

BELNET

Federation Documentation

AD FS 3.0 IDP for windows 2012R2 server and the Belnet Federation

Table of Contents

1	Preparing the server.....	2
1.1	Domain membership.....	2
1.2	DNS.....	2
1.3	NTP.....	2
1.4	IIS (optional).....	2
1.5	Service account.....	2
2	Install AD FS 3.0.....	3
3	Before Configuring AD FS 3.0.....	6
3.1	Certificate.....	6
4	Configure AD FS 3.0.....	7
4.1	General configuration of the AD FS server.....	7
4.2	Upload your Metadata.....	13
4.3	Configure the Service provider.....	14
4.4	Configure the attribute release.....	18
5	Some “claim rules” for Belnet SP:	24
5.1	sptest.belnet.be:	24
5.2	filesender.belnet.be	24
5.3	mconf.belnet.be	24
6	Interesting docs:.....	25

1 Preparing the server

1.1 Domain membership

The server on which we will install shibboleth IDP needs to be a member of the domain. This will not be covered by this document.

1.2 DNS

DNS resolution should be properly configured for internal and external addresses. Also not covered in this document.

1.3 NTP

You have to make sure that your server is synchronized with NTP. We advise using the NTP server of Belnet (ntp.belnet.be)

1.4 IIS (optional)

We suppose that IIS Role has already been installed on the server. There are no more dependencies between AD FS and IIS since the 2012 R2 release, but we will use it to create a self-signed certificate. If you already have a certificate, you can use this one.

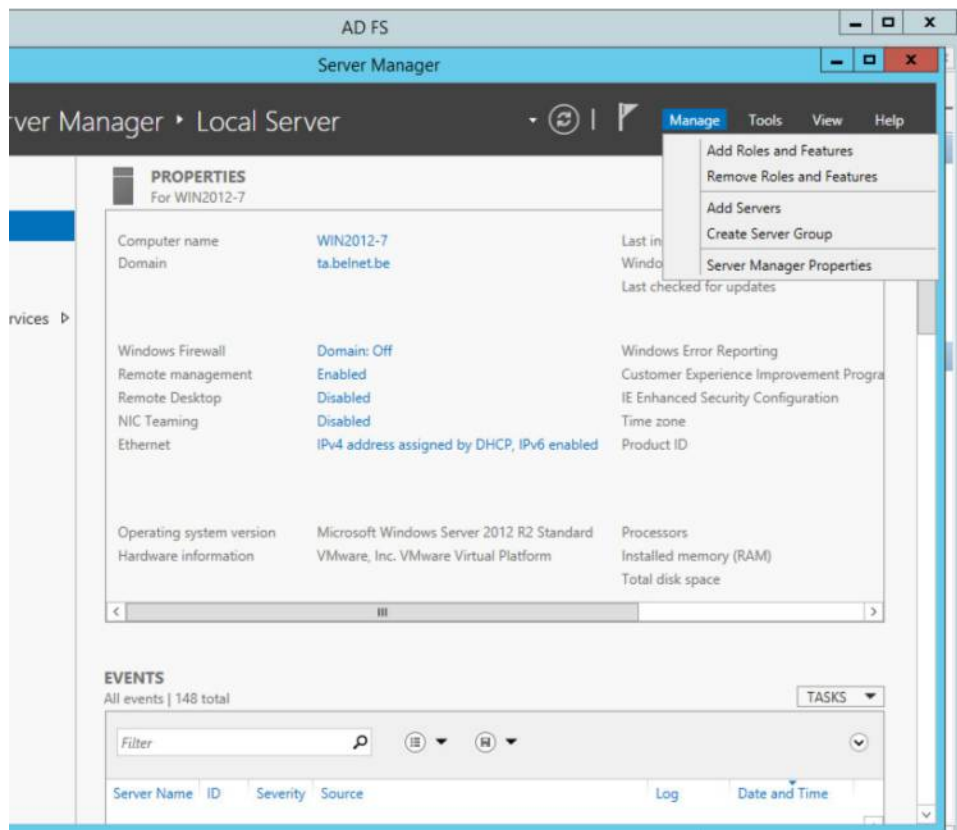
1.5 Service account

You will need a service account in order to launch the service.

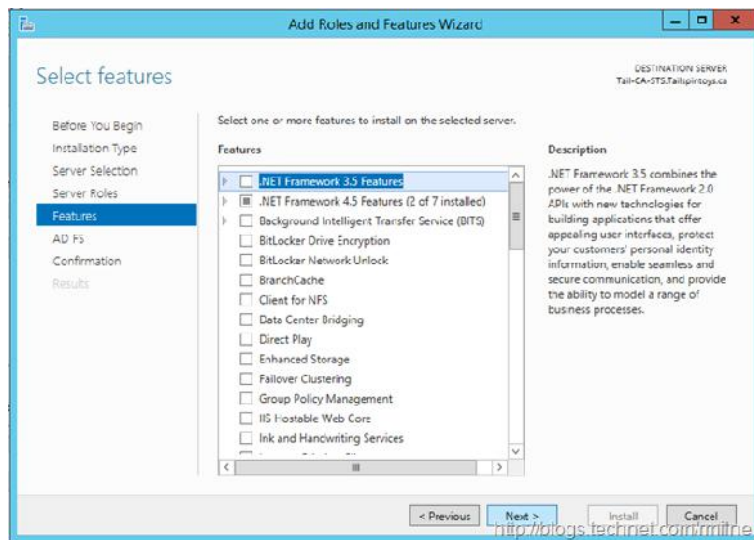
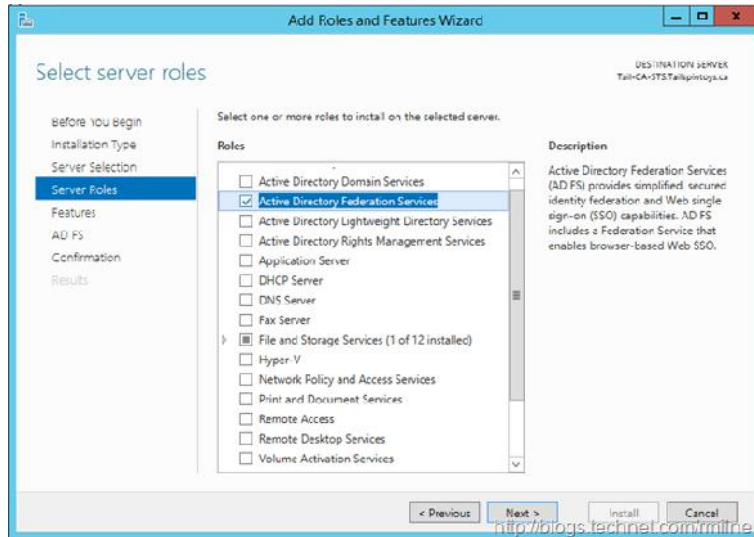
2 Install AD FS 3.0

In this document, we will only cover the creation of a federation server, in a new federation with a standalone server. If you already have a federation server active, you probably can skip this part and go directly to the configuration.

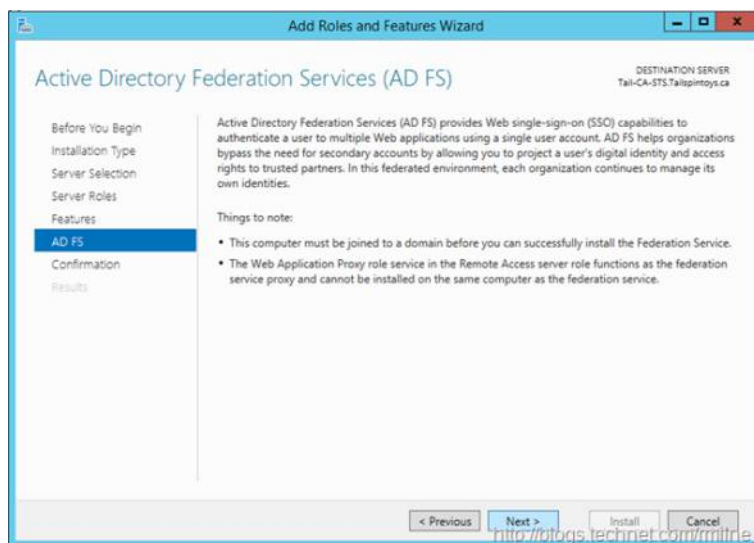
You will have to add the ADFS role to your server.

