

## Picketlink with Microsoft ADFSv2

We use the Portecle tool to create the key store file.

Download the Portecle tool using the link (<https://sourceforge.net/projects/portecle>) and extract the download file and use it.

Click the file to open Portecle tool

<Portecle\_Tool\_Extract\_Folder>/ portecle.jar

In the Portecle tool we do the following

- Create a certificate for the JBOSS container
- Export the ADFS Token signing certificate
- Import ADFS Token signing certificate into a key store file
- Export the keystore certificate file and then import it into the certificate store used by ADFS

### List of the content sections

Secure or Claims aware Web Application and picket link with the configuration files

In ADFS relying party configuration, under the signature tab add this certificate (i.e. the certificate with which the Java application was signed)

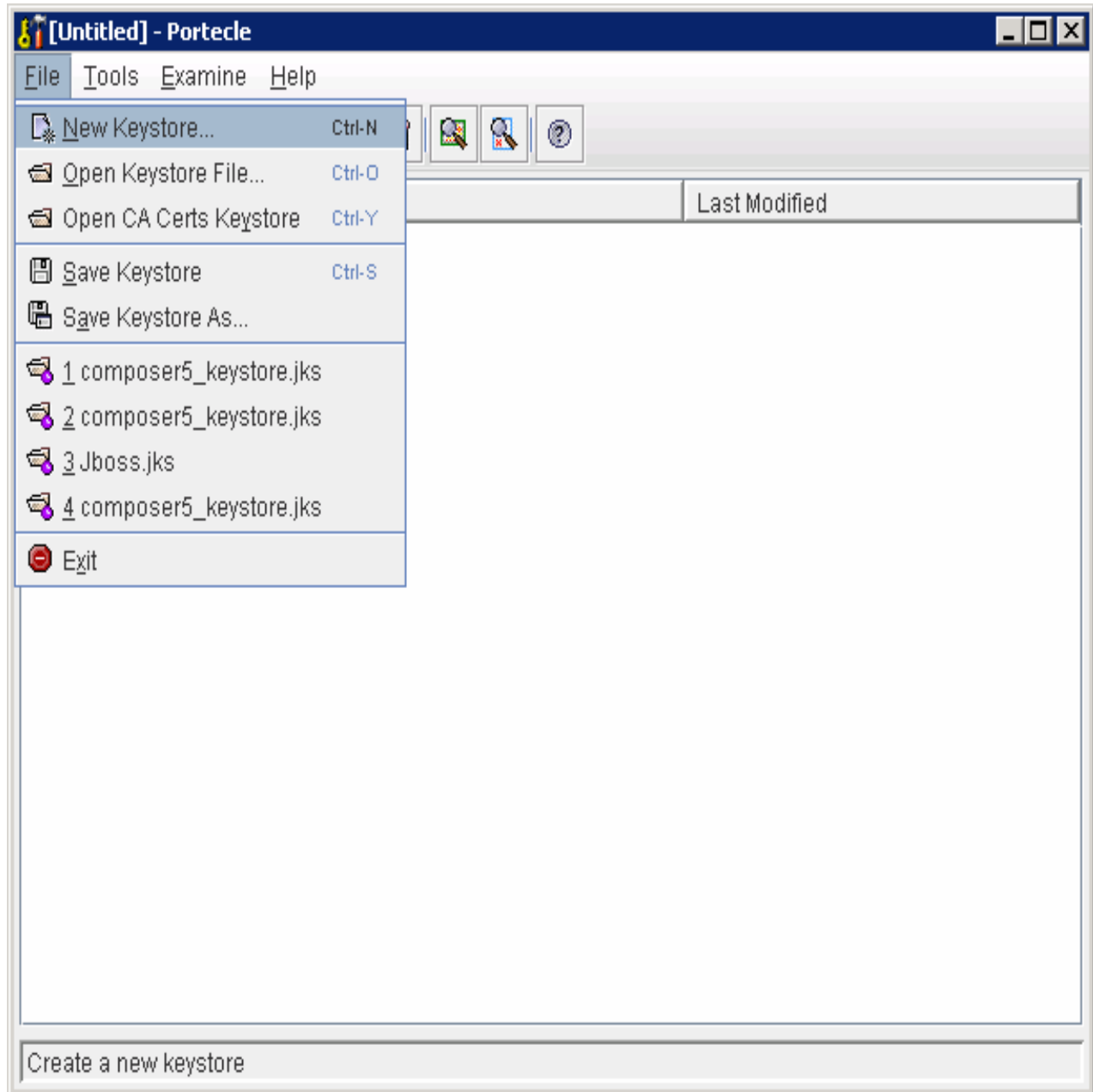
Web application Screen shots.

Add relying party in ADFS 2.0.

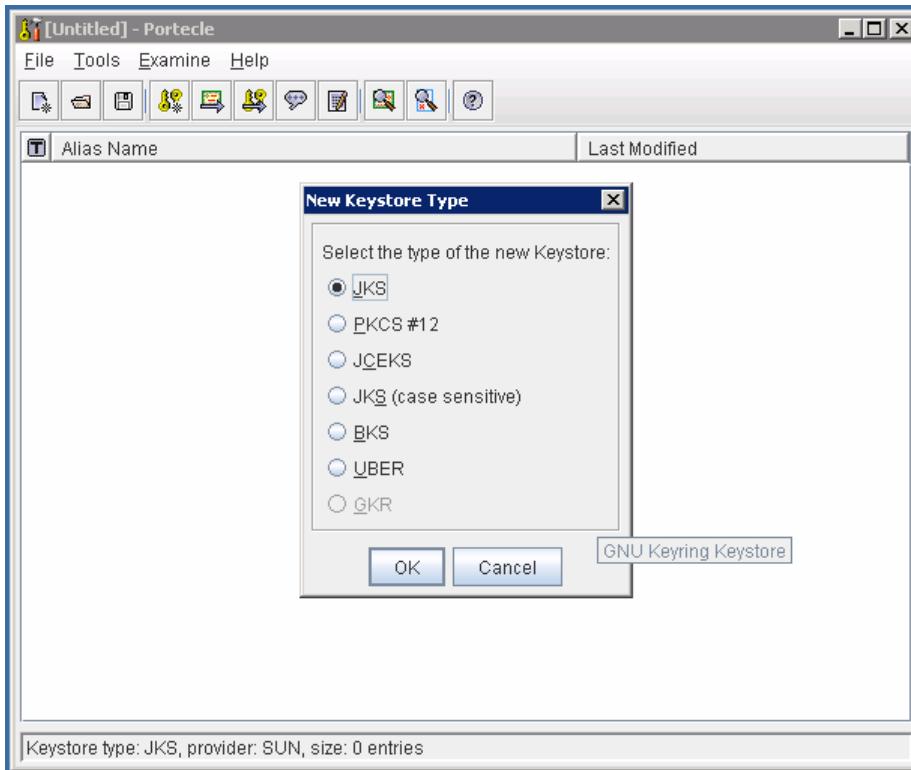
Reference web site links.

