

# Using DataHub Active Directory Federation Services (ADFS)

# as an OAuth 2 Authorization Server

# Prerequisites

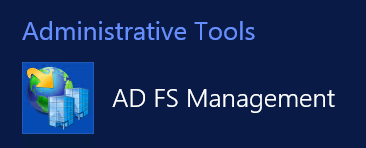
* An OAuth 2 capable client application - hereafter referred to as "Client App".
* An arbitrary (but agreed upon between ADFS and Client App) **resource** name for Client App.
  + "Resource" is used in the sense of an OAuth 2 Resource … the thing for which access permission is being given. This **resource** name will be transmitted in HTTP requests by Client App as required in OAuth 2 when redirecting users to an authorization server (ADFS), and will be **stored by ADFS** as the **Identifier** property of the **Relying Party Trust** that represents Client App.
  + A suggestion for **resource** name would be the Login URL for Client App (i.e., <https://ClientApp.com/login>).
* An arbitrary (but agreed upon between ADFS and Client App) **Client ID** value with which the client app will identify itself to ADFS in HTTP calls
  + This is a value that is required of all OAuth 2 clients, and so likely already exists (or is configurable) in Client App.
  + This Client ID should be a globally unique identifier (GUID), a name in the format *00001111-2222-3333-4444-555566667777*
  + The preceding numbers are arbitrary, but should only include the following hexadecimal characters: 0 1 2 3 4 5 6 7 8 9 0 a b c d e f
  + Either the ADFS administrator or the client app administrator can create the Client ID The main requirement is that it be unique.
* The URI at which an OAuth 2 capable client application (“Client App”) accepts an Authorization Code (<https://tools.ietf.org/html/rfc6749#section-1.3.1>).
* The URI at which an OAuth 2 capable client application (“Client App”) accepts a Token.

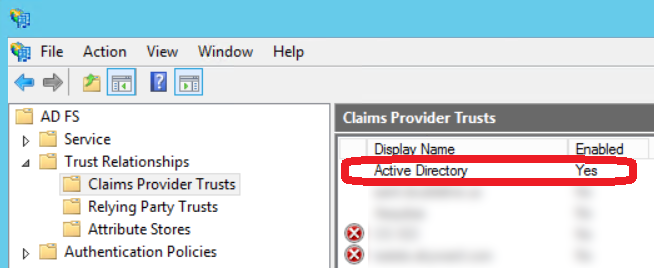
# Goal

The following procedure will configure single sign-on by **delegating authentication to Active Directory** when a user attempts to log in to Client App.

# ADFS steps (for DataHub SSO team)

1. Open the ADFS management console (found in the Admin Tools of the Active Directory Domain Controller or dedicated ADFS server). Click the **Claims Provider Trusts** node and ensure Active Directory appears as one of the entries in the Claims Provider Trust panel, and that **Yes** appears in the column **Enabled**.





1. Use Appendix "A" at the end of this document to configure Client App as a Relying Party Trust.
2. On the ADFS Server, run the following command:

Add-AdfsClient

-Name "Client App"

-ClientId "11111111-2222-3333-4444-555566667777"

-RedirectUri @("https://web.clientapp.com/sso/mi")

-Description "OAuth 2.0 client for Client App-Michigan Datahub integration"

-LogoutUri "https://web.clientapp.com/sso/mi"

|  |  |
| --- | --- |
| **Parmeter** | **Purpose** |
| Name | Friendly name to be displayed in the ADFS Management Console. |
| Client ID | The unique identifier agreed upon by DataHub and Client App that Client App will identify itself with when requesting an Authorization Code or Token from ADFS. This value may be exposed publiclly - it is a name, not a secrtet. This value is arbitrary … no meaning will be assumed other than it uniquely identifies Client App. |
| Redirect URI | One or more URL's to which Client App may request that ADFS send it an Authorization Code and/or Tokens. In OAuth 2, when a user wants to access Client App, it is expected the Client App will redirect the user's browser (with a HTTP 302 response) to ADFS's Autorization Endpoint accompanied by a ***Redirect URI*** … and after logging in at ADFS, the user will be redirected again to the **Redirect URI** provided by Client App, carrying the Authorization Code needed to this URI. ADFS will only redirect the logged in user back to Client App if the Redirect URI that ADFS receives match exactly (including any terminating slash characters) |
| Description | Long Text friendly description to be displayed in the ADFS Management Console. |
| LogoutUri | **OPTIONAL**: If Client App wants to be notified when a user logs out of ADFS, then Client App must provide a URI to which ADFS can send such notifications via HTTP. This communication could allow Client App to immediately invalidate any tokens related to this user, for example. |

