censornet.

Date: 28/04/2022
Version: 1.0

Censornet Archive – OAuth with ADFS Configuration Guide

Contents

Introduction

This document shows how to configure OAuth for user's authentication against on premise Microsoft Active Directory via Active Directory Federated Service (ADFS).

ADFS side Settings - Step by Step

A quick run through of the steps involved in supporting the Active Directory Federation Services (ADFS) for authentication using OAUTH2 in Windows Server 2012 R2 and above.

Step 1: Register an ADFS Client for OAuth

Register the OAuth client with ADFS by using calling PowerShell cmdlet. A template for the command is shown here – showing key elements that you will need to supply:

```
Add-AdfsClient -Name <ClientApp> -ClientId <ClientId> -RedirectUri
https://archive.clouduss.com/uss/microsoftoauth.do -Description "ADFS OAuth"
```

Where:

-ClientId: Client identifier for the OAuth client to register with ADFS. It can be set as any *random string that is likely to be unique for this client. Note that the same string needs to be specified under OAuth connection created at Censornet Archive end. Replace the **<ClientId>** with a reasonably long text string (e.g. "71d3cf488b4bf413547e83410e047885f3148ec5bcd90913") that cannot be easily guessed.

***Note – You can use an online UUID tool for the creation of the ClientId**

-RedirectUri: Specifies your Censornet Archive redirection URIs to register with ADFS against this OAuth client. Except URIs registered here, no other URI can be used as target endpoint for successful authentication by ADFS.
*Replace the <https://archive.clouduss.com/uss/microsoftoauth.do> with your Censornet Archive region hostname.

*Check Censornet support site for the URL's for Censornet's another Archive regions.

-Description: Any suitable description for the OAuth client

-Name: Any appropriate name for the OAuth client

Replace the <**ClientApp**> with a string value like "Client App", this is likely to be used as the target name for the app.

Example 1: Add a client

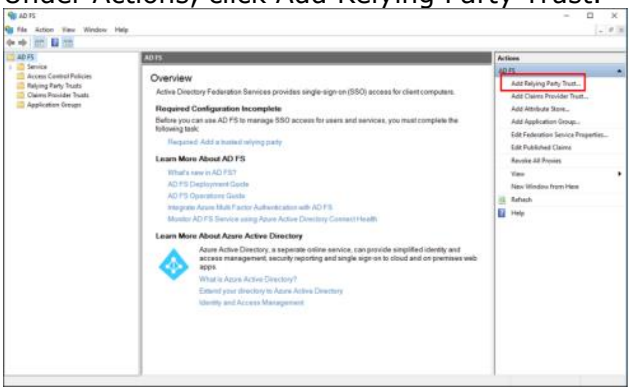
```
Add-AdfsClient -Name "CensornetADFS" -ClientId
"71d3cf488b4bf413547e83410e047885f3148ec5bcd90913" -RedirectUri
https://archive.clouduss.com/uss/microsoftoauth.do -Description "ADFS OAuth"
```

Step 2: Create a Relying Party Trust

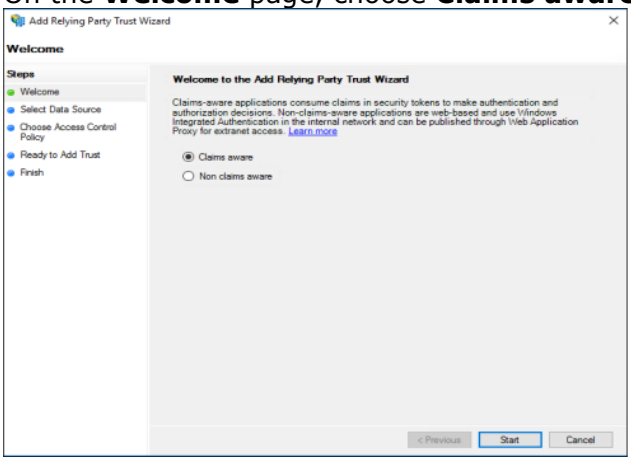
A new relying party trust needs to be added by using the ADFS Management snap-in. This requires following procedure.

Membership in Administrators or equivalent, on the local computer is the minimum required to complete this procedure.