## Create a certificate for the JBOSS container

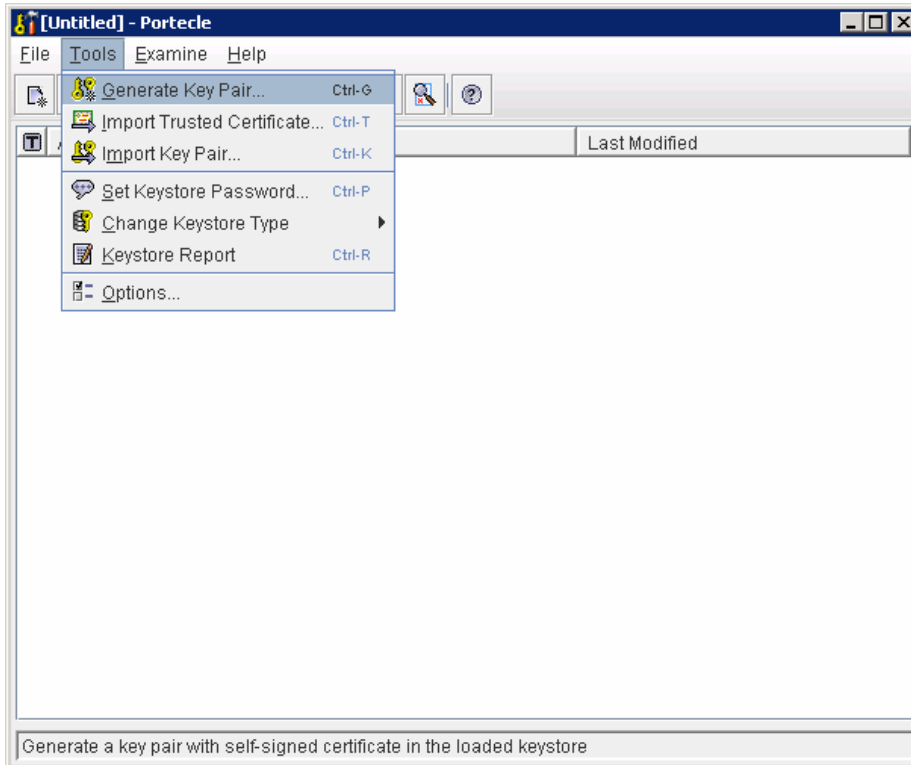
1. Select the new key store option from File Menu section to create the key store.



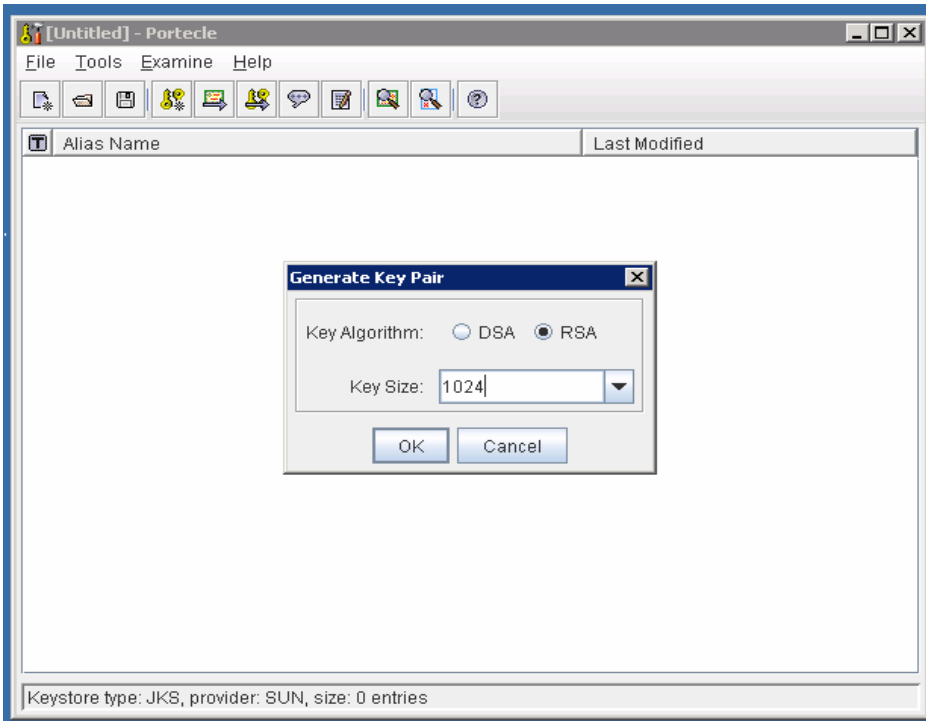
2. Select the JKS option and click the “OK” button.



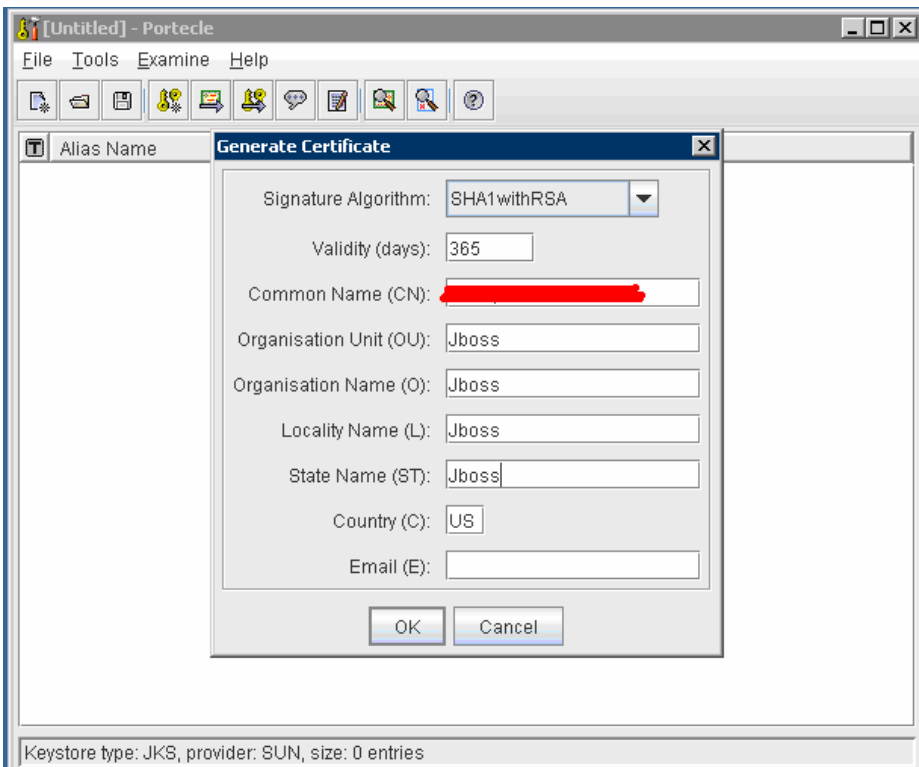
3. Select the Generate key pair from “Tools” Menu section.



