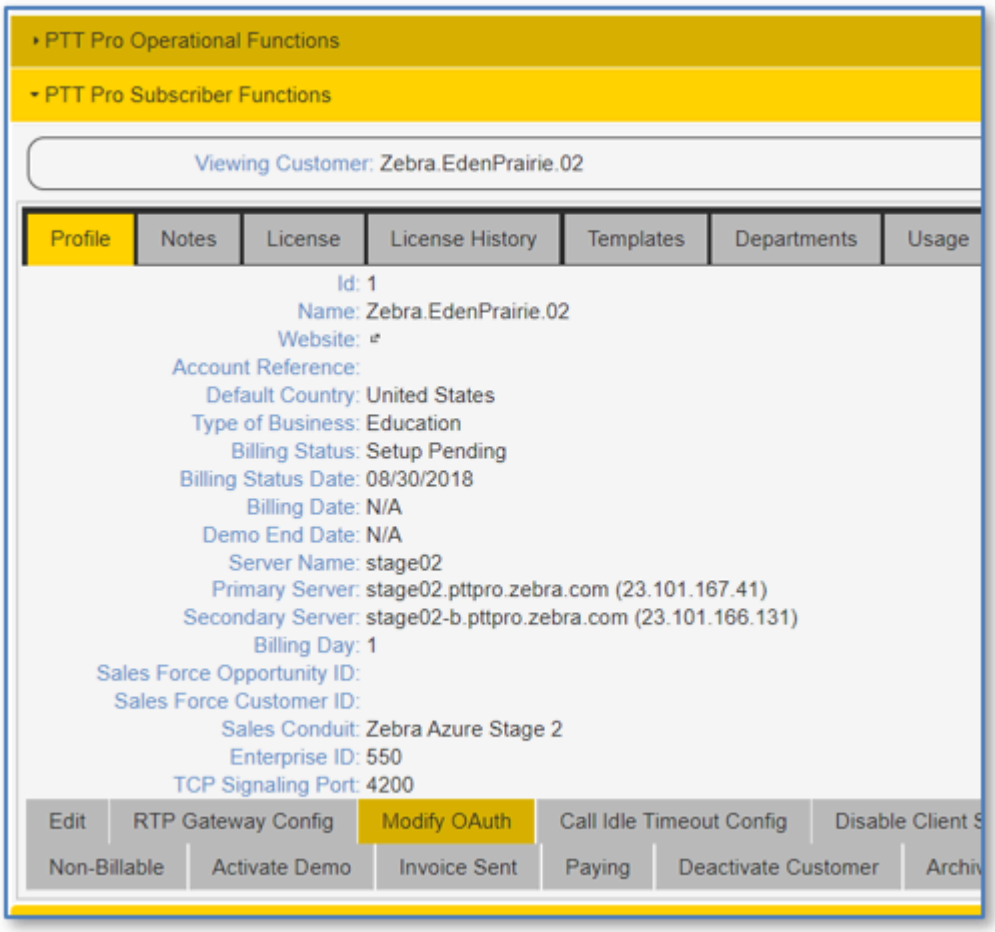
After you perform the validation steps, you can begin to configure the PTT Pro server and Profile Manager.

Configuring PTT-Pro to Support OAuth2

Configure the PTT Pro server to use OAuth2 to authorize requests and grants through tokens between the customer's identification provider and PTT Pro.

The OAuth2 Access URLs and the ADFS Signing Certificate token must be entered in the PTT Pro Management Portal.