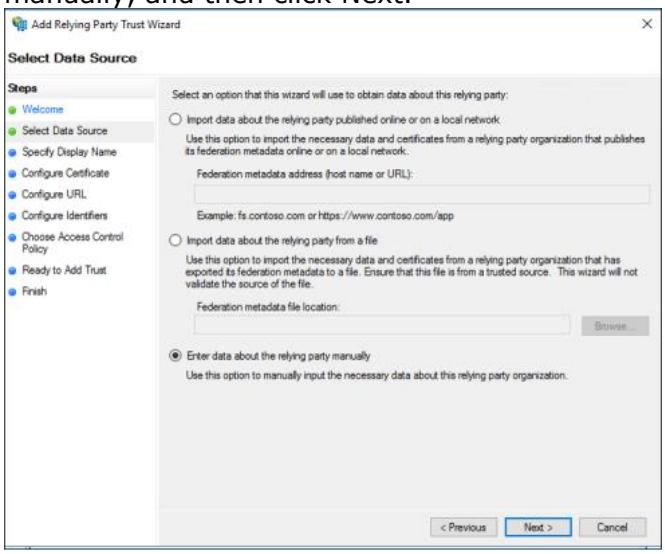
1. In Server Manager, click Tools, and then select **AD FS Management**.
2. Under Actions, click Add Relying Party Trust.



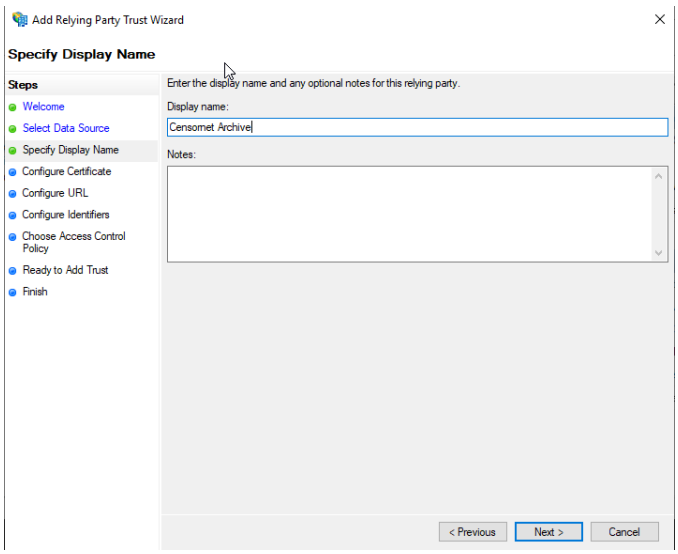
3. On the **Welcome** page, choose **Claims aware** and click **Start**.



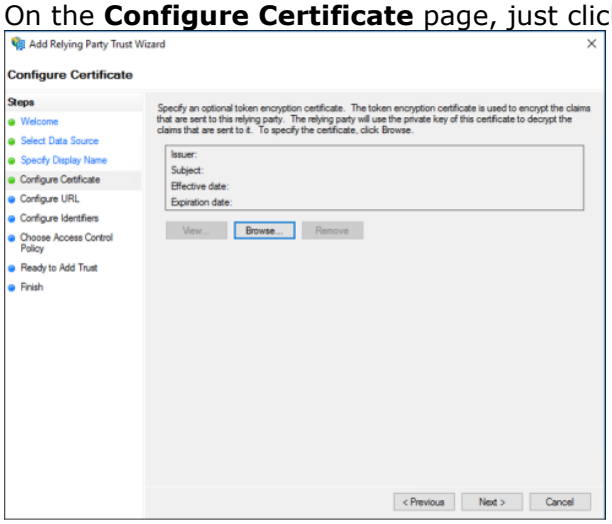
4. On the Select Data Source page, click Enter data about the relying party manually, and then click Next.