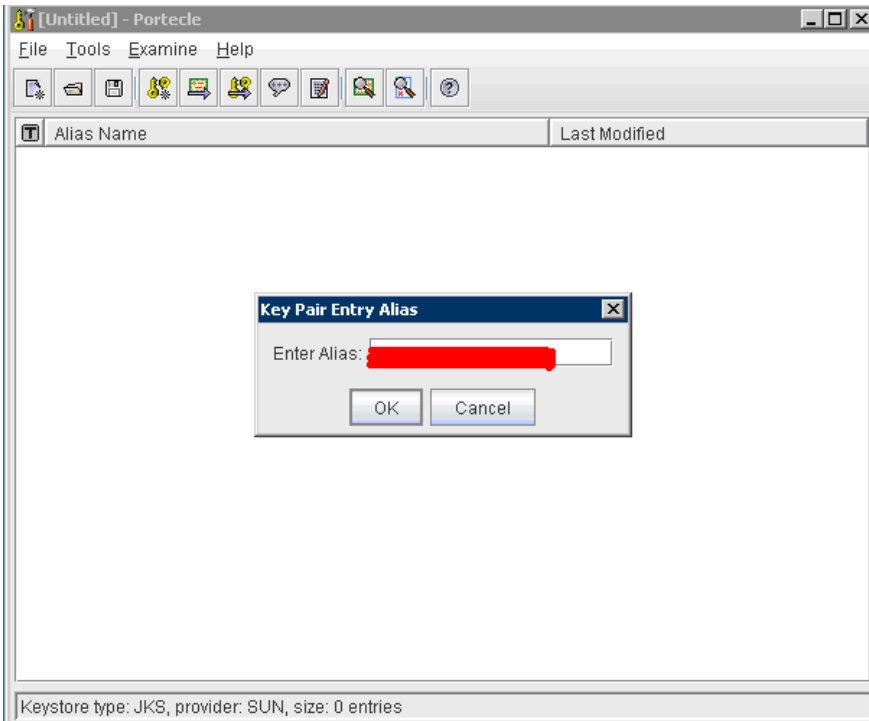
4. While generating the key value pair select the Key algorithm as **RSA** and Key size as **1024** and then click the “OK” button.



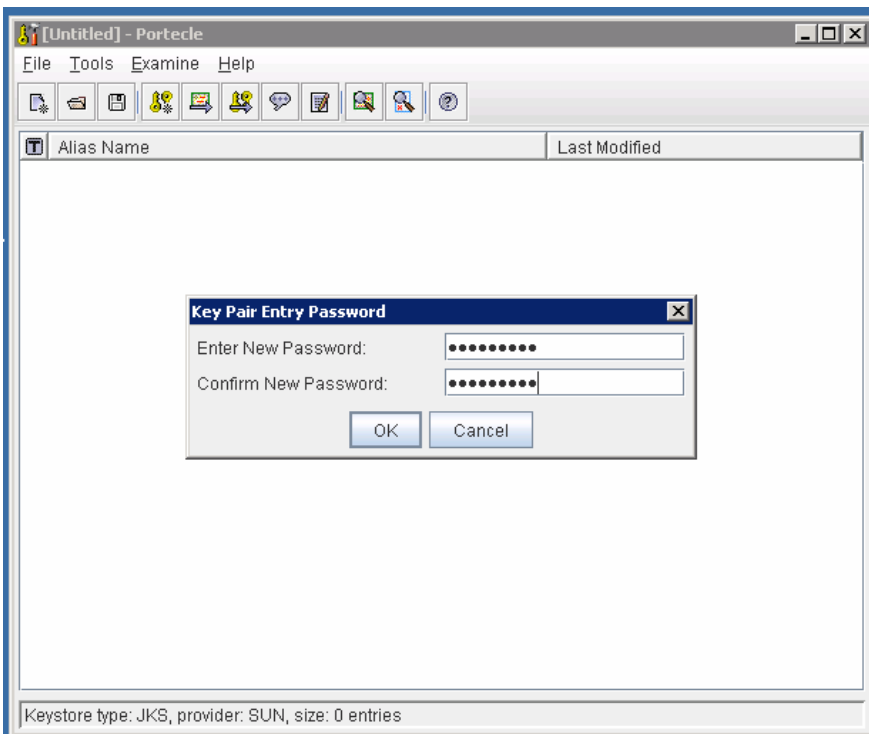
5. While creating the key value pair the Common name fully qualified domain name for example **sys09.jboss.com** and enter all the key values to click the “OK” button.



