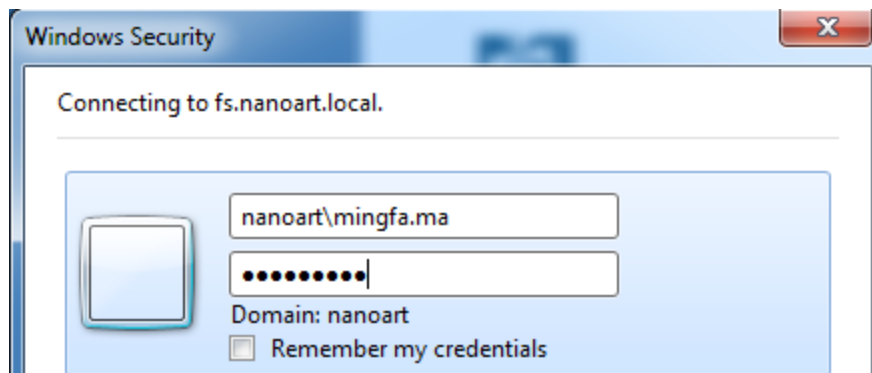


[Claims-Aware Applications](#), such as SharePoint, Lync 2013(passive authentication), are generally authenticated through Active Directory with ADFS.



After authentication (here is Windows integrated authentication), you can get to see the web content

https://adfstest.nanoart.local/ Certificate error Home Page

MY CLAIMSWEB

Home About

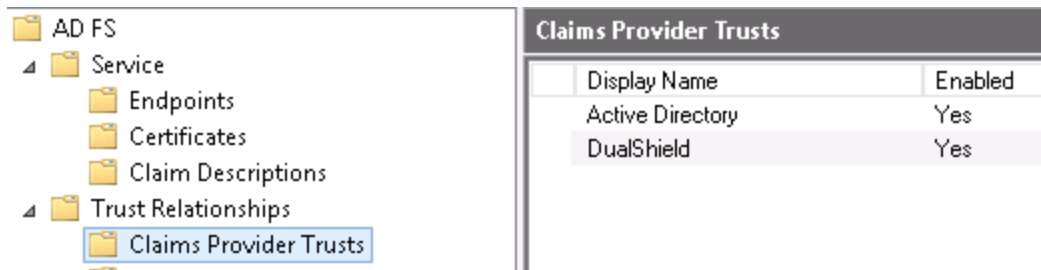
WELCOME CLAIMSWEB!

📄 🏠

Issued Identity	
Claim Type	Claim Value
http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod	http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant
http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant	2014-04-07T12:04:37Z

SAML Token	
⊕ Raw SAML Token	
Property	Value
SamlSecurityToken.Id	_cc000897-ae9-4042-a475-03ed74f5b99c
SamlSecurityToken.ValidFrom	4/7/2014 2:04:37 PM
SamlSecurityToken.ValidTo	4/7/2014 3:04:37 PM (60 minutes)
SamlSecurityToken.Assertion.AssertionId	_cc000897-ae9-4042-a475-03ed74f5b99c
SamlSecurityToken.Assertion.Issuer	http://fs.nanoart.local/adfs/services/trust
SamlSecurityToken.Assertion.IssueInstant	4/7/2014 2:04:37 PM
Intended Audience	https://adfstest.nanoart.local/

You can extend it to use DualShield to do the multi-factor authentication by adding DualShield as a **Claims Provider Trust**.



Now if you access the web application in browser

<https://adfstest.nanoart.local>

You will see the claim provider (IDP) selection (**Nano Art** in this sample is the built-in Active Directory).

Nano Art

Sign in with one of these accounts



Nano Art



DualShield

Select DualShield, it redirects to DualShield SSO authentication page.

The screenshot shows a web browser window with the address bar displaying `https://dualshield.deepnet...`. The page features the **DUALSHIELD** logo with the tagline **UNIFIED AUTHENTICATION**. Below the header, the text **Log in to application: owa** is displayed. A login form contains a **Login Name:** label and an adjacent text input field. A **Continue** button is positioned below the input field.

Once you finish the authentication, you will see the final web content.

MY CLAIMSWeb

Home About

WELCOME CLAIMSWEB!