# Example OAuth 2 Flow between Client App and ADFS (Data Hub)

The following is a step-by-step description of how the OAuth 2 Authorization Code Grant flow might happen:

* A user attempts to log in to Client App.
* Because Client App has indicated that it supports OAuth 2 Authorization Code Grant, it can be configured to redirect the user to the ADFS Authorization URL where the user will log in: https://[ADFSServerURL]**/adfs/oauth2/authorize?response\_type=code&client\_id=***ClientID***&resource=**https%3A%2F%2FClientApp.com%2Flogin**&redirect\_uri=**https%3A%2F%2FClientApp.com%2F*OurRedirectUri*

|  |  |
| --- | --- |
| **Parmeter** | **Purpose** |
| response\_type | Always **code**. Tells ADFS that Client App wants back an Authorization Code after the user successfully authenticates with ADFS. |
| client\_id | This is the **Client ID**, the public, unique identifier created to represent Client App to ADFS, as described earlier in this document. |
| resource | This is defined by Client App, but saved as the **Identifier** property of the Relying Party Trust created in ADFS to represent Client App.  Normally, in OAuth 2's standard flow, this would be the name of some data/documents/applications belonging to the user, but residing on a third party server, that the user will grant permissions to Client App to use.  But since we are creating an authentication (not an authorization) protocol, this should be an identifier that describes the Client App logon URL in the ADFS **Relying Party** like http://ClientApp.com/login. |
| redirect\_uri | An HTTP endpoint that Client App provides in order to receive authorization tokens from the OAuth 2 authorization server (ADFS, in this case). |

* ADFS compares the **redirect\_uri** parameter to the list of Redirect URI's provided in the PowerShell command given earlier in the document. If the redirect\_uri matches one of them, precisely, including any trailing slash that might be present, then the user is prompted to log in.
* The user logs in, then ADFS automatically redirects the user's browser to the redirect\_uri issued by Client App, carrying with it a newly created Authentication Code.
* Client App uses the Authentication Code and requests a token from ADFS (note that the client secret isn't required, nor used, by ADFS - it can be included, but will be ignored):

POST /adfs/oauth2/token HTTP/1.1

Content-Type: application/x-www-form-urlencoded

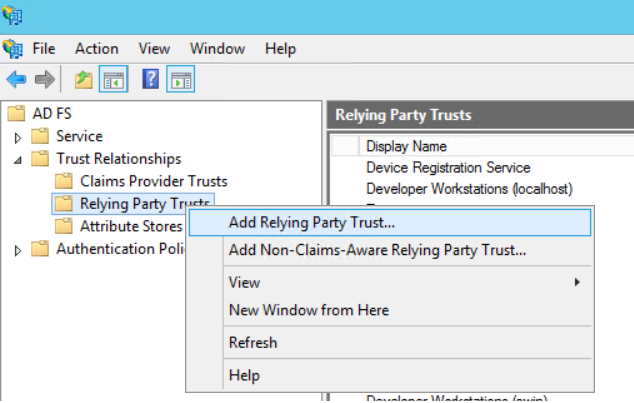
Host: adfs.michigandatahub.mi.us 🡨 ***This Host Name is provided by the data hub ADFS administrator to Client App***