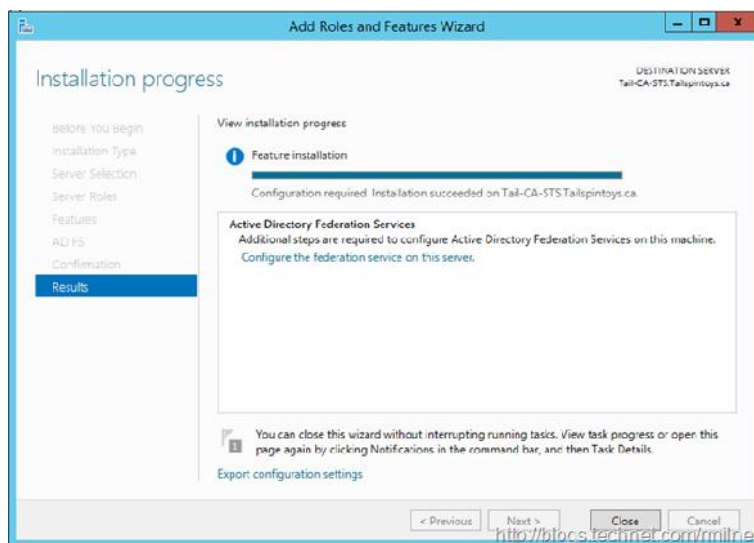
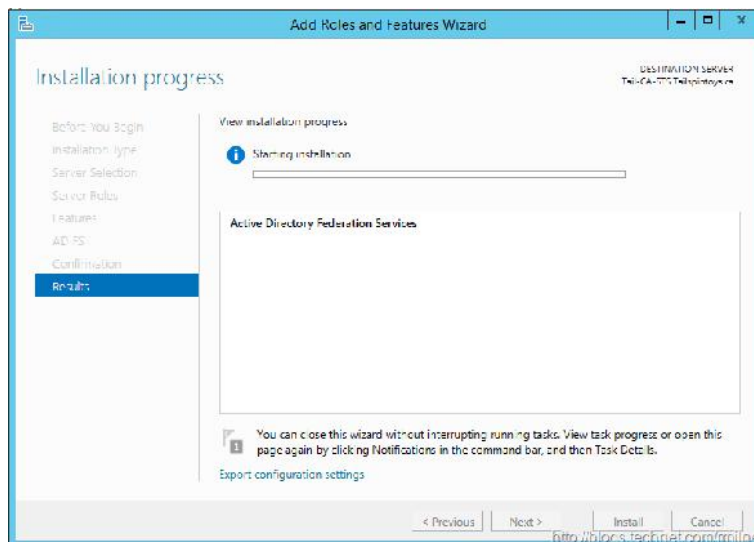
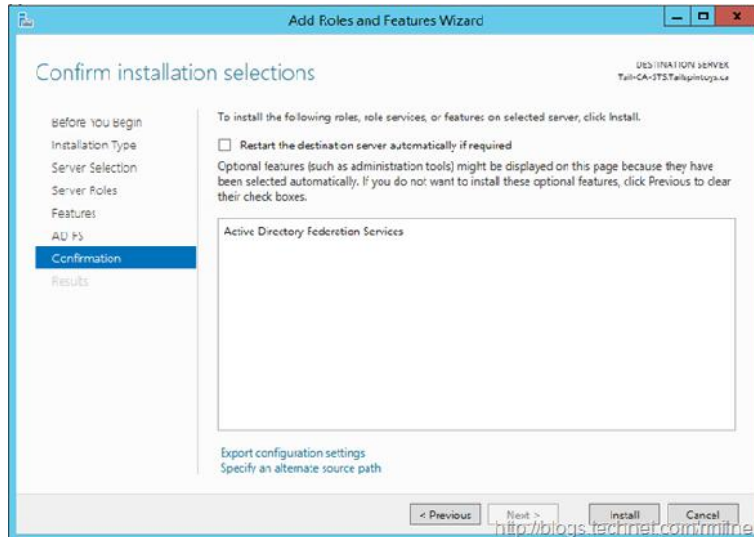


Then follow the instructions:



No additional feature is needed.

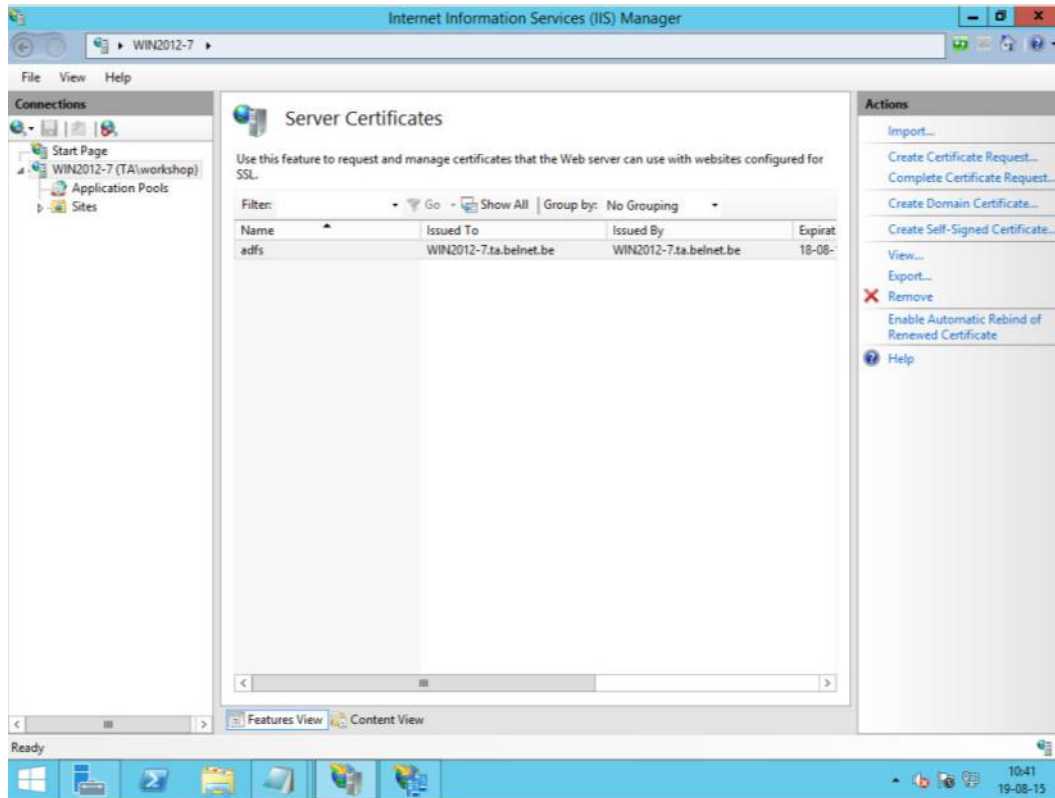




3 Before Configuring AD FS 3.0

3.1 Certificate

You will need a certificate for the AD FS implementation to work. In this document we will create a self-signed certificate that we will generate from IIS8



This certificate will be used for accessing the AD FS server.

You can also download your own certificate created specifically for the AD FS implementation.

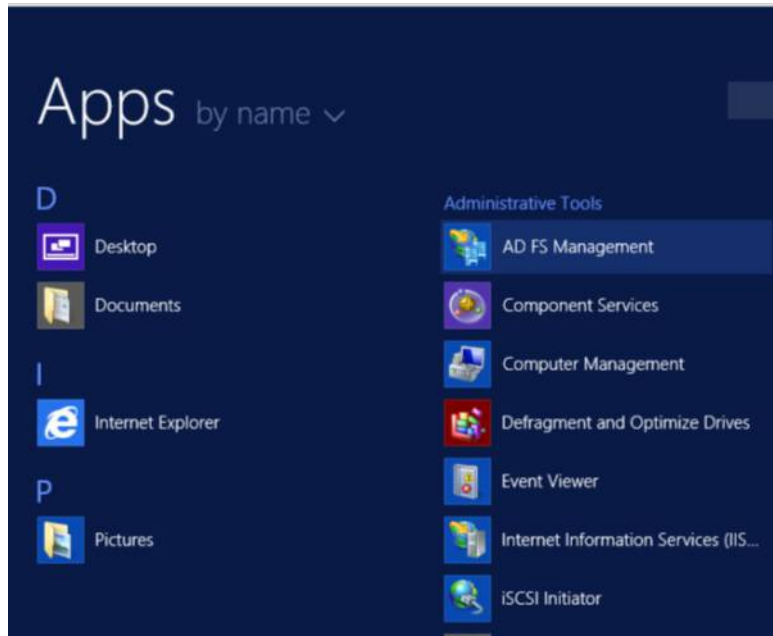
Other certificate will be automatically generated to sign the sending of the attributes.