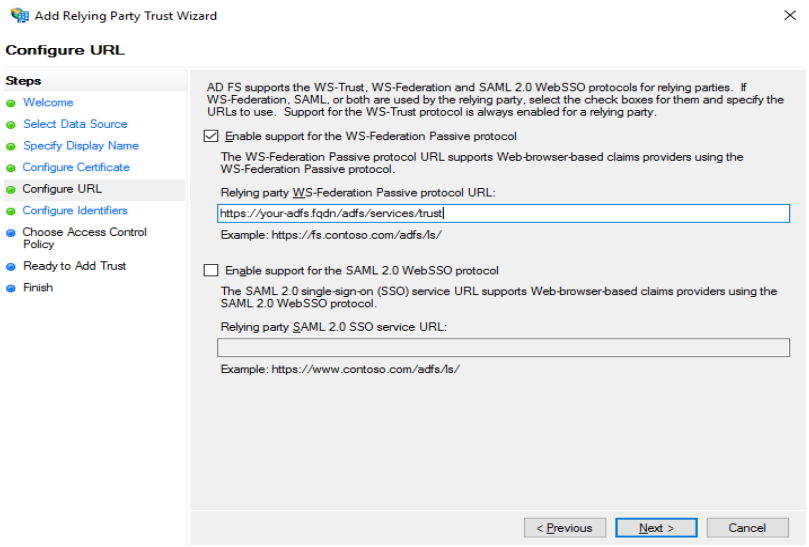
5. On the **Specify Display Name** page, type any appropriate name in **Display name**, under **Notes** type a description for this relying party trust, and then click **Next**. This name is to be used as the Target Name for this Relying Party.



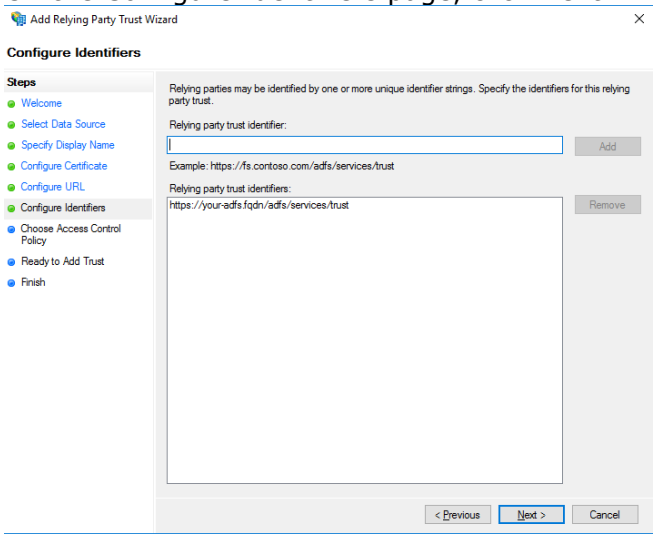
6. On the **Configure Certificate** page, just click on the **Next** button.



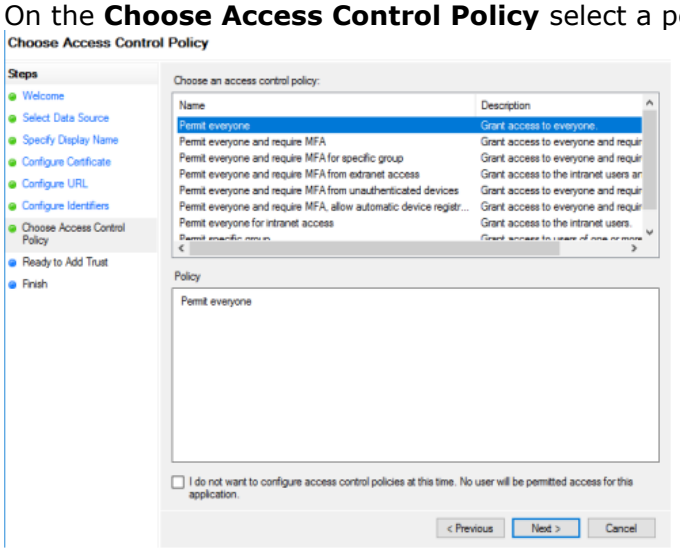
7. On the **Configure URL** page, select the **Enable support for the WS-Federation Passive protocol** check box. Under **Relying party WS-Federation Passive protocol URL**, type the URL for this relying party trust, and then click **Next**.
   This URL can be like https://<your-adfs.fqdn>/adfs/services/trust this must be unique amongst the Relying Parties. Please note that, for the purposes of OAuth, this URL does not need to be a real web address – it is just used as a unique identifier name. It will be used later in Censornet's Archive OAuth configuration and in the ***Grant-AdfsApplicationPermission PowerShell command.***