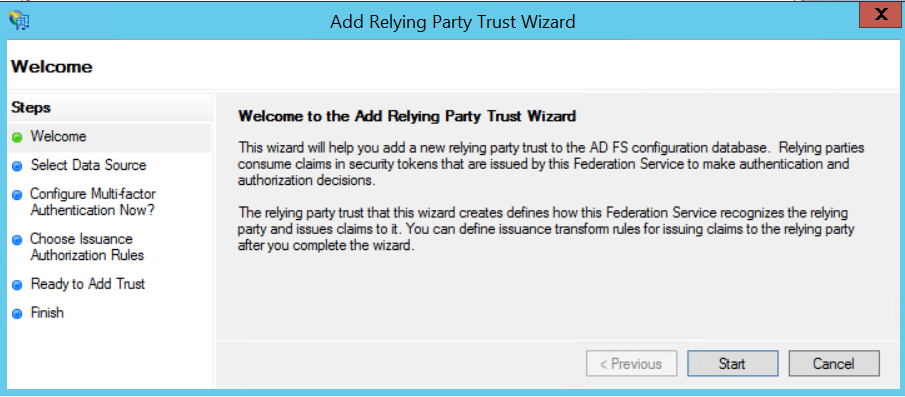
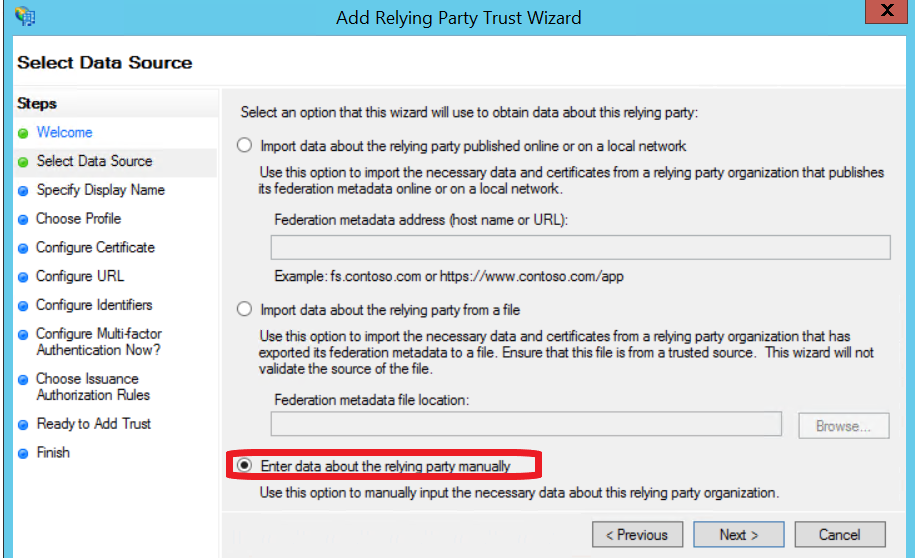
**grant\_type=authorization\_code&client\_id=***ClientID***&redirect\_uri**=http%3A%2F%2FClientApp.com%2F*giveMeToken***&code=***authorizationCode*

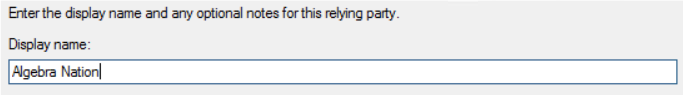
* As before, ADFS checks the redirect\_uri against those provided in the ADFS PowerShell setup command, and if there is a match, ADFS sends a token to Client App at the Redirect URI given in the request. The token is defaulted to 1 hour expiration, too. In this case, though, Client App needs the token merely for authentication, and can provide authorization merely by verifying one of the provided user claims inside the JWT token.
* Client App receives the a JWT token (JSON Web Token) with claims of the user's identity inside (name, email) from ADFS/Active Directory.
* Client App can make any authorizations it would as if the user had signed in directly, given that it supports the OAuth 2 Authorization Code Grant.

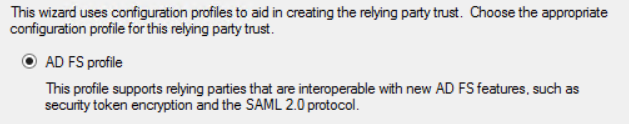
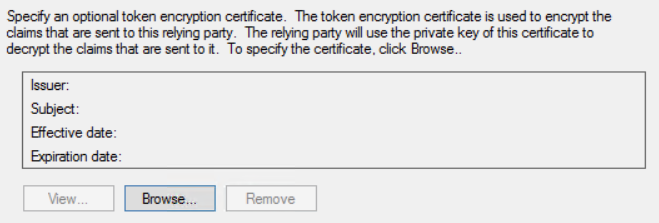
# Appendix A - Setting up a Relying Party Trust for Client App inside ADFS