4 Configure AD FS 3.0

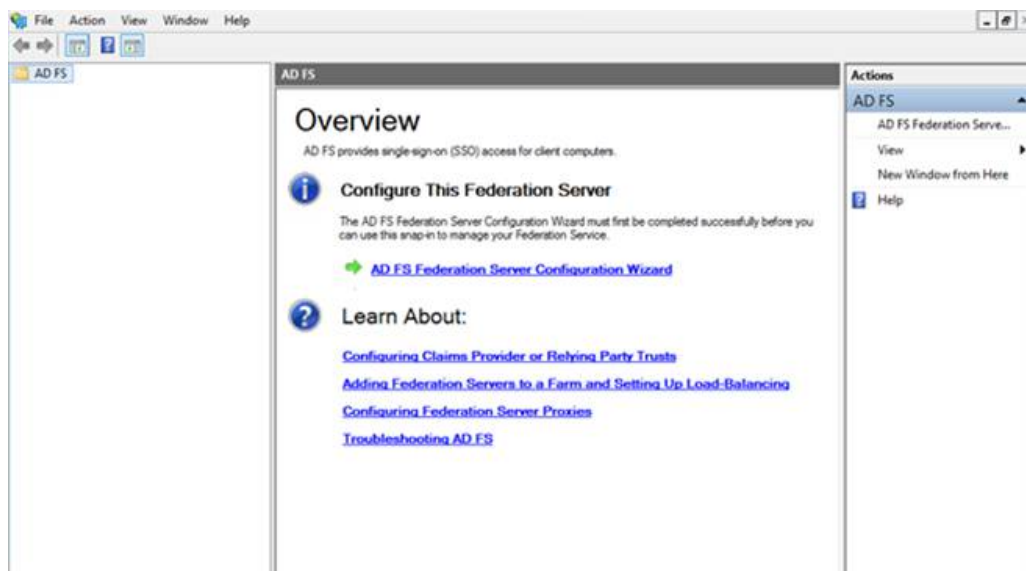
4.1 General configuration of the AD FS server

In order to configure AD FS 3.0 you will need **Domain Admin** rights

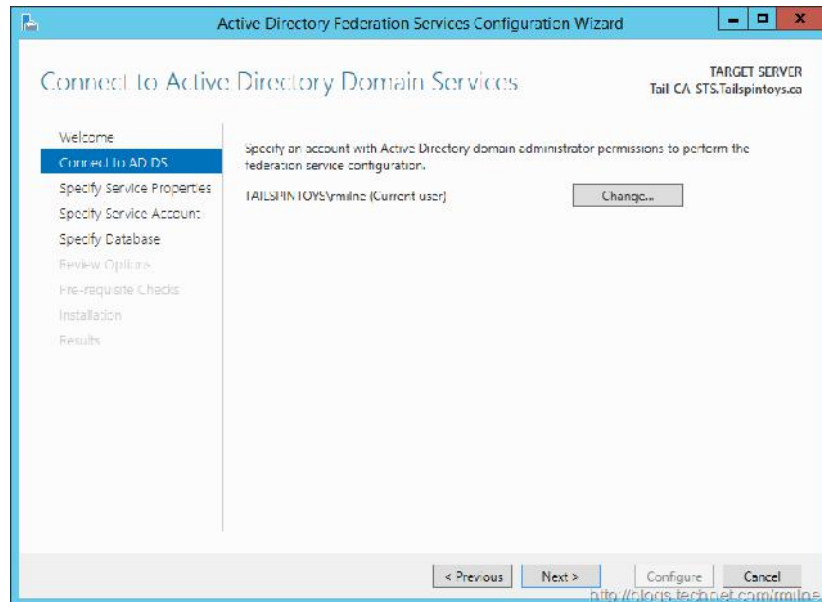
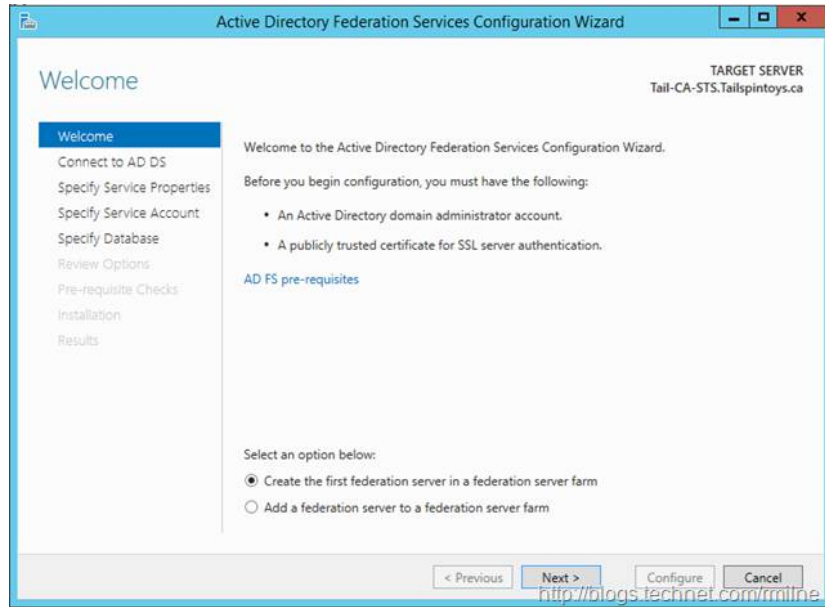
You can now go in the AD FS management snap-in

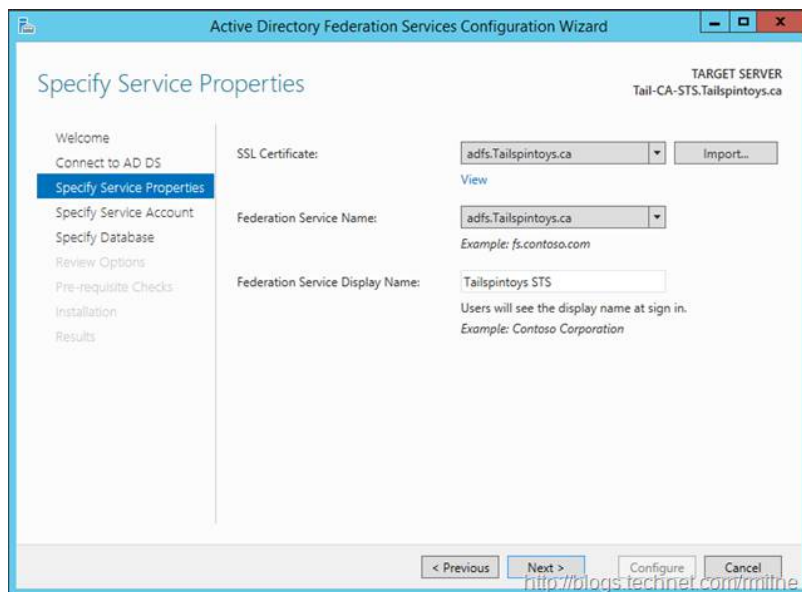


You can start configuring your AD FS server by clicking on the configuration wizard