Issued Identity

Claim Type	Claim Value
http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod	http://schemas.microsoft.com/ws/2008/06/ident
http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant	2014-04-07T14:45:19.606Z

SAML Token

Raw SAML Token

Property	Value
SamlSecurityToken.Id	_dc7323c3-ba82-4a6c-9532-05036a152636
SamlSecurityToken.ValidFrom	4/7/2014 2:45:53 PM
SamlSecurityToken.ValidTo	4/7/2014 3:45:53 PM (60 minutes)
SamlSecurityToken.Assertion.AssertionId	_dc7323c3-ba82-4a6c-9532-05036a152636
SamlSecurityToken.Assertion.Issuer	http://fs.nanoart.local/adfs/services/trust
SamlSecurityToken.Assertion.IssueInstant	4/7/2014 2:45:53 PM
Intended Audience	https://adfctest.nanoart.local/
SamlSecurityToken.Assertion.MinorVersion	1
SamlSecurityToken.Assertion.MajorVersion	1
Signature Algorithm	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Signing Certificate	[Subject] CN=ADFS Signing - fs.nanoart.local [Issuer] CN=ADFS Signing - fs.nanoart.local [Serial Number] 2220D0556A8D61B548C9325379AF603D [Not Before] 4/3/2014 9:54:03 AM [Not After] 4/3/2015 9:54:03 AM [Thumbprint] 834A5AC753F91342A7DBE8C0C170A21964A776ED

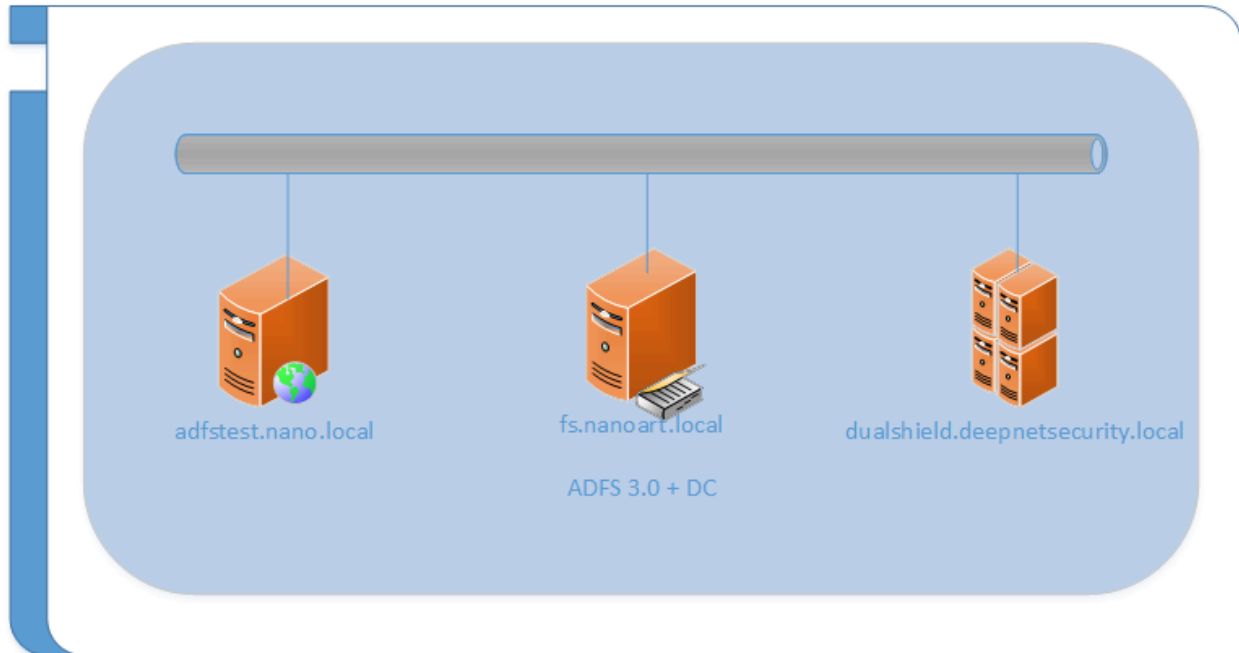
[Download Certificate](#)

Now you may wonder how it works, how you can set up a test lab?