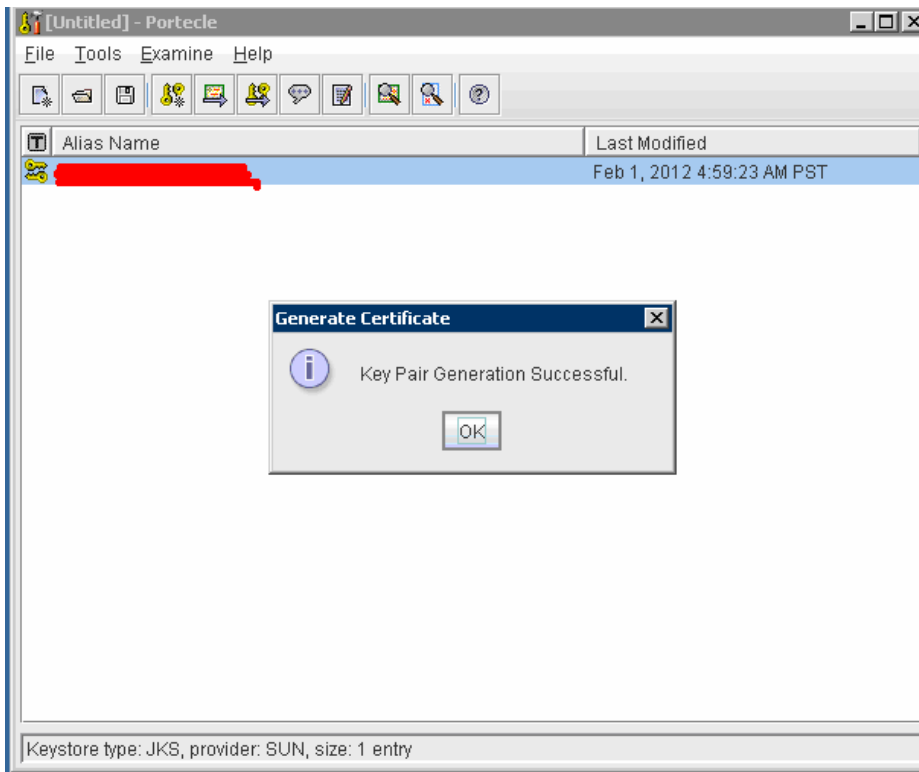
6. Give the key store alias name as **sys09.jboss.com** and to click the “OK” button.



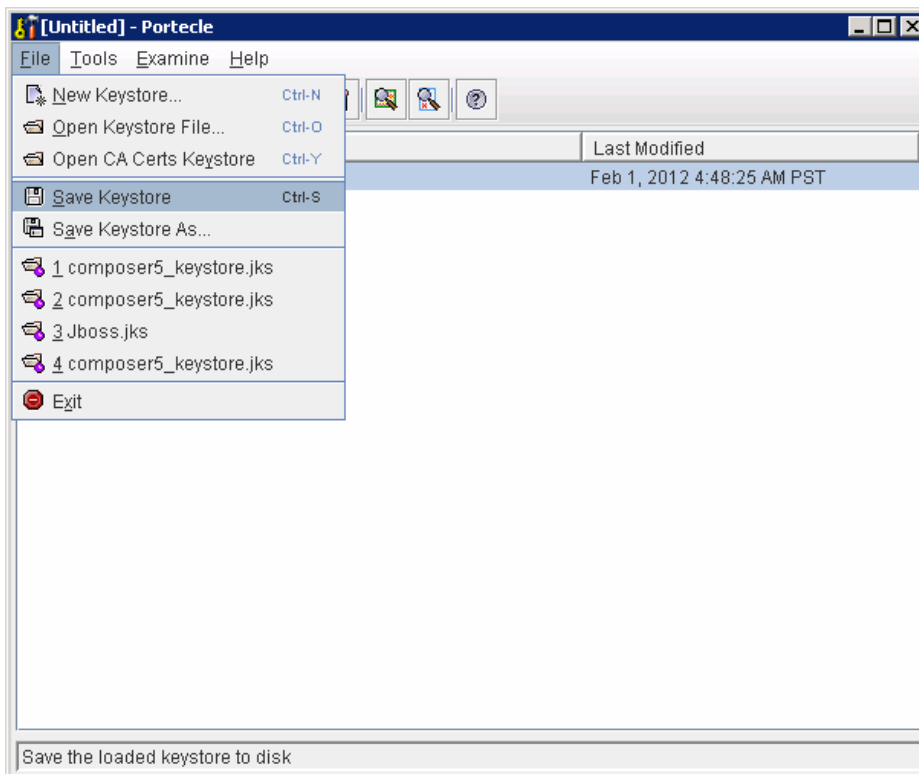
7. Give the key pair entry password (this needs to be used on the picketlink-ided.xml for the attribute “SigningKeyPass”) and then click the “OK” button.



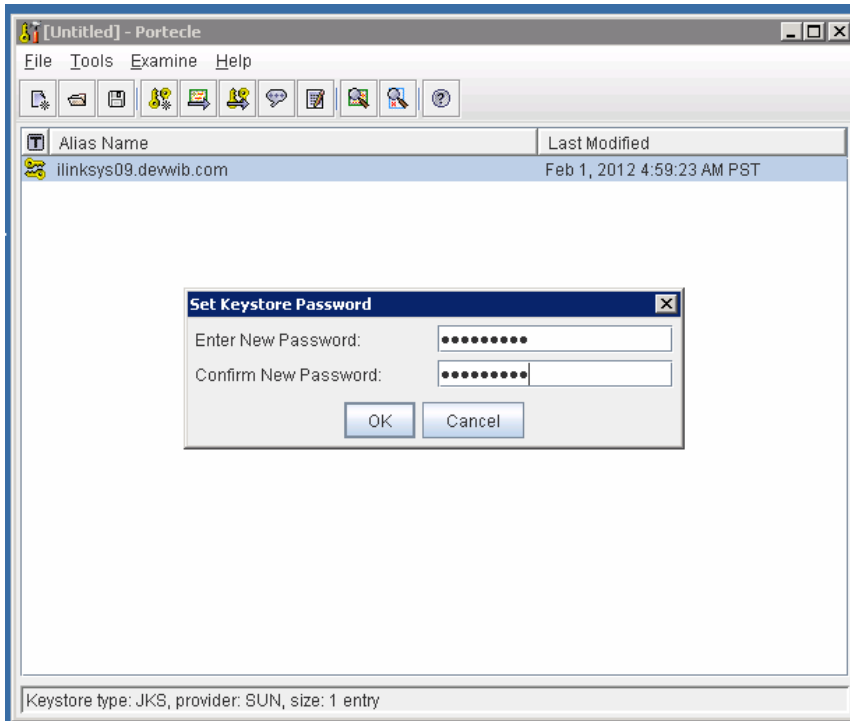
8. Generate the key value pair alias name as **sys09.jboss.com** and to click the “OK” button.