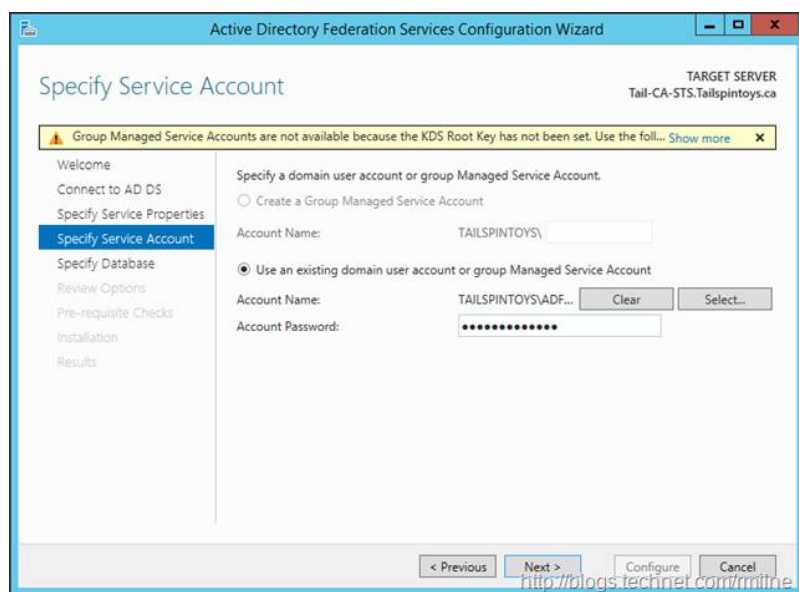


We will now create a new federation with one standalone AD FS server

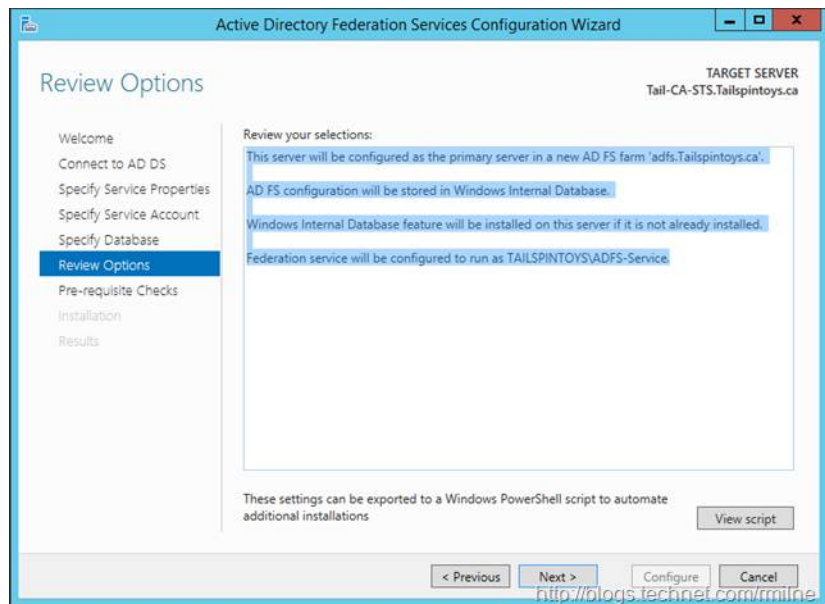
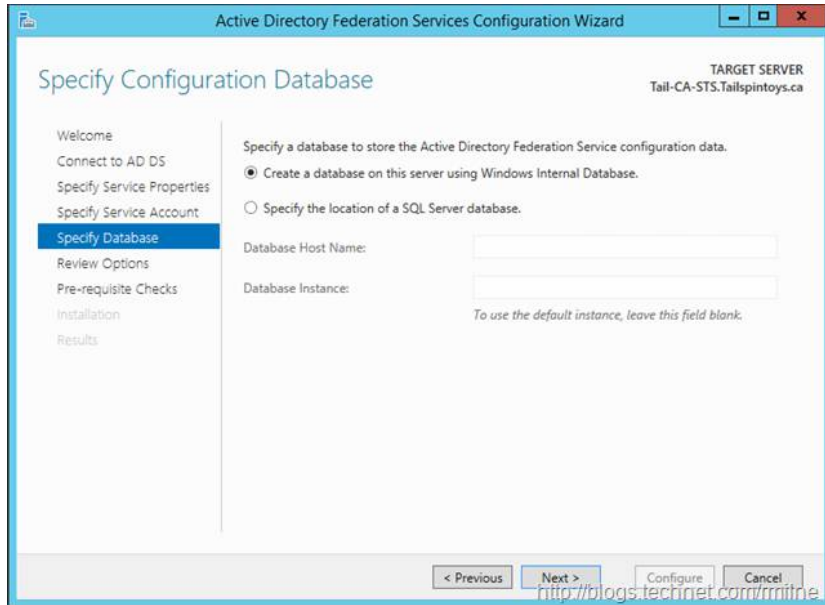


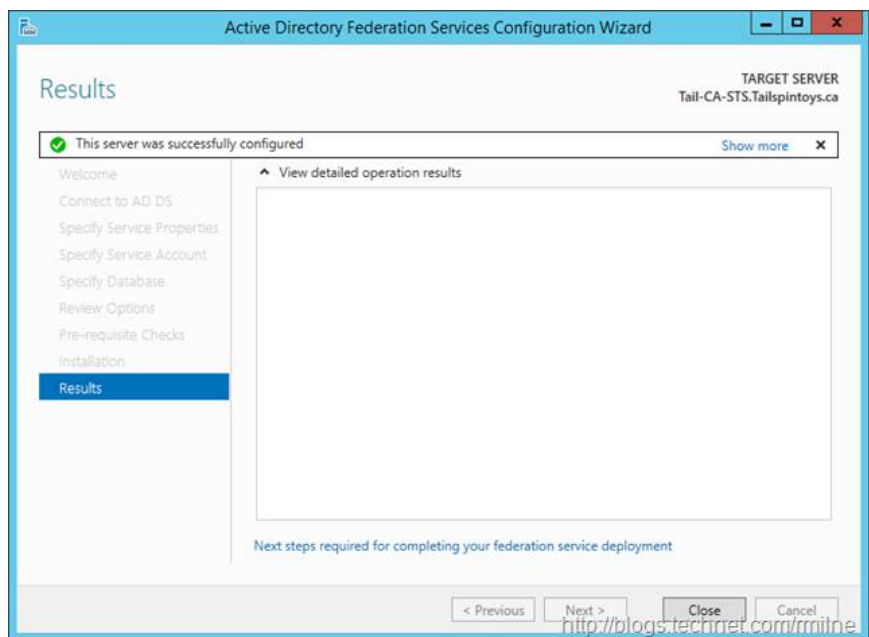
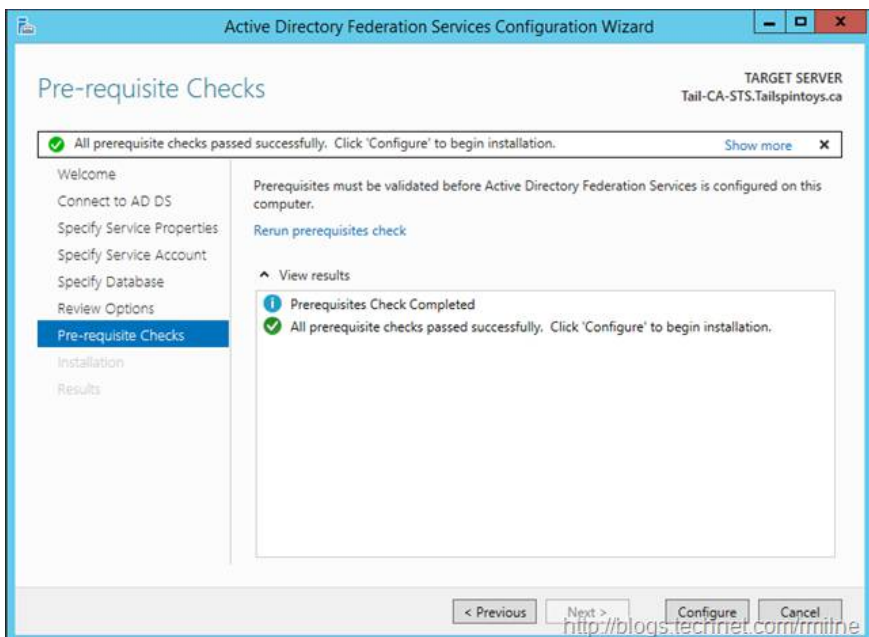


You should see here the Self Signed certificate that you created. Please note that the name of your server **MUST** match the name in your certificate (actually, you can't change it anyway)



If you plan to use a group Managed Service account, you will need a win2012 server AD. In this document we will be using a windows 2008R2 server so we will simply use a service account.

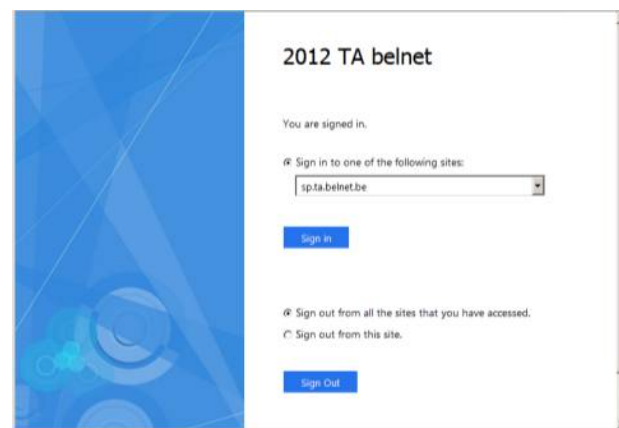




You can test the installation by following the url <https://hostname/adfs/ls/idpinitiatesignon.htm>



You can test here the authentication.

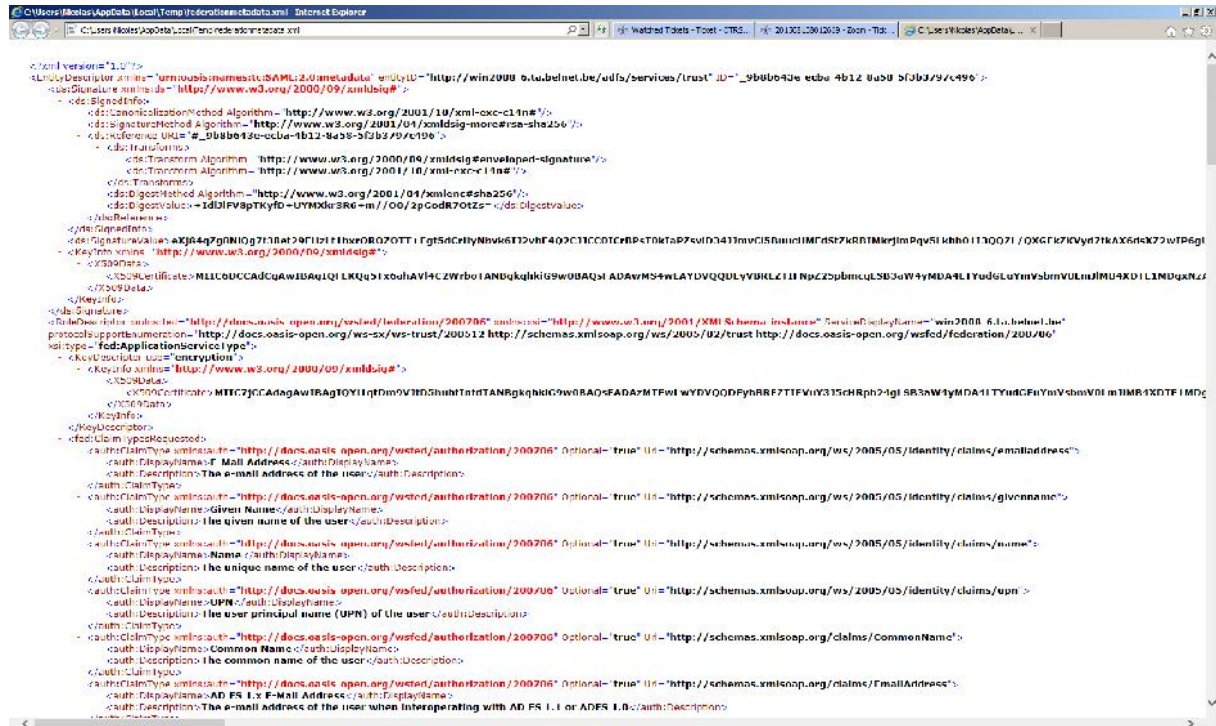


4.2 Upload your Metadata

If the installation has been completed successfully, to get your metadata, follow this URL:

<https://hostname/federationmetadata/2007-06/federationmetadata.xml> where hostname should be replaced by the name of your server.