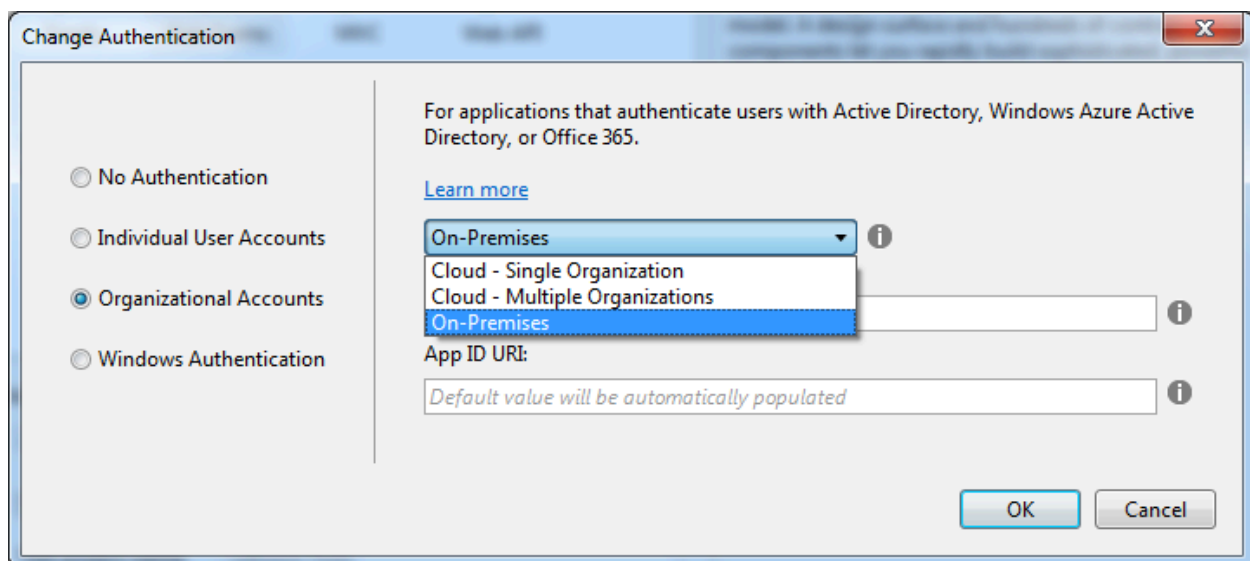
Keep reading.

You need 3 servers

- DualShield
- ADFS + DC. ADFS 3.0 can be installed on DC
- Web application server (IIS)



You can use Visual Studio to create a ASP.NET Web Application. In VS 2013, you can specify the authentication



For simplicity, here we use the instruction and package from [How to Build Your ADFS Lab on Server 2012, Part2: Web SSO](#). It is much easier to use its powershell script to deploy a ASP.NET application.

In theory, the web server is not necessary to be a member server of AD, but let us just follow the instruction and prerequisites. Make sure `ADFSserver dc01.nanoart.local` is accessible from this web server machine.

```
PS C:\temp> .\deploy-testsite.ps1 -SourcePath C:\temp\deploy -SiteName adfstest
-SitePhysicalPath c:\sites\adfstest -ADFSserver dc01.nanoart.local -AppFQDN
adfstest.nanoart.local
```

You can look at the content of the script to understand what it is trying to do, but how it makes a web as claims-aware application? the answer is, it modifies the web.config file

```
<microsoft.identityModel>
  <service saveBootstrapTokens="true">
    <audienceUris>
      <add value="https://adfstest.nanoart.local/" />
    </audienceUris>
    <federatedAuthentication>
      <wsFederation passiveRedirectEnabled="true"
issuer="https://fs.nanoart.local/adfs/ls/" realm="https://adfstest.nanoart.local"
requireHttps="true" />
      <cookieHandler requireSsl="true" />
    </federatedAuthentication>
    <applicationService>
      <claimTypeRequired>
        <!--Following are the claims offered by STS
'http://fs.nanoart.local/adfs/services/trust'. Add or uncomment claims that you
require by your application and then update the federation metadata of this
application.-->
        <claimType
type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" optional="true" />
        <claimType
type="http://schemas.microsoft.com/ws/2008/06/identity/claims/role" optional="true" />
        <!--<claimType
type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
optional="true" />-->
      </claimTypeRequired>
    </applicationService>
    <certificateValidation certificateValidationMode="None" />
    <issuerNameRegistry
type="Microsoft.IdentityModel.Tokens.ConfigurationBasedIssuerNameRegistry,
Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35">
      <trustedIssuers>
        <add thumbprint="834A5AC753F91342A7DBE8C0C170A21964A776ED"
name="http://fs.nanoart.local/adfs/services/trust" />
      </trustedIssuers>
    </issuerNameRegistry>
  </service>
</microsoft.identityModel>
```