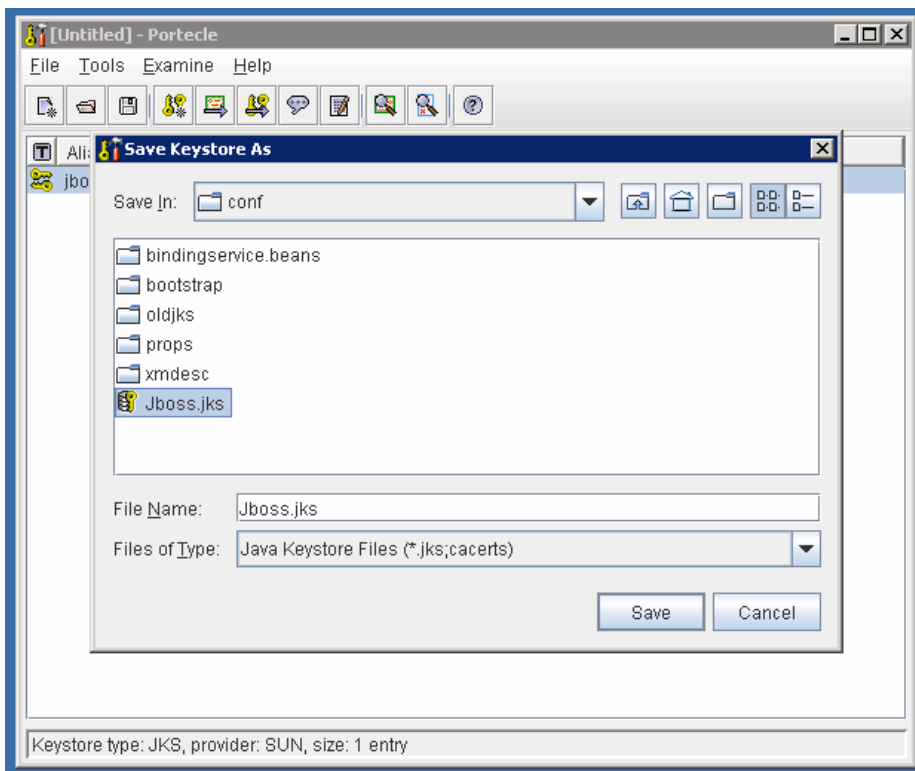
9. Select the Save key store option from File Menu section to save the key store file.



10. Give the key store password it can be used the picketlink-idfed.xml name “KeyStorePass” and to click the “OK” button.



11. To click the “Save” button to save the keystore with generated key value pairs.



## Configure the keystore on jboss server

Moved the newly created **jboss.jks** keystore file moved to <Jboss\_Home>/server/default/conf

Configure the jboss.jks key store file in the config file mentioned below

File Location : <Jboss\_Home>/ server/default/deploy/jbossweb.sar/server.xml

```
<Connector protocol="HTTP/1.1" SSLEnabled="true" port="8443"
address="{jboss.bind.address}"
```

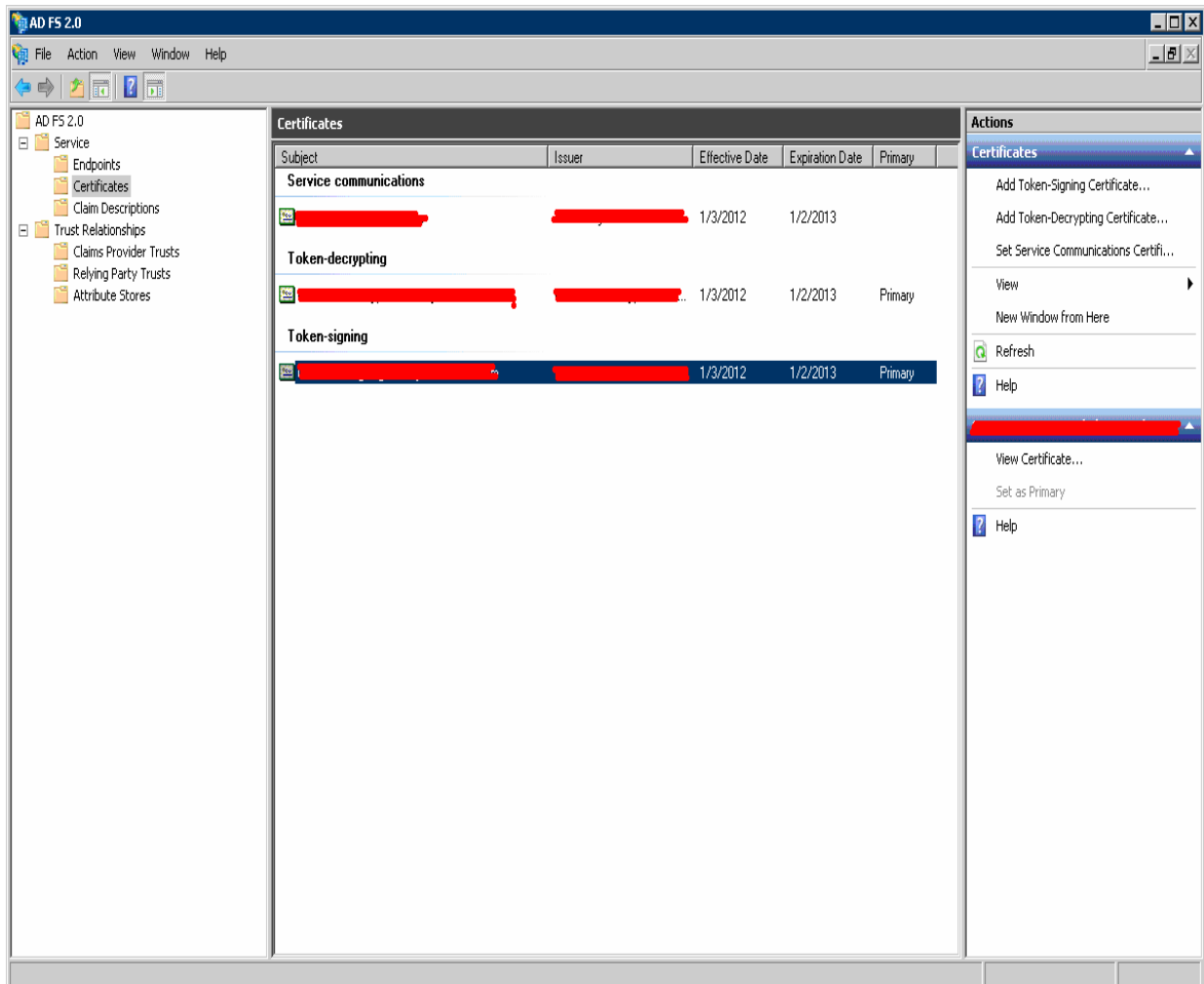
```
  scheme="https" secure="true" clientAuth="false"
```

```
  keystoreFile="{jboss.server.home.dir}/conf/jboss.jks" keystorePass="jbosspass"
```

```
  sslProtocol = "TLS" />
```