You should see :



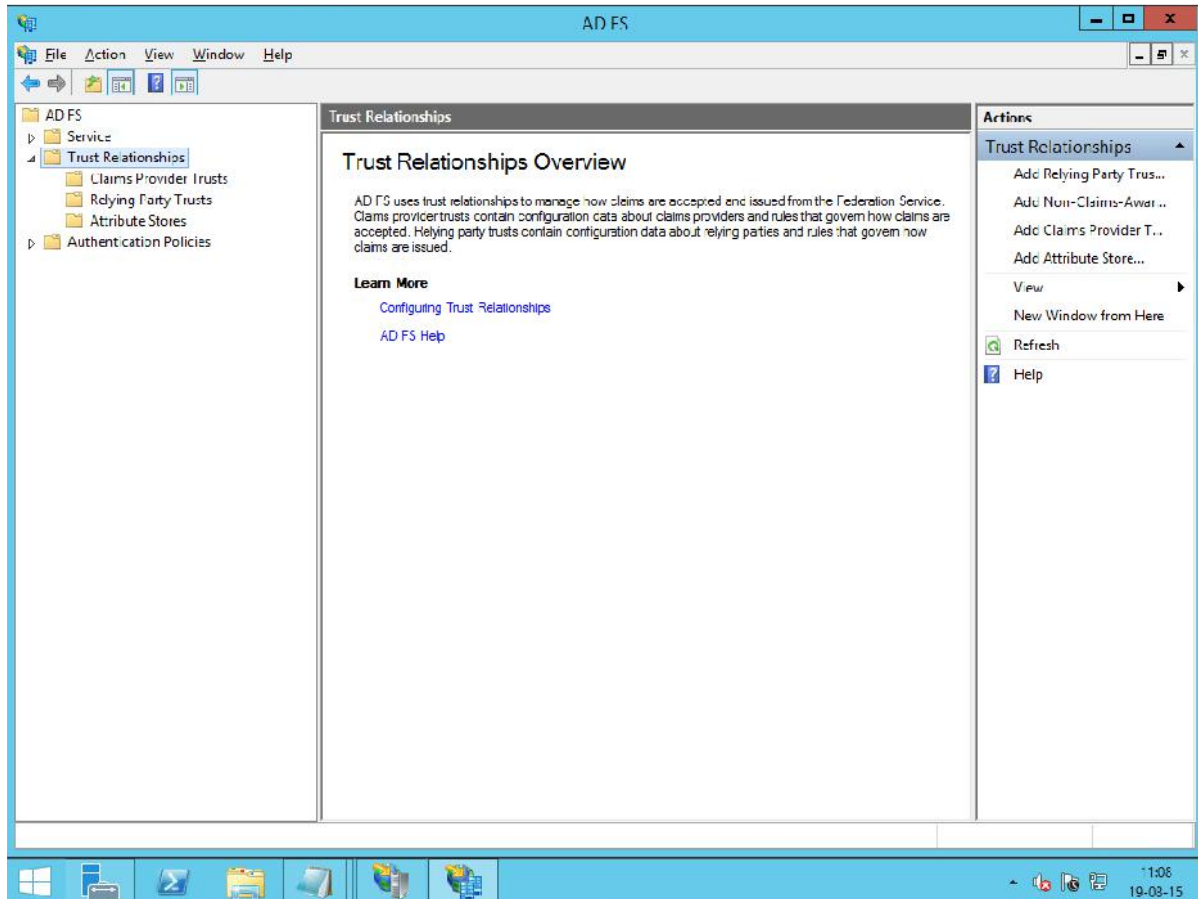
Prior to upload your metadata to the Belnet federation, you must clean it up as it contains a lot of information dedicated to Microsoft federation:

- Remove all tags between <ds:Signature ...> </ds:Signature>
- Remove all tags between: <Role Descriptor xmlns:fed....></Role Descriptor> x2
- Add scoped element:
 - <EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:shibmd="urn:mace:shibboleth:metadata:1.0" entityID="..."
 -
 - Just after <IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
 - <Extensions><shibmd:Scope regexp="false">ta.belnet.be</shibmd:Scope></Extensions>
- Keep the rest

You can upload this to <https://federation.belnet.be>

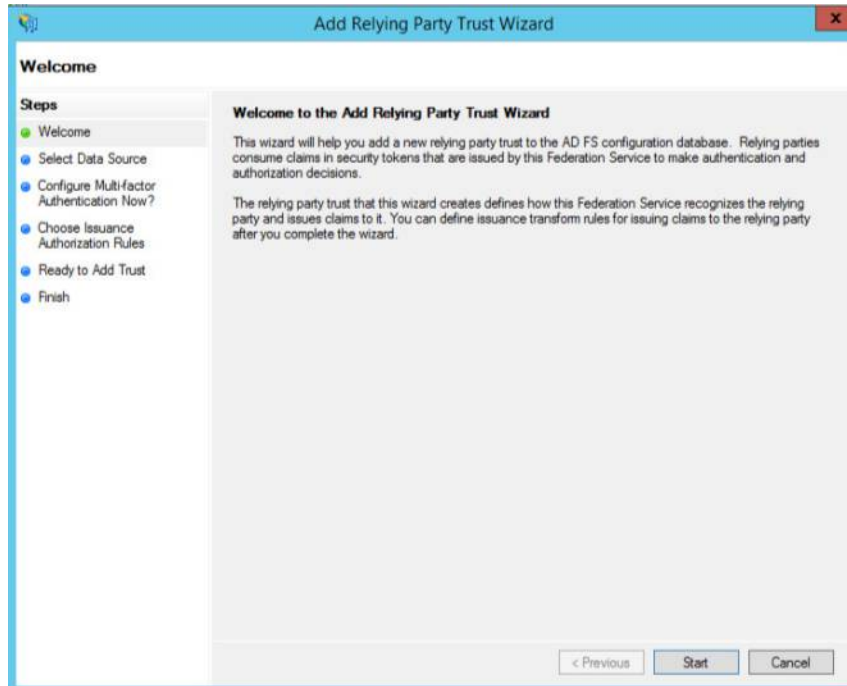
4.3 Configure the Service provider

Open the AD FS management console:



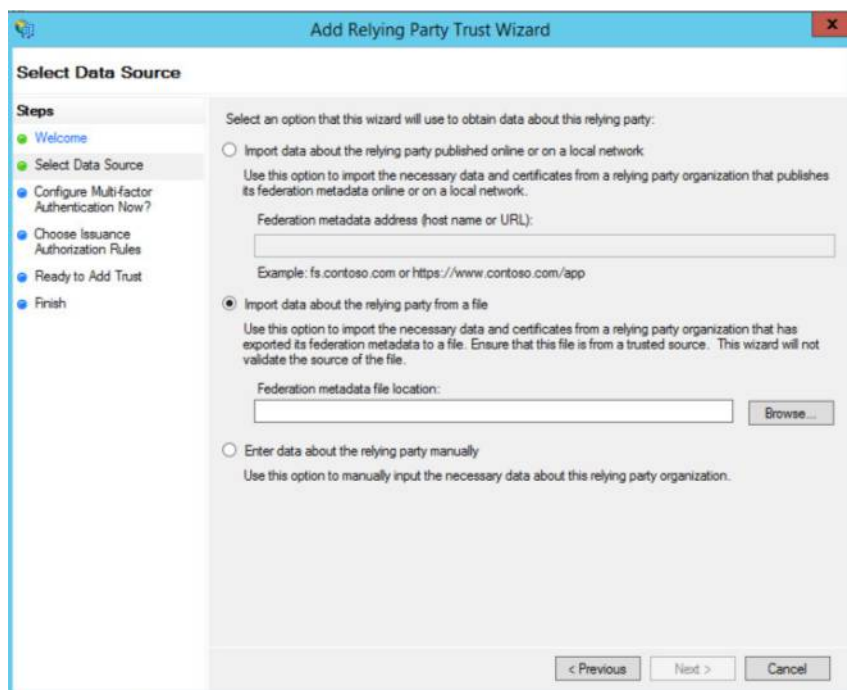
The service provider is configured under “Relaying Party Trust”

In our case, we will add a relying party trust called sp.ta.belnet.be:

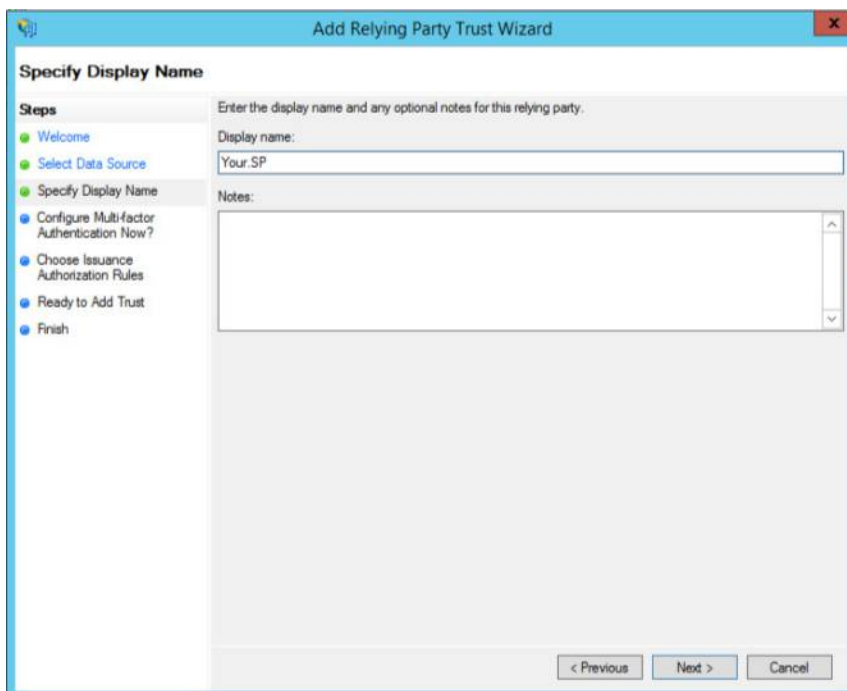


The biggest problem with AD FS is that it can't read the federated metadata. So you will have to get the metadata related to your SP. You will have to do this for every single SP you want to work with.

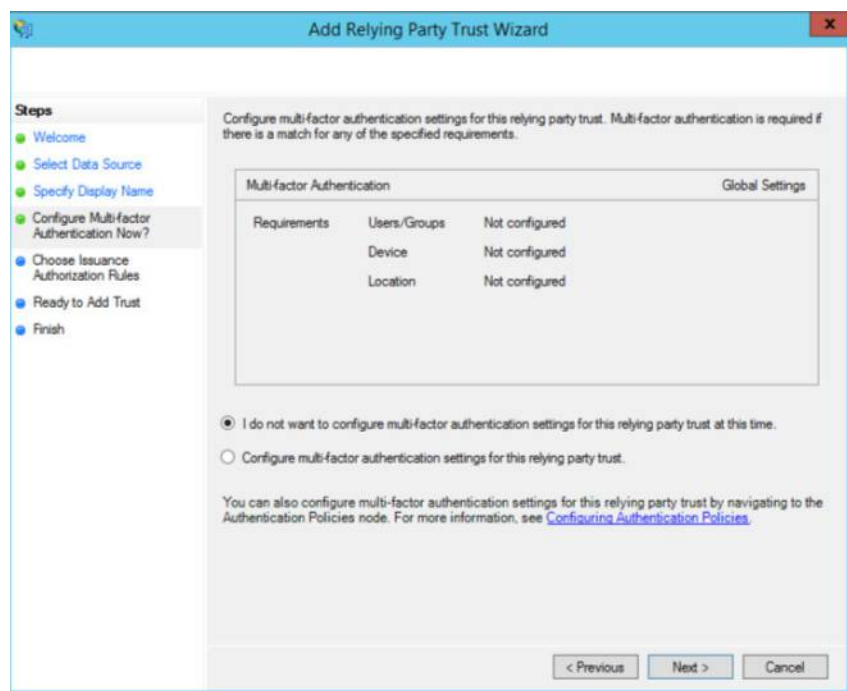
In the next step you can download the metadata (either from a URL or from a file). I would avoid the "manual" part:



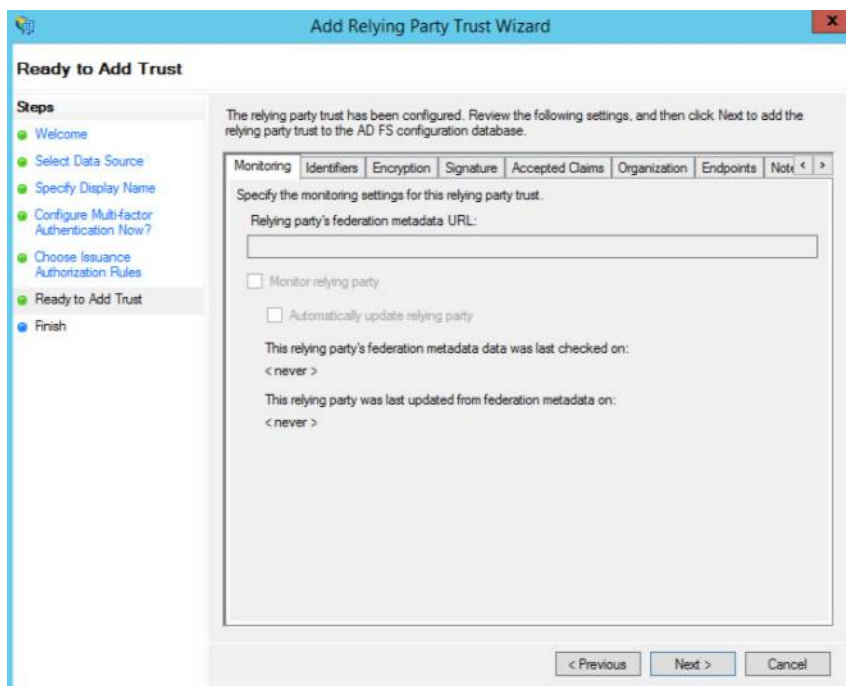
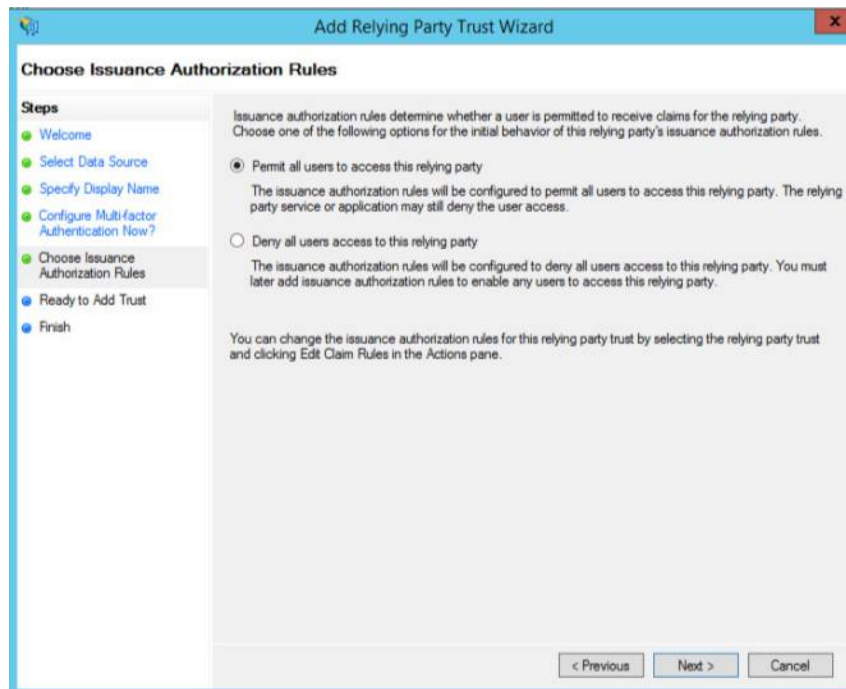
Give your SP a name and some information:



You can configure Multi-Factor Authentication. This is not covered in this document.



Permit all users by default:

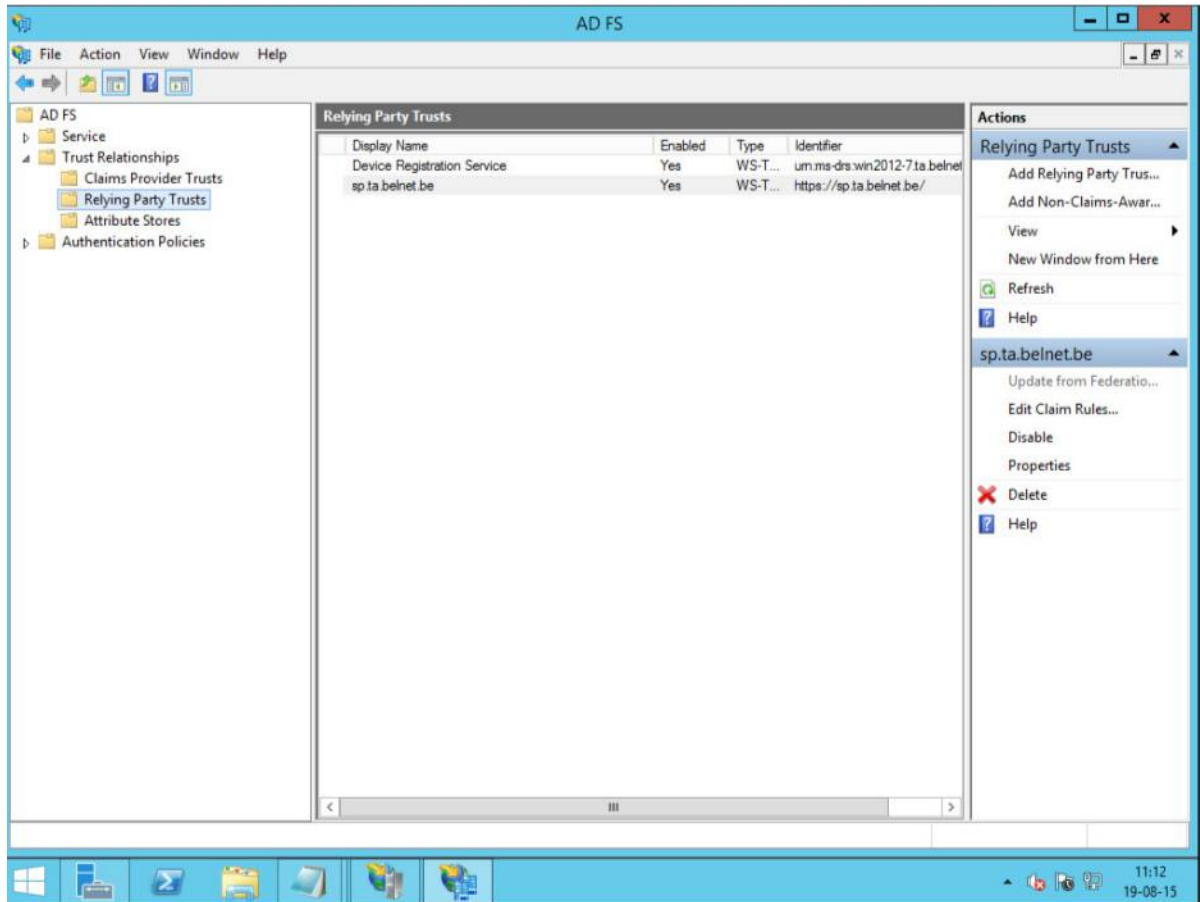


And you're done (well, to define the SP).

You can now test the authentication. You should be able to login with no attributes being sent to the SP.

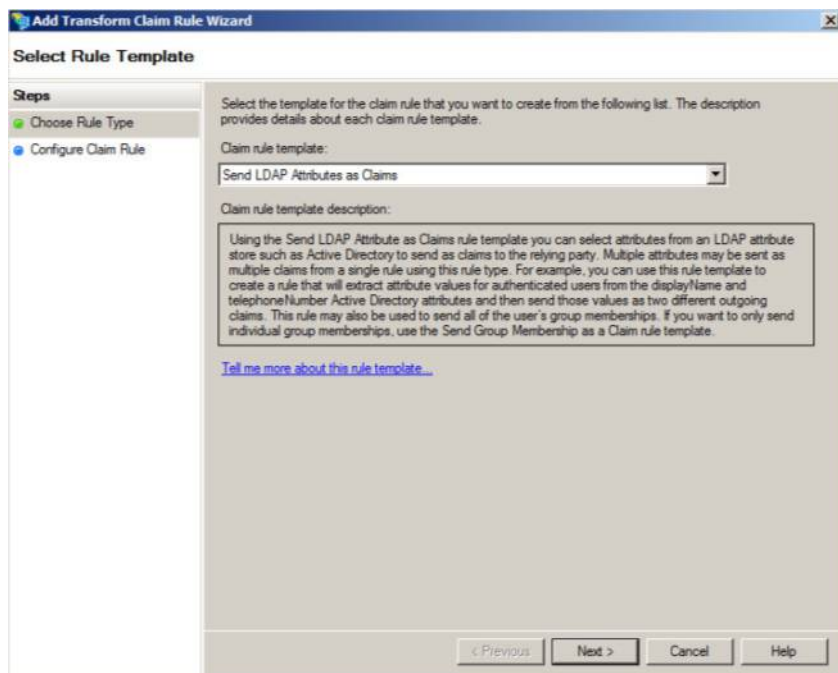
4.4 Configure the attribute release

Now, go and edit the “Claims Rules” (that’s the way attribute release policy is called)

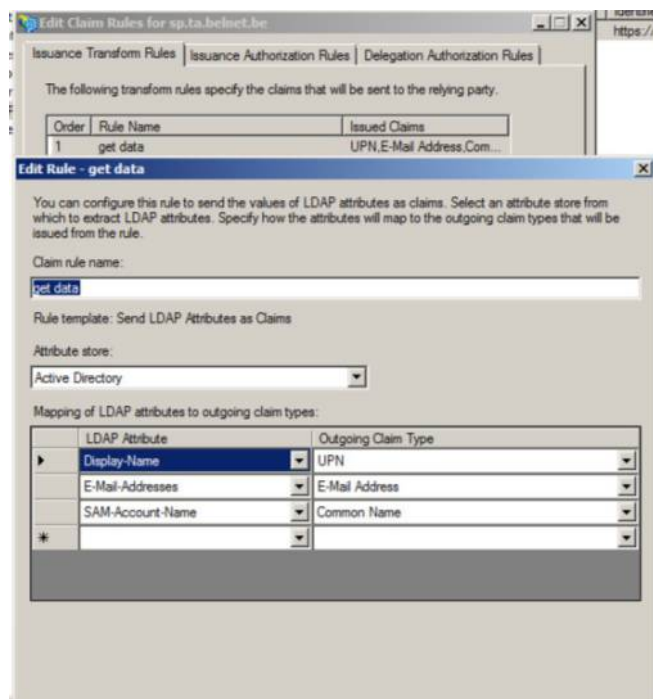


And this is where the fun begins.

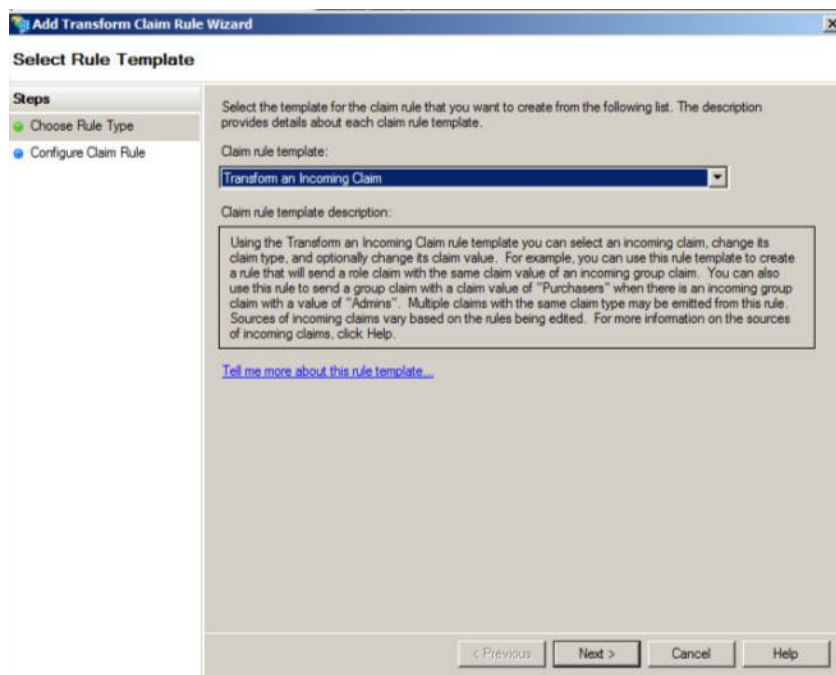
Click on add rules in the “Issuance Transform Rules”



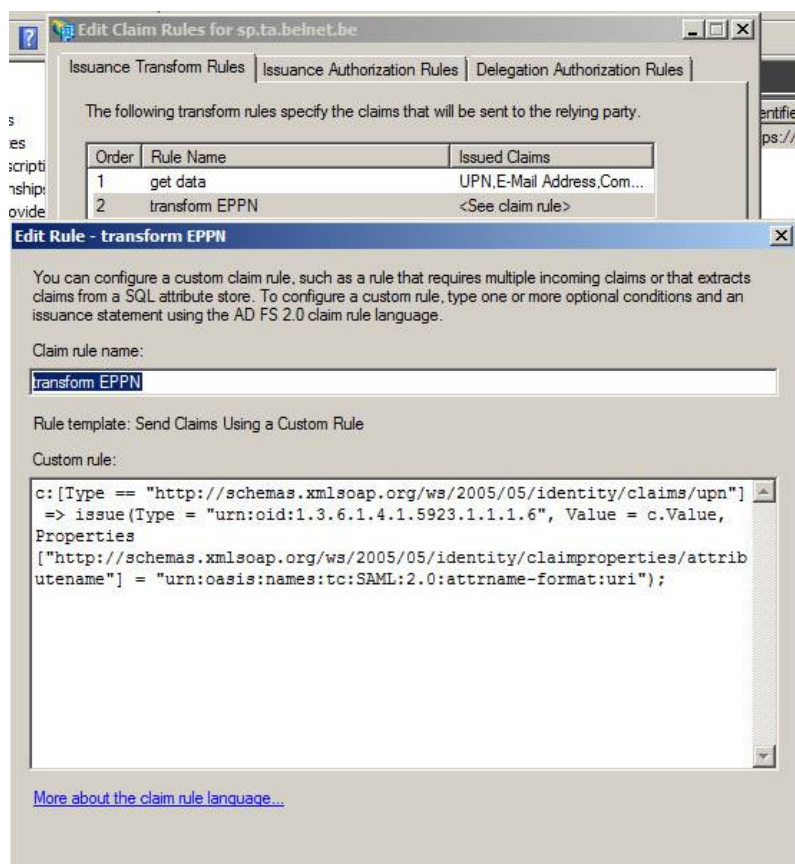
The goal is to release 3 attributes: **EPPN**, **CN** and **email**. Of course, **EPPN** doesn't exist in windows AD FS by so temporarily we will use the **UPN** (user principal name) instead.