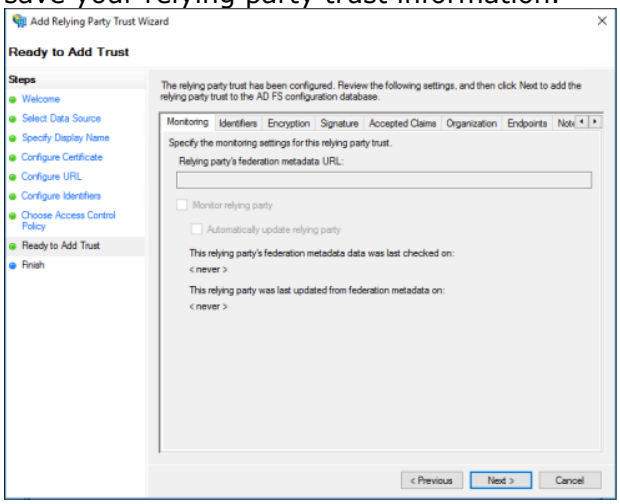
8. On the Configure Identifiers page, click Next.



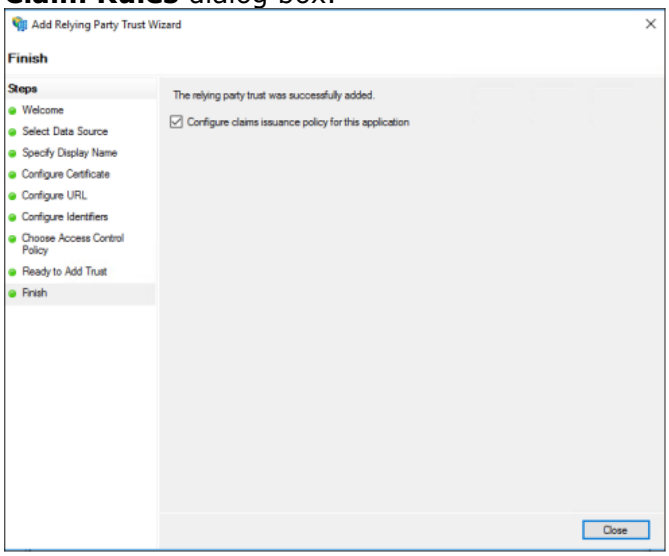
9. On the **Choose Access Control Policy** select a policy and click **Next**.



10. On the **Ready to Add Trust** page, review the settings, and then click **Next** to save your relying party trust information.



11. On the **Finish** page, click **Close**. This action automatically displays the **Edit Claim Rules** dialog box.

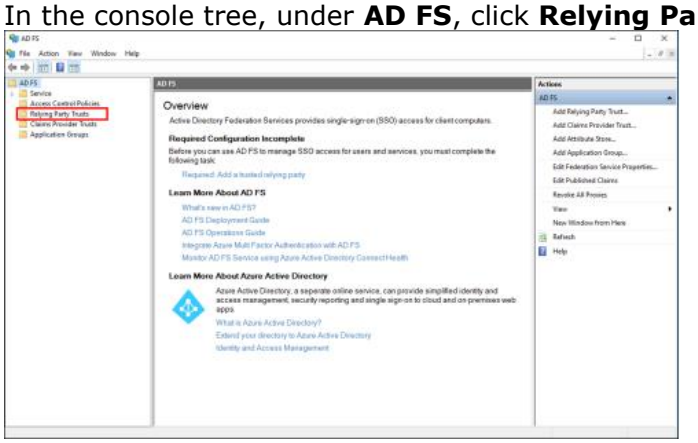


Step 3: Create a Rule to Send LDAP Attributes as Claims for a Relying Party Trust

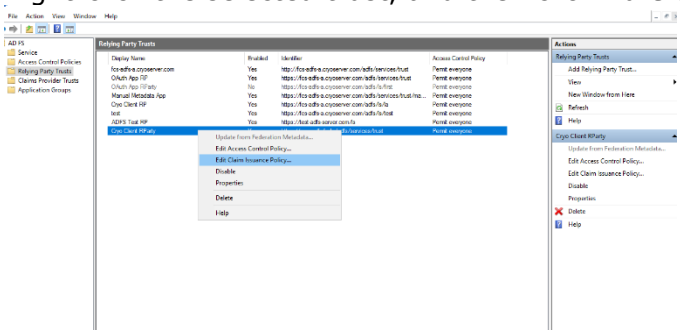
Using the "Send LDAP Attributes as Claims rule" template in AD FS, we can create a rule that will select required attributes from a LDAP attribute store, to send as claims to the relying party.