1. Open the PTT Pro Management Portal through a Web browser and navigate to the Customer Configuration.



2. Click **Modify OAuth** or **Enable OAuth**.

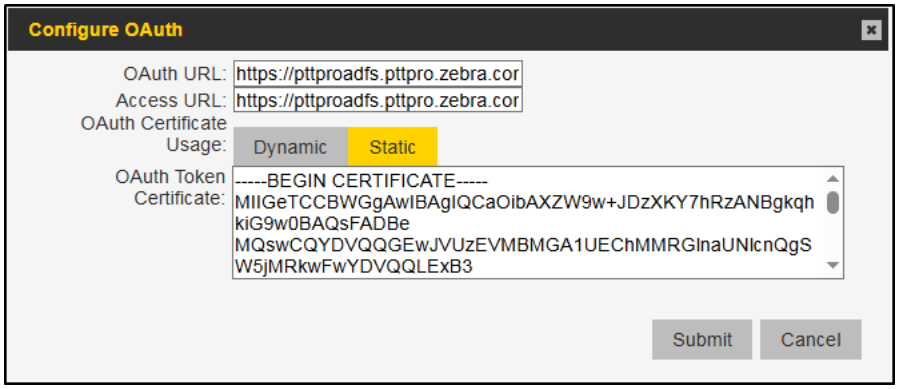
   The **Configure OAuth** dialog box appears.

3. Enter the OAuth URL and the Access URL.

   - OAuth URL example: `https://<server.domain.com>/adfs/oauth2/authorize?resource=pttpro-id`
   - Access URL example: `https://<server.domain.com>/adfs/oauth2/token`

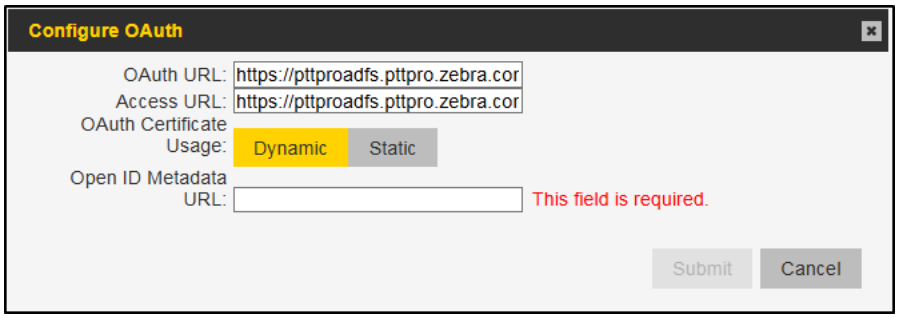
4. Select **Dynamic** or **Static** for the **OAuth Certificate Usage**.

   - If you select **Static**, copy the ADFS Signing Certificate token you created previously and paste it into the **OAuth Token Certificate** field. If the certificate changes, you must update the token to maintain the Replying Party Trust with the PTT Pro Server.



   - If you select **Dynamic**, enter the Open ID Metadata URL into the **Open ID Metadata URL** field. The URL automatically handles certificate rotation.

     For example, `<Server URL>/.well-known/openid-configuration` .