Now it is time to add the web application as a **Relying Party Trust** in ADFS. Choose the option "**Import data about the relying party from a file**". This file `FederationMetadata.xml` is under the folder `C:\sites\adfstest\FederationMetadata\2007-06`, You may be keen to see what the content is,

```
<?xml version="1.0" encoding="utf-8"?>
<EntityDescriptor ID="_0db3255c-8a7c-4fa0-a7c8-cd49d89866d6"
entityID="https://adfstest.nanoart.local/"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
  <RoleDescriptor xsi:type="fed:ApplicationServiceType"
xmlns:fed="http://docs.oasis-open.org/wsfed/federation/200706"
protocolSupportEnumeration="http://docs.oasis-open.org/wsfed/federation/200706"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <fed:ClaimTypesRequested>
      <auth:ClaimType
Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" Optional="true"
xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706" />
      <auth:ClaimType
Uri="http://schemas.microsoft.com/ws/2008/06/identity/claims/role" Optional="true"
xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706" />
    </fed:ClaimTypesRequested>
    <fed:TargetScopes>
      <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
        <Address>https://adfstest.nanoart.local/</Address>
      </EndpointReference>
    </fed:TargetScopes>
    <fed:PassiveRequestorEndpoint>
      <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
        <Address>https://adfstest.nanoart.local/</Address>
      </EndpointReference>
    </fed:PassiveRequestorEndpoint>
  </RoleDescriptor>
</EntityDescriptor>
```

Add Relying Party Trust Wizard

Select Data Source

Steps

- Welcome
- Select Data Source
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Select an option that this wizard will use to obtain data about this relying party:

☐ Import data about the relying party published online or on a local network

Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.

Federation metadata address (host name or URL):

Example: fs.contoso.com or https://www.contoso.com/app

☒ Import data about the relying party from a file

Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.

Federation metadata file location:

C:\temp\FederationMetadata.xml Browse...

☐ Enter data about the relying party manually

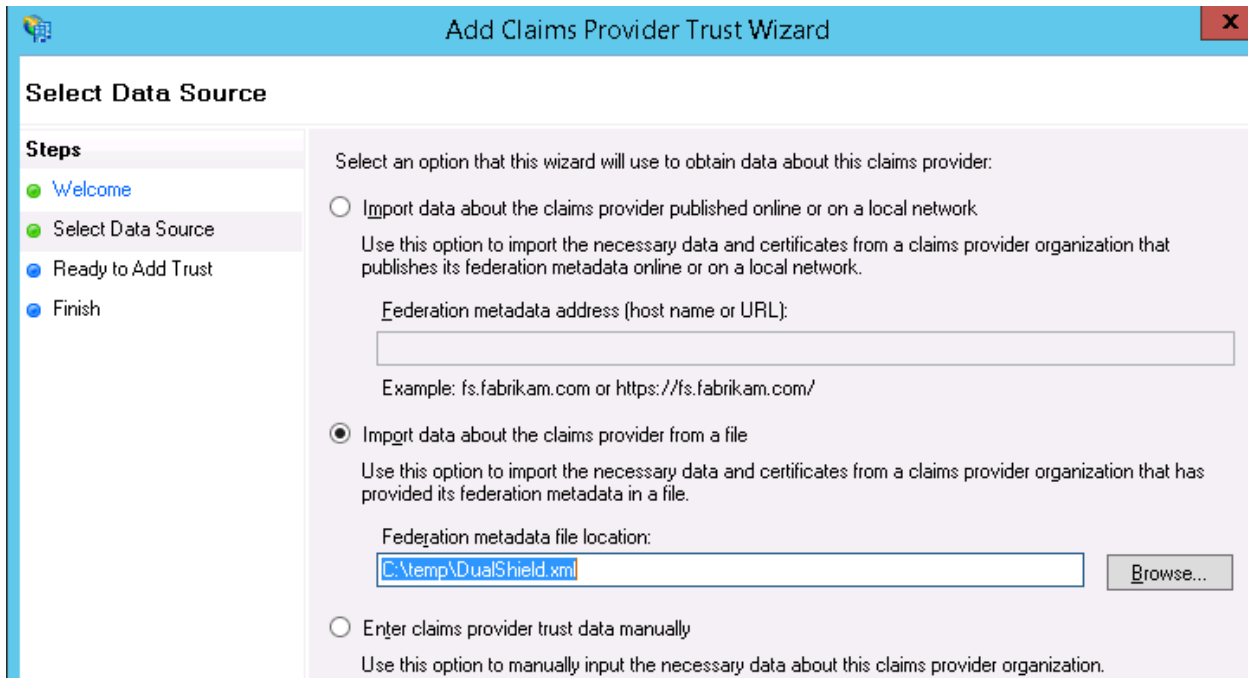
Use this option to manually input the necessary data about this relying party organization.