Membership in **Administrators**, or equivalent, on the local computer is the minimum required to complete this procedure.

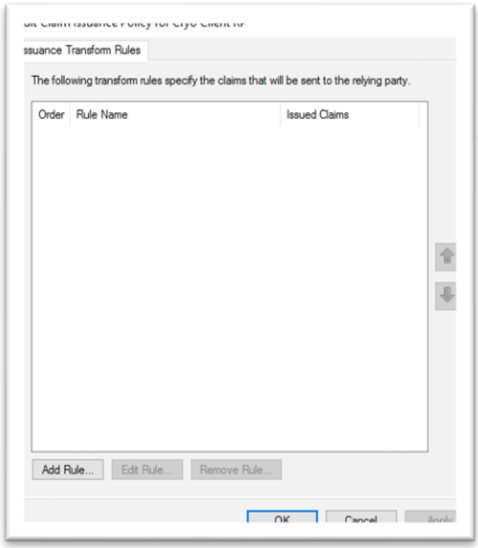
1. In Server Manager, click **Tools**, and then select **AD FS Management**.
2. In the console tree, under **AD FS**, click **Relying Party Trusts**.



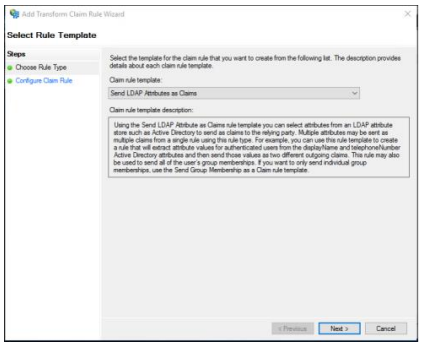
3. Right-click the selected trust, and then click **Edit Claim Issuance Policy**.



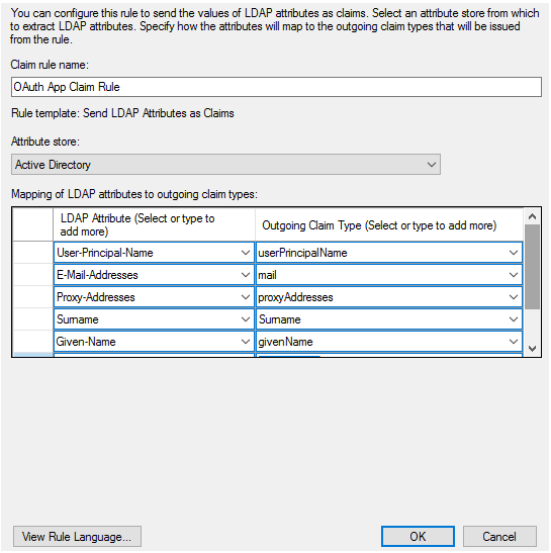
4. In the **Edit Claim Issuance Policy** dialog box, under **Issuance Transform Rules** click **Add** Rule to start the rule wizard.



5. On the Select Rule Template page, under Claim rule template, select Send LDAP Attributes as Claims from the list, and then click Next.



6. On the **Configure Rule** page under **Claim rule name** type the display name for this rule, select the **Attribute Store** (Active Directory) and then select each LDAP attribute required by Censornet Archive, and map it to the outgoing claim type. Add the Claims with the same Outgoing Claim Type name corresponding to the LDAP Attributes as shown:



LDAP Attribute	*Outgoing Claim Type*
User-Principal-Name	*userPrincipalName*
E-Mail-Addresses	*mail*
Proxy-Addresses	*proxyAddresses*
Surname	*Surname*
Given-Name	*GivenName*

7. Click the **Finish** button.
8. In the **Edit Claim Rules** dialog box, click **OK** to save the rule.

Step 4: Set the option “EnableJWT” to true

Set the option “EnableJWT” to true on the Relying Party Trust you configured.