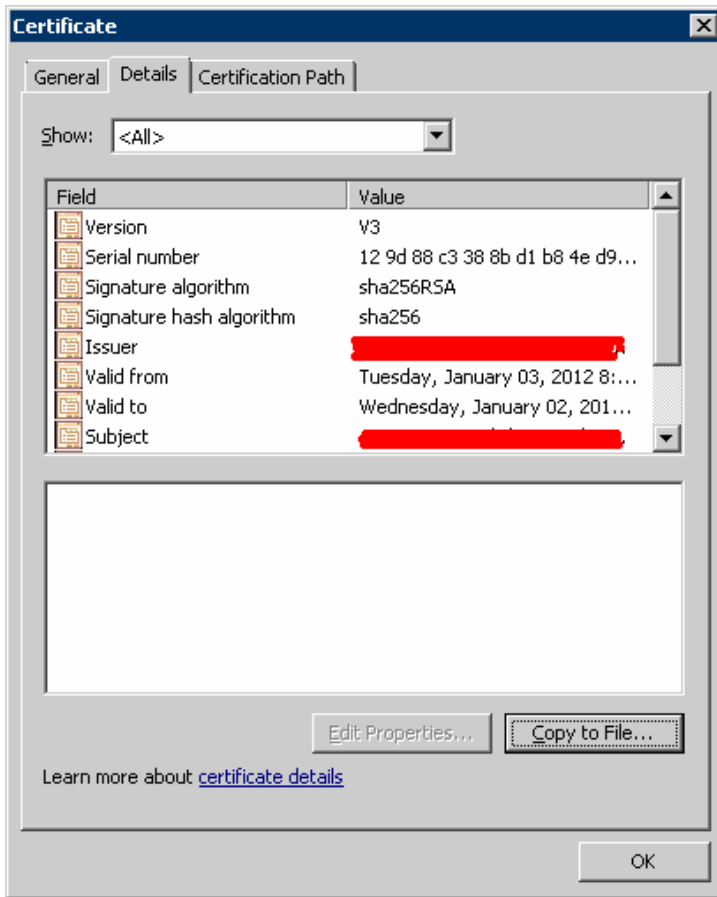
This completes securing the JBOSS container

## Export the ADFS Token signing certificate





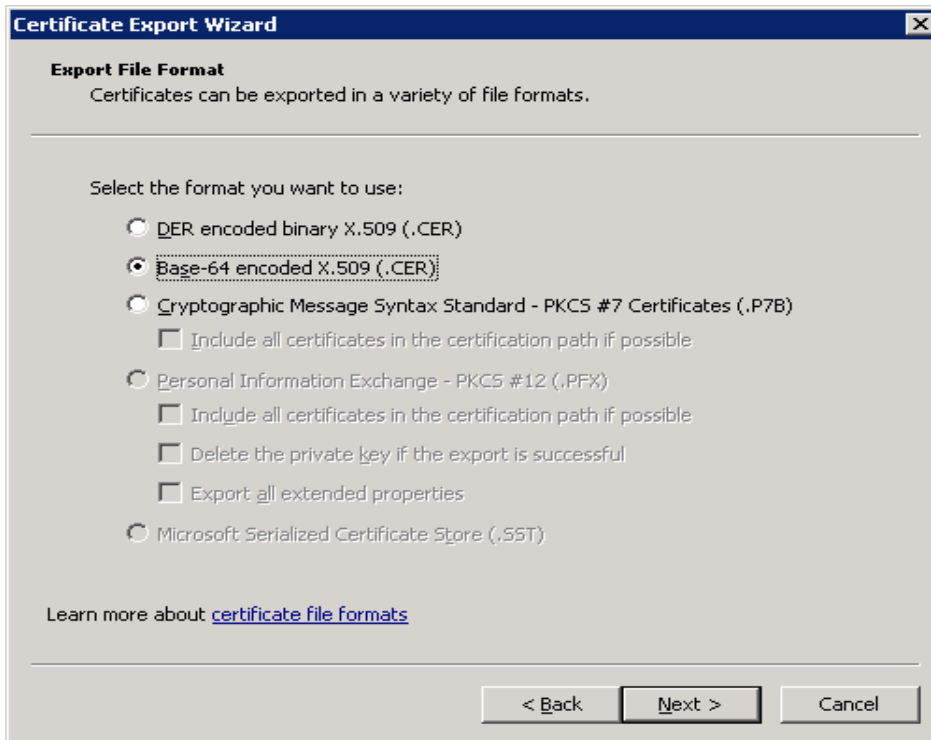
1. Right click the ADFS Token signing certificate and go to Details tab section.



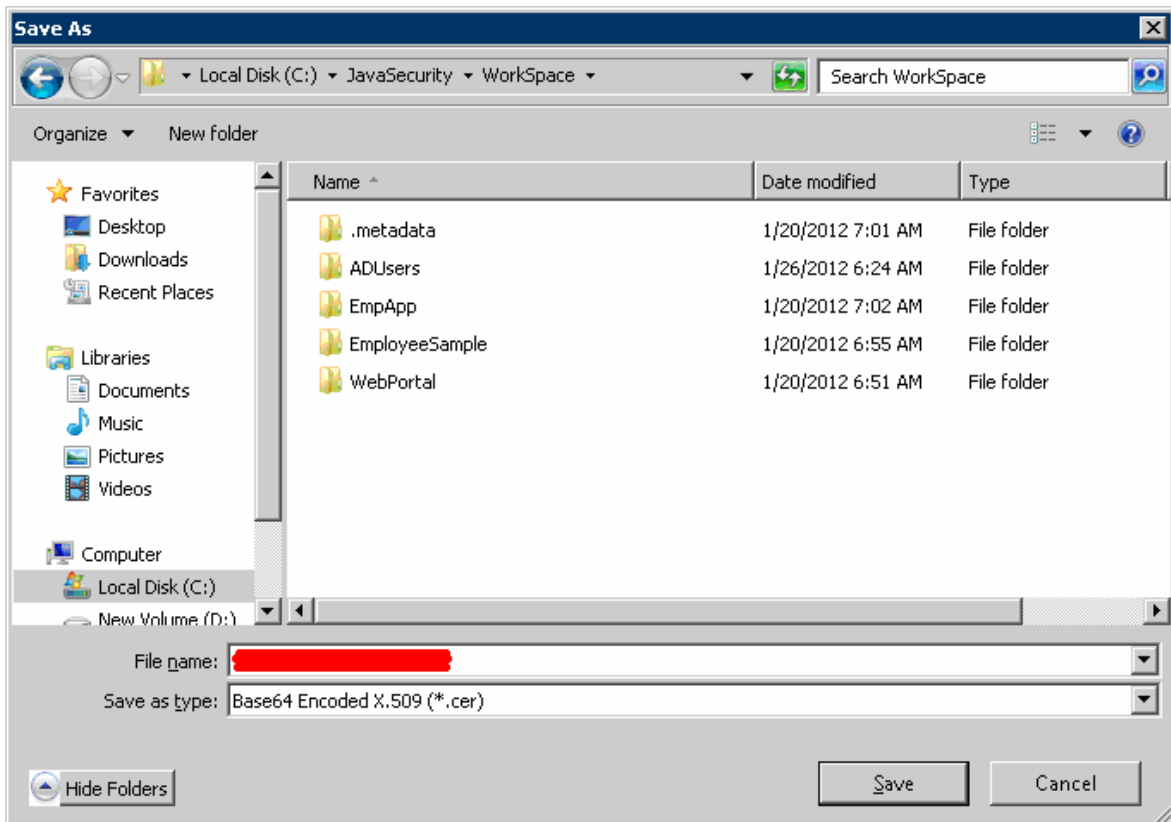
2. To click the "Copy to File" button to export the ADFS Token signing certificate.