You should see a newly added **Relying Party Trust** called “**ADFS Test**”.

Relying Party Trusts			
Display Name	Enabled	Type	Identifier
Device Registration Service	Yes	WS-Trust / SAML / WS-Federation	urn:ms-drs:fs.nanoart.local
Lync-Ext	Yes	WS-Trust / SAML / WS-Federation	https://fe02.nanoart.local/
ADFS Test	Yes	WS-Trust / SAML / WS-Federation	https://adfstest.nanoart.local/

It is wise to test the built-in Active Directory authentication right now by accessing <https://adfstest.nanoart.local>

If everything is right, then let us go ahead to configure DualShield as a third party claim provider. Obviously it is done on ADFS server,



Add Claims Provider Trust Wizard

Select Data Source

Steps

- Welcome
- Select Data Source
- Ready to Add Trust
- Finish

Select an option that this wizard will use to obtain data about this claims provider:

☐ Import data about the claims provider published online or on a local network

Use this option to import the necessary data and certificates from a claims provider organization that publishes its federation metadata online or on a local network.

Federation metadata address (host name or URL):

Example: fs.fabrikam.com or https://fs.fabrikam.com/

☒ Import data about the claims provider from a file

Use this option to import the necessary data and certificates from a claims provider organization that has provided its federation metadata in a file.

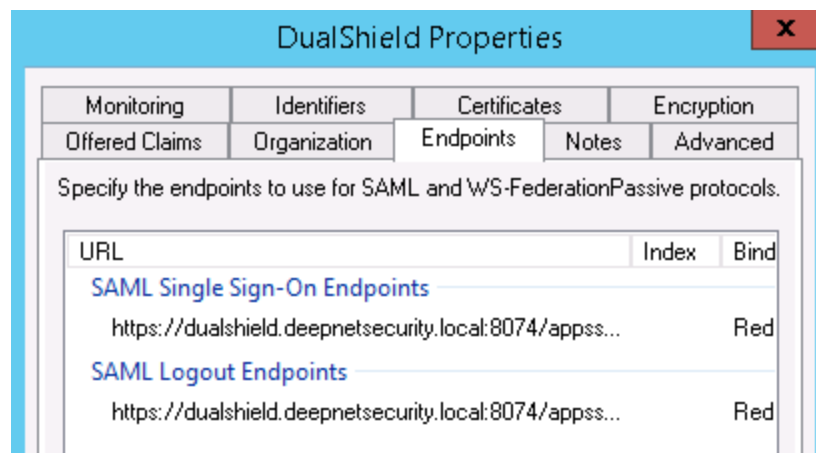
Federation metadata file location:

C:\temp\DualShield.xml Browse...

☐ Enter claims provider trust data manually

Use this option to manually input the necessary data about this claims provider organization.

The file `DualShield.xml` is an IDP metadata file which is downloaded from DualShield server. You must have **SSL** on DualShield SSO port, otherwise, the import won't finish successfully.



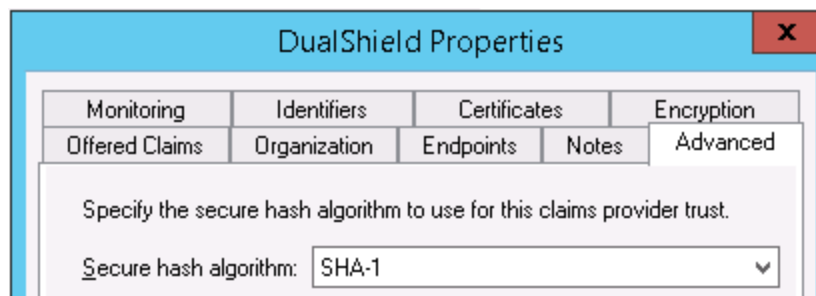
DualShield Properties

Monitoring	Identifiers	Certificates	Encryption
Offered Claims	Organization	Endpoints	Notes

Specify the endpoints to use for SAML and WS-FederationPassive protocols.