Execute the PowerShell Set-AdfsRelyingPartyTrust command. The template of this command is like this:

```
Set-AdfsRelyingPartyTrust -TargetName "<Relying Party Display Name>" -EnableJWT
$true
```

Replace the ***<Relying Party Trust Name>*** with the display name of the Relying Party, that you specified in the **Step 2, point 5.**

Example:

```
Set-AdfsRelyingPartyTrust -TargetName “Censornet Archive" -EnableJWT $true
```

Step 5: Grant permission to registered client app to access relying party

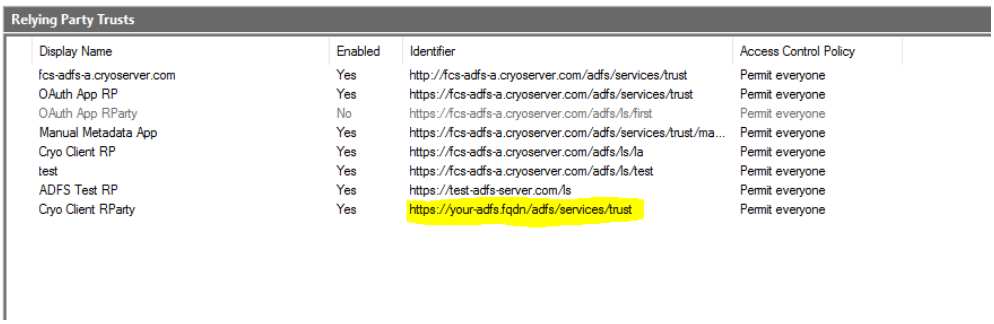
Run this command to explicitly grant clients permission to a resource

```
Grant-AdfsApplicationPermission -ClientRoleIdentifier "<ClientId>" -
ServerRoleIdentifier "<Relying Party Identifier>"
```

Where,

-ClientRoleIdentifier Client ID, that is set while adding the client.
Refer to the **Step 1.**

-**ServerRoleIdentifier RP Identifier**. It should be the trust URL associated with Relying Party Trust as shown below



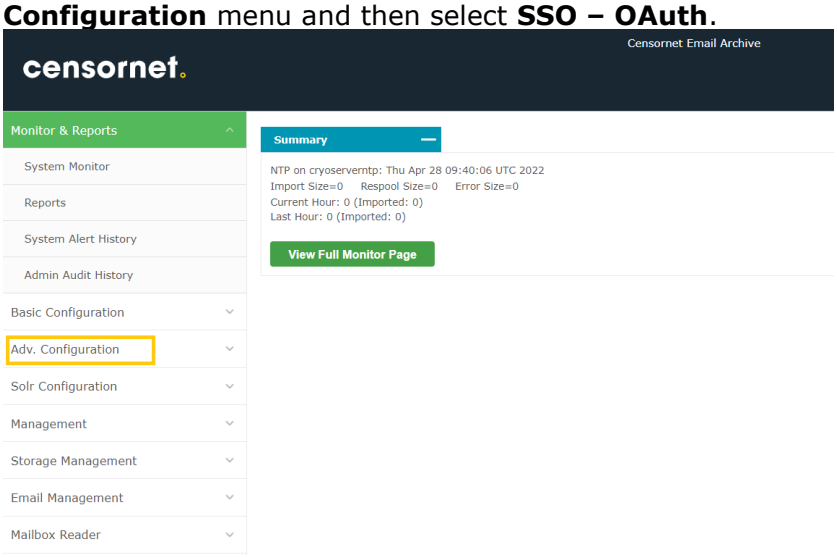
Example:

```
Grant-AdfsApplicationPermission -ClientRoleIdentifier
"71d3cf488b4bf413547e83410e047885f3148ec5bcd90913" -ServerRoleIdentifier
https://<your-adfs.fqdn>/adfs/services/trust
```