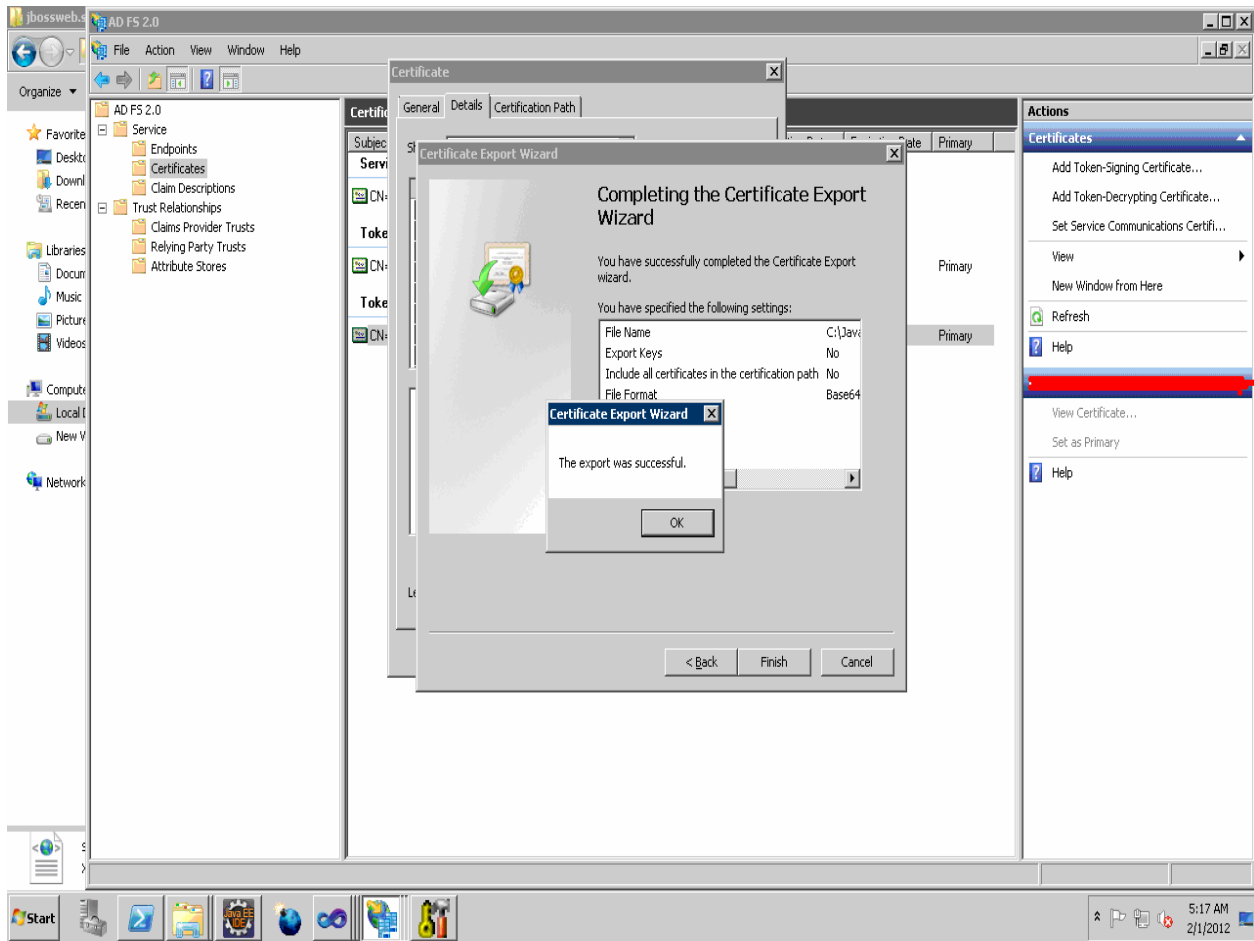
3. Select the Base-64 format option to export the certificate.



4. Save the certificate file name as **sys09.jboss.com.cer**.



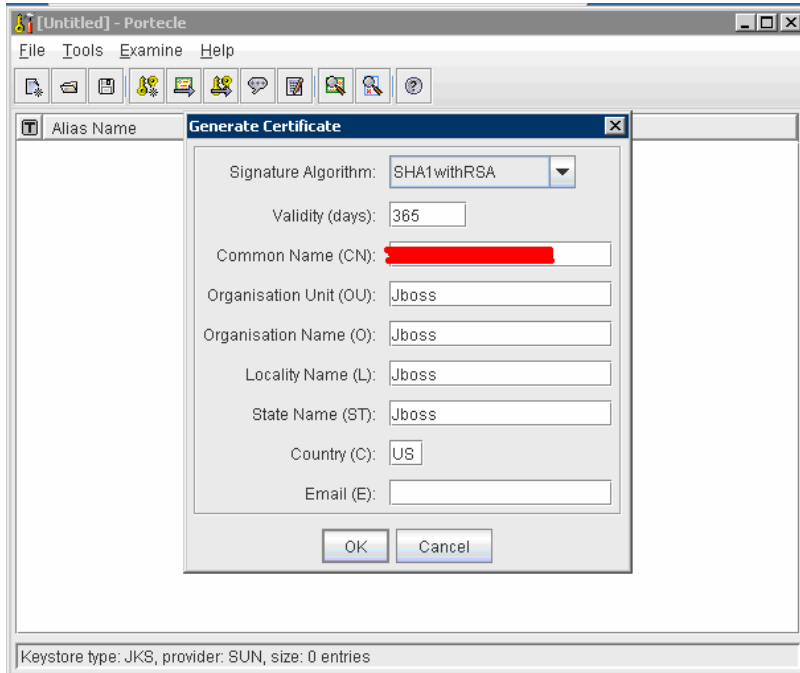
5. Export the ADFS Token signing certificate successfully from the ADFS section (ADFS2.0 -> Service -> Certificates)