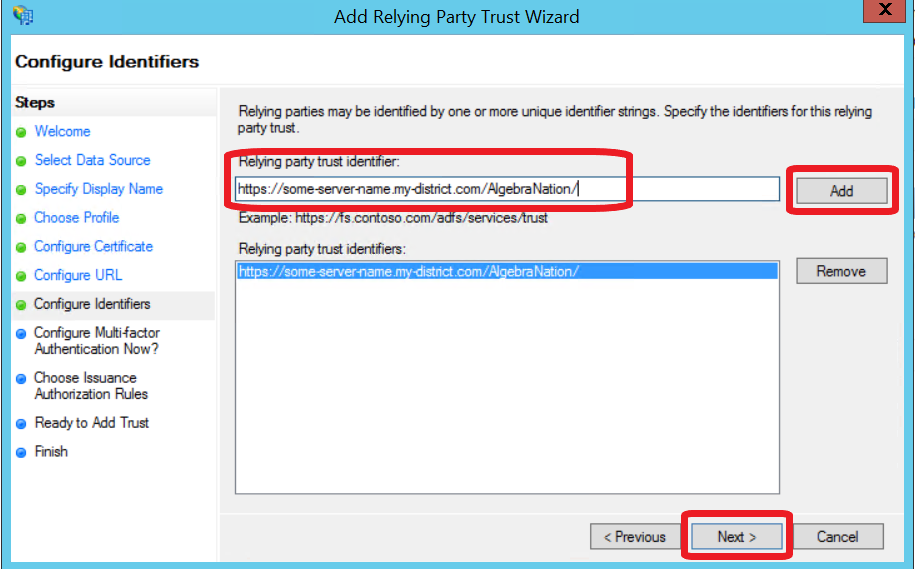
1. In the ADFS management console, right-click the **Relying Party Trusts** node and and click Add Relying Party Trust.