5. Click **Submit**.

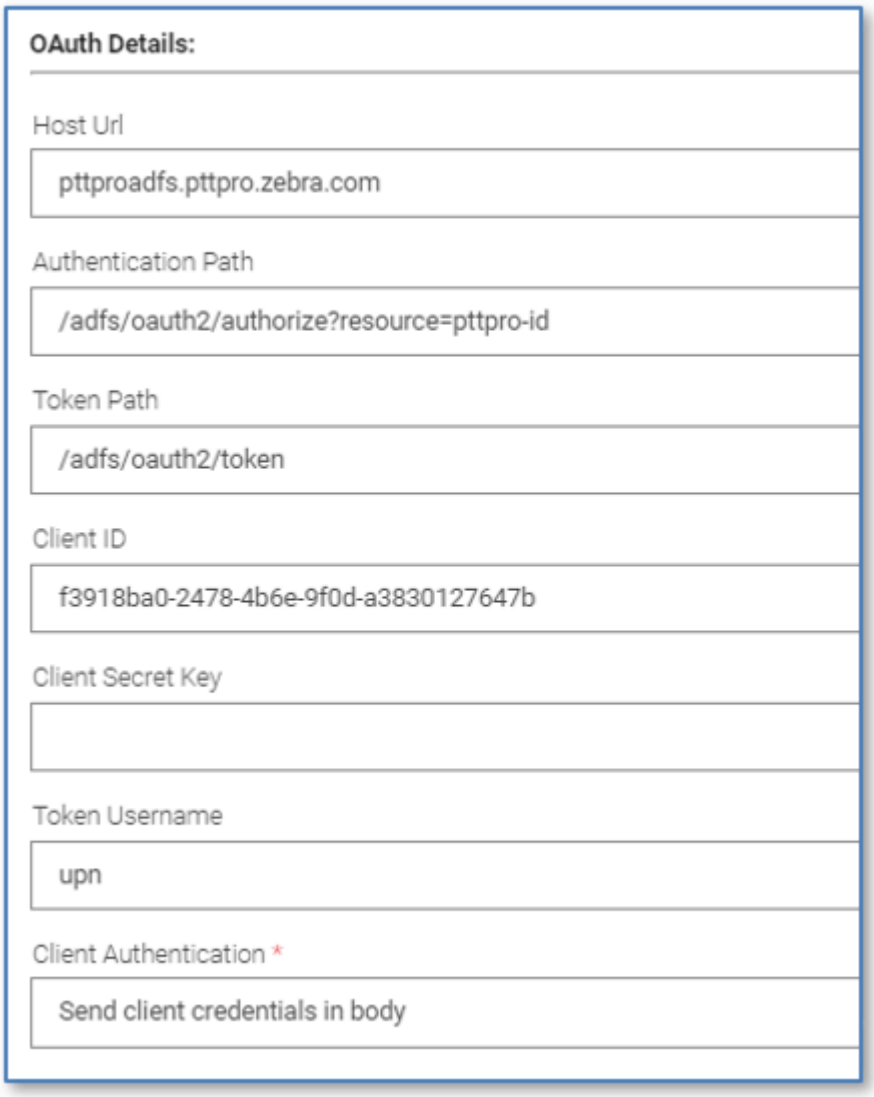
The Relying Party Trust is established in the PTT Pro Server.

Configuring Profile Manager to Support OAuth2

The steps for configuring Profile Manager to support OAuth2 differ from the steps to configure PTT Pro.

Profile Manager configuration requires five elements, each element can be derived from the Well- Known URL, the ADFS Configuration, or the Token Certificate.

- Enter each of the elements in the **OAuth Details** screen, as shown in the example below.



Host URL
The is the customer domain. The Host URL is the prefix for the Authentication and Token Path. https is assumed and automatically added by the system.

Authentication Path
The authorization path is found by browsing to the Well-Known URL. The Resource query of the Relying Trust Identifier must be added to the URL, as in `?resource=pttpro-id`.

Token Path
The token path is appended to the Host URL as it is captured in the JSON response from the Well-Known URL.

Client ID
The Client ID is provided by the customer and copied at the time of creating the Relying Trust.

Client Secret Key
The secret is provided by the Customer and it is created when the Native Application Group is established.

Token Username
The Token Name can be determined by using the JWT.IO web site to examine the access token retrieved by Postman.

External References

There are a number of ways to configure ADFS but the configuration should allow Workcloud Communication servers to communicate with the ADFS deployment.

The ADFS administrator may find the following links helpful.

- Single log-out for OpenID Connect with AD FS | Microsoft Docs
- How to enable cross-app SSO on Android using MSAL - Microsoft identity platform | Microsoft Docs
- Mobile application authentication documentation | Microsoft Docs
- The mystery of the missing ADFS OAuth JWT claims | by Rory Braybrook | The new control plane | Medium
- How to configure intranet forms-based authentication for devices that do not support Windows Integrated Authentication (WIA) | Microsoft Docs
- AD FS 2016 Single Sign On Settings | Microsoft Docs
- AD FS OpenID Connect/OAuth Concepts | Microsoft Docs

Revision History

Changes to the guide are listed below:

Change	Date	Description
MN-004591-01	08/2022	First version.
MN-004591-02	02/2024	Added topic for changing the access and refresh token lifespans.
MN-004591-03	05/2024	Updated OAuth configuration for the PTT Pro Server.