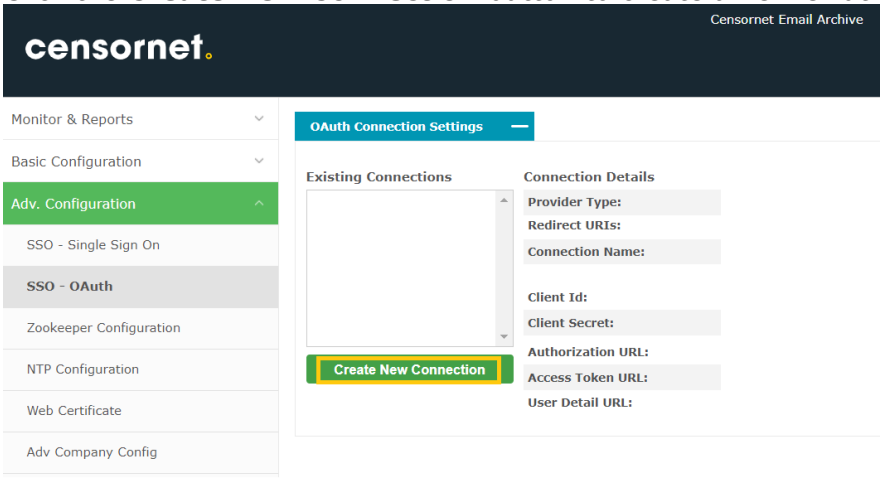
Censornet Side Settings

Create the OAuth connection for Microsoft ADFS provider type

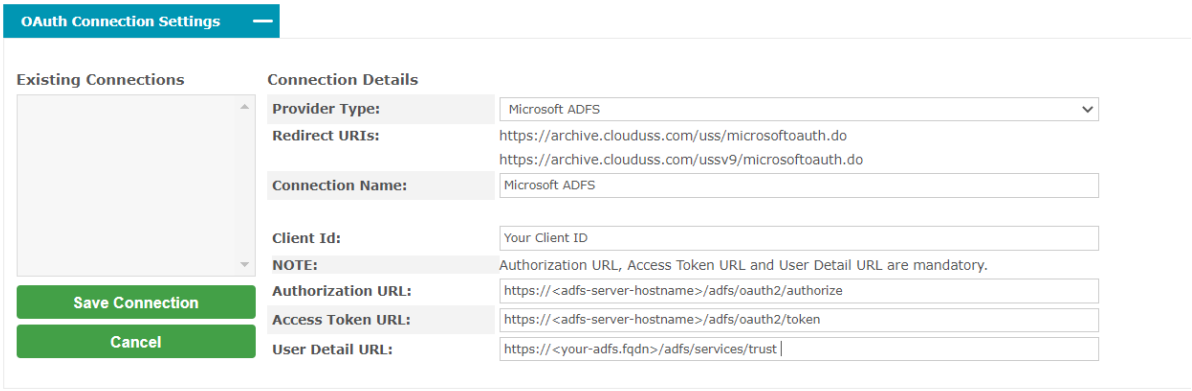
1. Login as admin user to your Censornet Archive enviroment, click on **Adv. Configuration** menu and then select **SSO – OAuth**.



2. Click the **Create New Connection** button to create a new OAuth Connection.

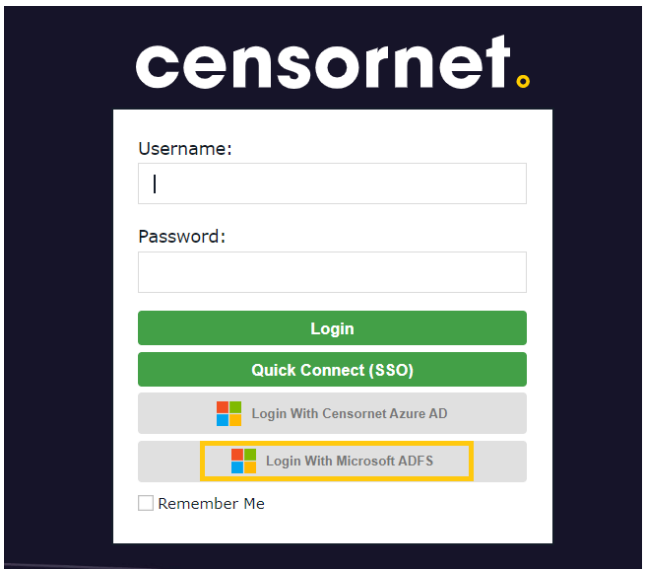


3. Under the Create New Connection, select **Microsoft ADFS** from the provider type drop down and enter the following details