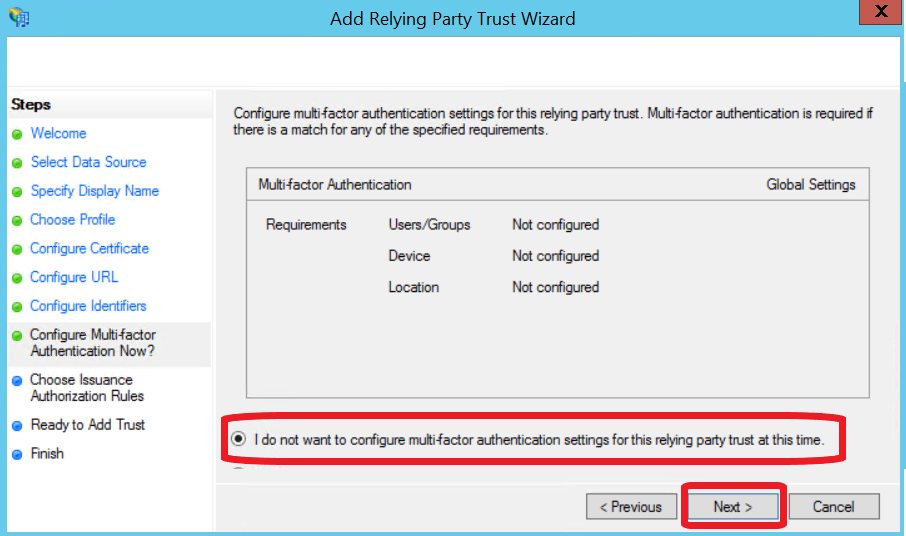
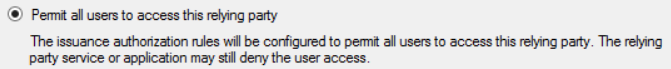
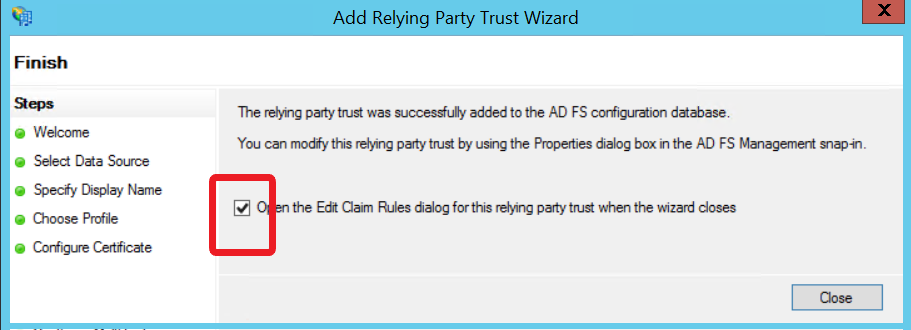
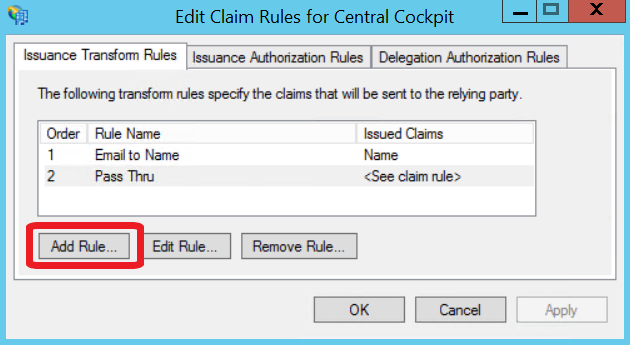
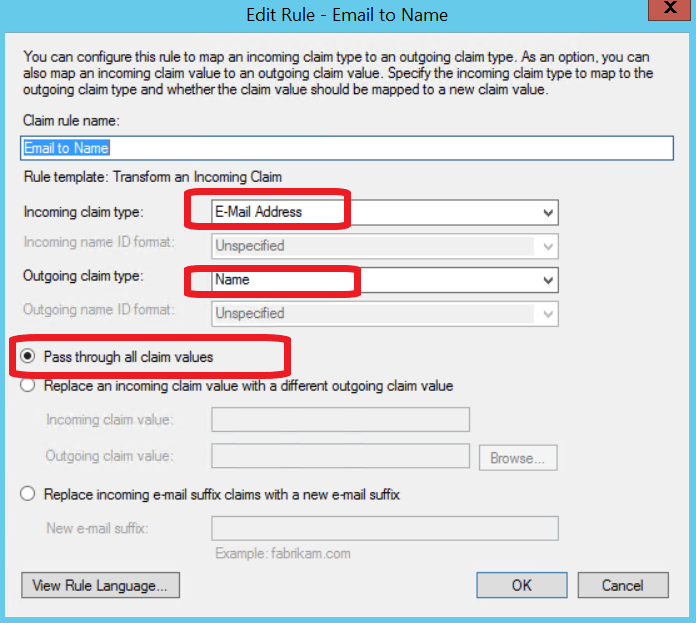
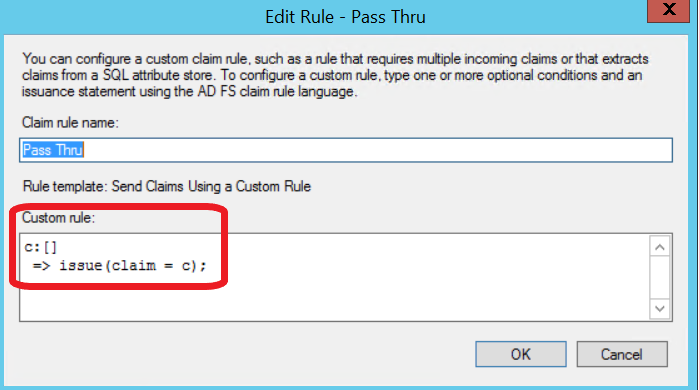


1. On the first step of the wizard, click **Start**.
2. On the second step of the wizard, select the option **Enter data about the relying party manually.** Click **Next**.
3. On the next step, enter the name of the application as you would wish an I.T. administrator to see it displayed.



1. During the next two steps, simply accept the default, or leave the input form blank.  
     
     
   
2. On the Configure Identifiers step, add the base URL for Client App.   
   In fact, this value is arbitrary, but must match the value Client App's web application provides for the **resource** in the HTTP request/redirect that seeks an Authorization Code from the user/ADFS. See the section **Example OAuth 2 Flow between Client App and ADFS (Data Hub)** for details.



1. Accept the default for the **Multi-Factor Authentication** step. This can be changed at a later time if desired.  
   
2. Accept the default for the **Choose Issuance** step. This can be changed at a later time if desired.   
   
3. On the step **Ready to Add Trust**, click **Next**.
4. On the step **Finish**, leave the checkbox **Edit Claims Rules** checked.  
   
5. Add the following two Claims Rules:  
     
     
   ***SEE SCREENSHOTS BELOW***  
     
   
6. NOTE: The ADFS Certifcate Thumbnail for ADFS may also be needed by Algrebra Nation. It can be found at: **ADFS** >> **Certificates** >> right-click **Token-Signing Certificate** >> **Properties …** >> tab **Details** >> **Thumbprint**.  
   