And of course, out of the box, those attributes will not be understood by shibboleth so we have to play a bit. We will create “transform rules” in order to translate an attribute from the AD into an attribute known by Shibboleth.

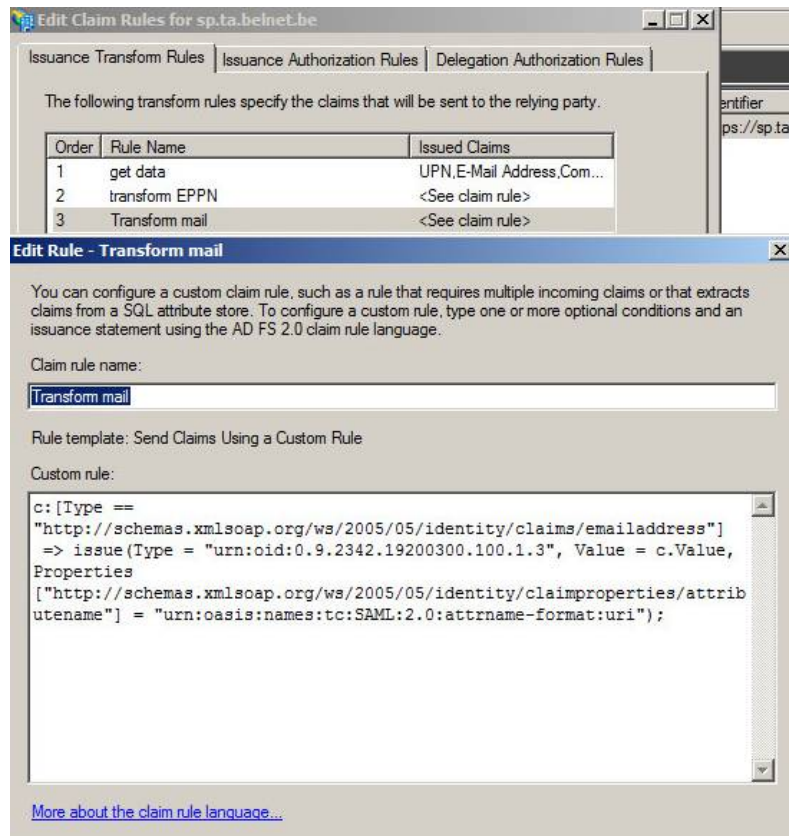


First UPN to EPPN:



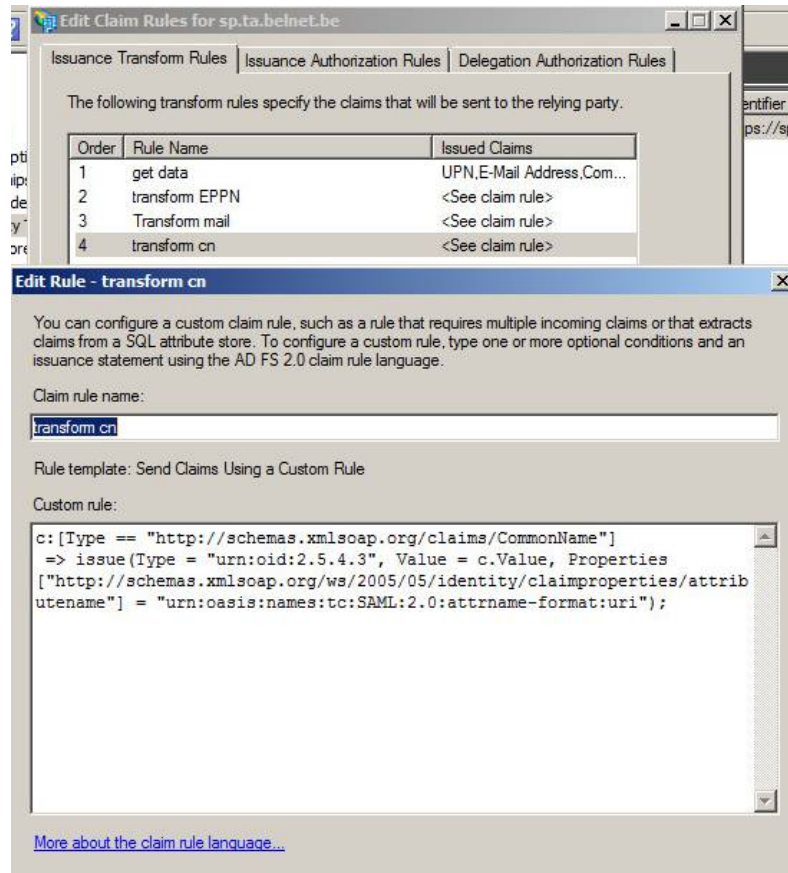
Here, we translate the claim **UPN** into SAML attribute with **OID 1.3.6.1.4.1.5923.1.1.1.6**. (Normally, your service provider should be able to provide you this information as those attribute mappings are defined in the Shibboleth SP (attribute-map.xml). "c.Value" is the value that was taken from the AD.

We will do the same for mail:



Here, we translate the claim **emailaddress** into SAML attribute with **OID 0.9.2342.19200300.100.1.3**.

And finally the cn:



Here, we translate the claim **CommonName** into SAML attribute with **OID 2.5.4.3**.

FYI, some extract of the SP "attribute-map.xml"

```
*****
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
  <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="urn:oid:2.5.4.3" id="cn"/>
<Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
*****
```

You can now try to login again and you should see your attributes:

Welcome on the IDP test page

CONGRATULATIONS! you have successfully logged in.

But this was only the first part of the hands-on.

The purpose of this SP is to verify the release of some specific attributes.

During this workshop, we will focus on 3 attributes (the same that are needed for fileSender): cf In order to properly understand the mechanism of the attribute release policy, you should make it available one by one.

EduPersonPrincipalName

Received eppn: workshop@ta.belnet.be

E-mail address

Received e-mail address: workshop@ta.belnet.be

Common name

◀

5 Some “claim rules” for Belnet SP:

5.1 sptest.belnet.be:

1) Get attribute (UPN, E-mail-Address, Common Name)

2) Transform UPN to EPPN:

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"]
=> issue(Type = "urn:oid:1.3.6.1.4.1.5923.1.1.1.6", Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

3) Transform email

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"]
=> issue(Type = "urn:oid:0.9.2342.19200300.100.1.3", Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

4) Transform cn

```
c:[Type == "http://schemas.xmlsoap.org/claims/CommonName"]
=> issue(Type = "urn:oid:2.5.4.3", Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

5.2 filesender.belnet.be

1) Get attribute (UPN, E-mail-Address, Common Name)

2) Transform namedID

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"]
=> issue(Type = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier",
Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/format"] =
"urn:oasis:names:tc:SAML:2.0:nameid-format:transient");
```

3) Transform UPN to EPPN:

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"]
issue(Type = "urn:oid:1.3.6.1.4.1.5923.1.1.1.6", Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

4) Transform email

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"]
issue(Type = "urn:oid:0.9.2342.19200300.100.1.3", Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

5) Transform cn

```
c:[Type == "http://schemas.xmlsoap.org/claims/CommonName"]
issue(Type = "urn:oid:2.5.4.3", Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

5.3 mconf.belnet.be

1) Get attribute (UPN, E-mail-Address, Common Name)

2) Transform UPN to EPPN:

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"]
=> issue(Type = "urn:oid:1.3.6.1.4.1.5923.1.1.1.6", Value = c.Value + "@yourdomain.be",
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
```

```
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

3) Transform email

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"]  
=> issue(Type = "urn:oid:0.9.2342.19200300.100.1.3", Value = c.Value,  
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =  
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

4) Transform cn

```
c:[Type == "http://schemas.xmlsoap.org/claims/CommonName"]  
=> issue(Type = "urn:oid:2.5.4.3", Value = c.Value,  
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =  
"urn:oasis:names:tc:SAML:2.0:attrname-format:uri");
```

6 Interesting docs:

- <https://technet.microsoft.com/en-us/library/hh831502.aspx>
- <http://blogs.technet.com/b/rmilne/archive/2014/04/28/how-to-install-ads-2012-r2-for-office-365.aspx>
-