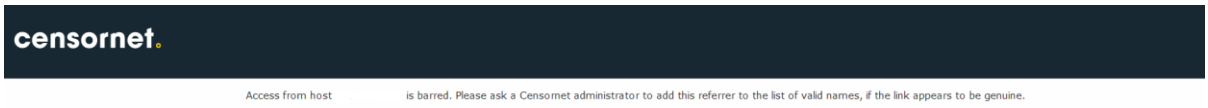
where,
- **Connection Name:** A relevant connection name that will also get displayed with label of OAuth login button on the login page.
- **Client Id:** Identity of the registered OAuth Client App. This should be the same "clientId" string as specified under step 1 of ADFS side settings.
- **Authorization URL:** URL where frontend will redirect the user for authorization, It is of the form "https://<adfs-server-hostname>/adfs/oauth2/authorize"
- **Access Token URL:** URL for getting access token against authorization code. It is of the form "https://<adfs-server-hostname>/adfs/oauth2/token"
- User Detail URL: Specify the Relying party trust identifier. It should be the same URL as first provided in **Step 2**: Create a Relying Party Trust, **part 7** (**Relying party WS-Federation Passive protocol URL**) and later as the ServerRoleIdentifier value specified under Step 5 of ADFS side settings.

4. Now save the connection and you are done!
5. The login page should now display a new button with the label "Login with <Connection Name>", to allow user login via **Microsoft ADFS** as shown



Access Control

For security, URL links pointing to Censornet's Archive will only be allowed for registered domains. The domains that are unexpected, the system will respond with an "Access barred" error message:



To fix this, you must contact **Censornet's Support Team** and ask them to add your OAuth service as a valid "referrer" to Censornet's Archive solutions.

Local User Accounts for OAuth

When a user accesses Censornet's Archive for the first time using OAuth, it will create a Local User Account entry within Censornet's Archive solution.

To review the accounts created by OAuth logins, visit the **Basic Configuration > Local User Accounts**. The users accounts will show the user's email address as their username. They will be set to "external authorization" (meaning that their password is not held in Censornet's Archive solution so must be validated with some external system).

Censornet Archiving OAuth Configuration

Modify registration settings for an OAuth client registered with ADFS

The **Set-AdfsClient** cmdlet modifies registration settings for an OAuth 2.0 client registered with Active Directory Federation Services (AD FS). Use this cmdlet to modify the settings, including the client identifier, redirection URI, name, or description of the OAuth 2.0 client. You can also use this cmdlet to register additional redirection URIs for the OAuth 2.0 client.

```
Set-AdfsClient -TargetName "<ClientApp>" -RedirectUri
@("https://<archive.clouduss.com>/uss/microsoftoauth.do",
“https://<archive.uk.clouduss.com>/uss/microsoftoauth.do”)
```

Example:

```
Set-AdfsClient -TargetName "Client App" -RedirectUri @("
https://archive.clouduss.com/uss/microsoftoauth.do",
“https://archive.uk.clouduss.com/uss/microsoftoauth.do”)
```

Check the registration settings for an OAuth client registered with ADFS

Use the following cmdlet template

```
Get-AdfsClient -name "<ClientApp>"
```

Example:

```
Get-AdfsClient -name "CensornetADFS"
```

Remove OAuth client registered with ADFS

The Revoke-AdfsApplicationPermission cmdlet revokes permission for an application in Active Directory Federation Services (AD FS).

Use the following cmdlet template

```
Revoke-AdfsApplicationPermission -TargetClientRoleIdentifier "clientID " -TargetServerRoleIdentifier
"https://<your-adfs.fqdn>/adfs/services/trust"
```

Example:

```
Revoke-AdfsApplicationPermission -TargetClientRoleIdentifier "2960ba77-37fc-4c91-a8dd-f6e5093b1ea2" -
TargetServerRoleIdentifier "https://ADFS.Censornet.com/adfs/services/trust/censornetarchive"
```

References:

https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/operations/create-a-relying-party-trust#to-create-a-claims-aware-relying-party-trust-using-federation-metadata