URL	Index	Bind
SAML Single Sign-On Endpoints		
https://dualshield.deepnetsecurity.local:8074/appss...		Red
SAML Logout Endpoints		
https://dualshield.deepnetsecurity.local:8074/appss...		Red

After import, you may need to manually change the hash algorithm to SHA-1



DualShield Properties

Monitoring	Identifiers	Certificates	Encryption
Offered Claims	Organization	Endpoints	Notes

Specify the secure hash algorithm to use for this claims provider trust.

Secure hash algorithm: SHA-1

Are we finished? Not yet, you have to add **ADFS server** as a service provider in DualShield. Certainly you need the metadata of ADFS server, which can be downloaded through

<https://fs.nanoart.local/FederationMetadata/2007-06/FederationMetadata.xml>

Metadata			
Yes	Yes	/adfs/services/trust/mex	WS-MEX
Yes	Yes	/FederationMetadata/2007-06/FederationMetadata.xml	Federation Metadata
Yes	No	/adfs/fs/federationsservice.asmx	ADFS 1.0 Metadata

Unfortunately, DualShield doesn't accept this complicated xml file, so we have to simplify it! Here is the working version.

```
<EntityDescriptor entityID="http://fs.nanoart.local/adfs/services/trust"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <SPSSODescriptor
    AuthnRequestsSigned="false"
    WantAssertionsSigned="true"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://fs.nanoart.local/adfs/ls/" />

    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</NameIDFormat>

    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</NameIDFormat>

    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

    <AssertionConsumerService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://fs.nanoart.local/adfs/ls/" index="0" isDefault="true"/>

  </SPSSODescriptor>
</EntityDescriptor>
```

Now you can test with DualShield Authentication!

In reality, you need to create claim rules to transfer the claims (SAML attributes). Please check the details in [AD FS 2.0 Claims Rule Language Primer](#).

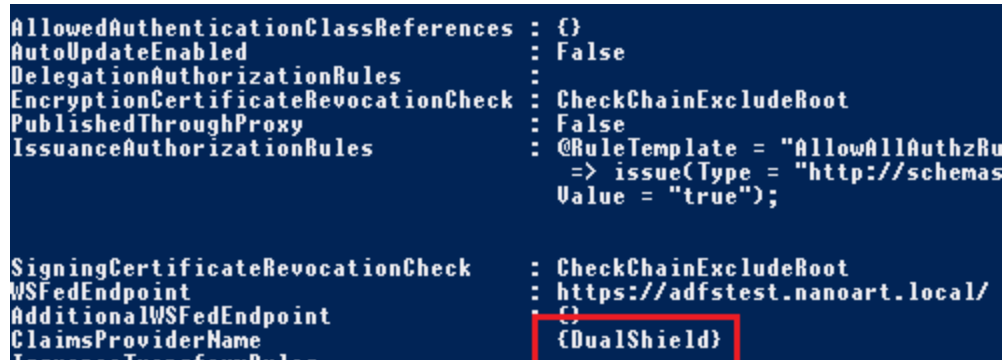
For some scenarios, an organizations might want end users to only see the claims

providers that are specific to an application so that only a subset of claims provider are displayed on the home realm discovery page. Want to use DualShield claim provider directly? use the following Windows PowerShell cmdlet and syntax.

```
Set-AdfsRelyingPartyTrust -Targetname "ADFS Test" -ClaimsProviderName @("DualShield")
```

You can check it with

```
Get-AdfsRelyingPartyTrust -Name "ADFS Test"
```

A screenshot of a PowerShell console window with a dark blue background and white text. It shows the output of the 'Get-AdfsRelyingPartyTrust' command for a trust named 'ADFS Test'. The output lists various properties and their values. The 'ClaimsProviderName' property is highlighted with a red rectangular box, showing its value as '{DualShield}'.

```
AllowedAuthenticationClassReferences : {}
AutoUpdateEnabled                   : False
DelegationAuthorizationRules         :
EncryptionCertificateRevocationCheck : CheckChainExcludeRoot
PublishedThroughProxy                : False
IssuanceAuthorizationRules           : @RuleTemplate = "AllowAllAuthzRu
=> issue(Type = "http://schemas
Value = "true");

SigningCertificateRevocationCheck     : CheckChainExcludeRoot
WSFedEndpoint                        : https://adfstest.nanoart.local/
AdditionalWSFedEndpoint               : {}
ClaimsProviderName                   : {DualShield}
```

References

[Identity Developer Training Kit](#)

[Security Token Visualizer Control for Visual Studio 2010](#)