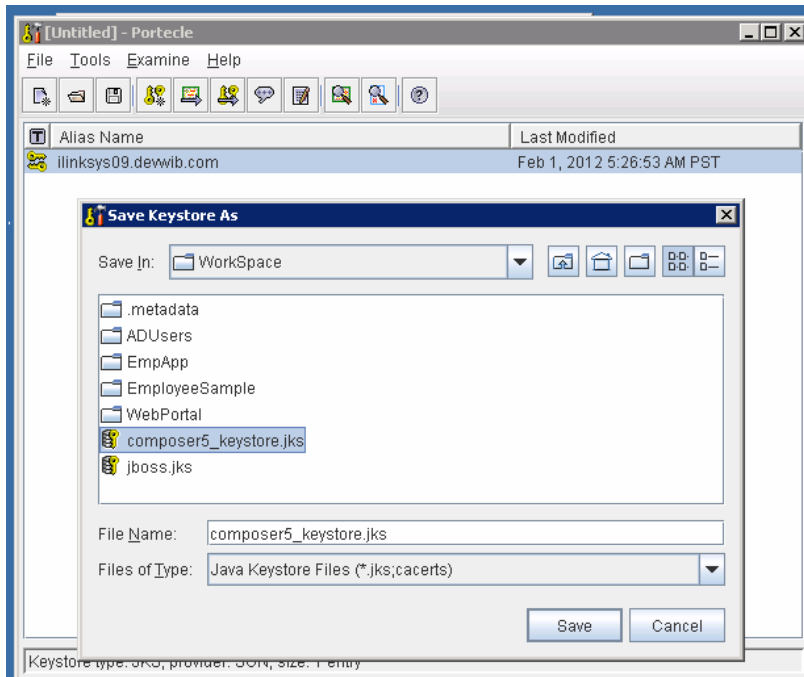
## Import ADFS Token signing certificate into a key store file

Follow steps from 1 thru 11 in the section (**Create a certificate for the JBOSS container**) create a certificate for the JBOSS container to create another key store file with the name **composer5\_keystore.jks**.

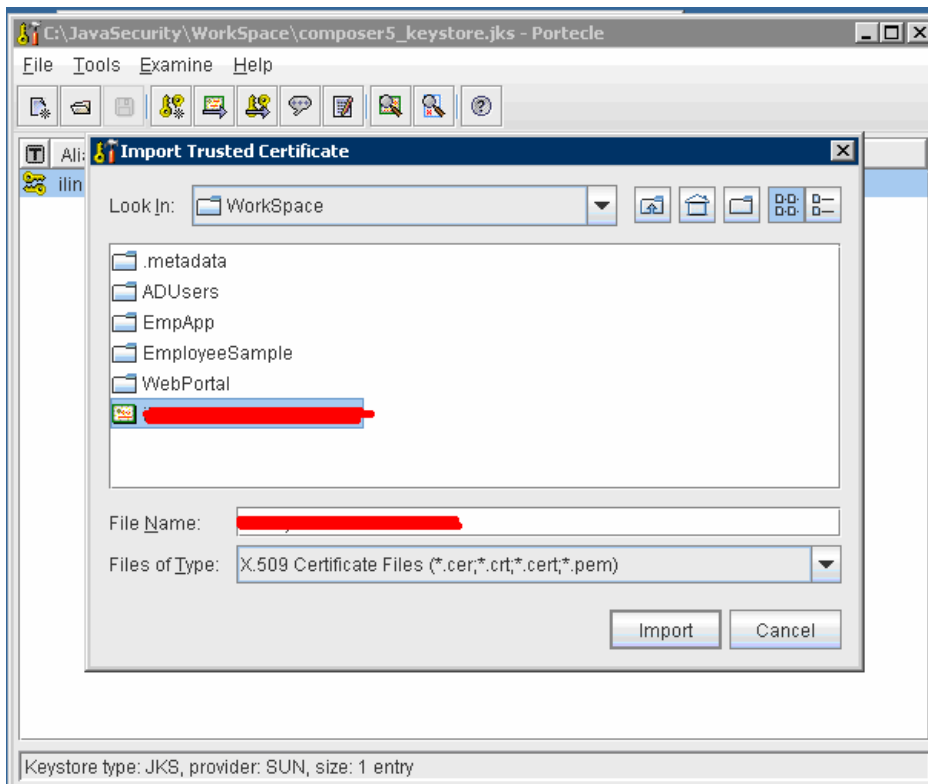
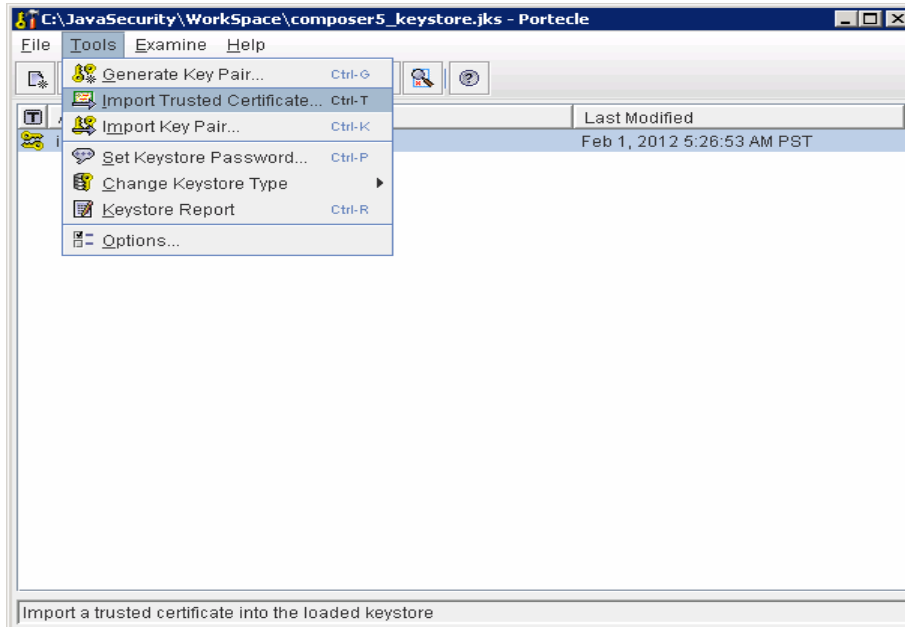
1. Given below are steps to generate key value



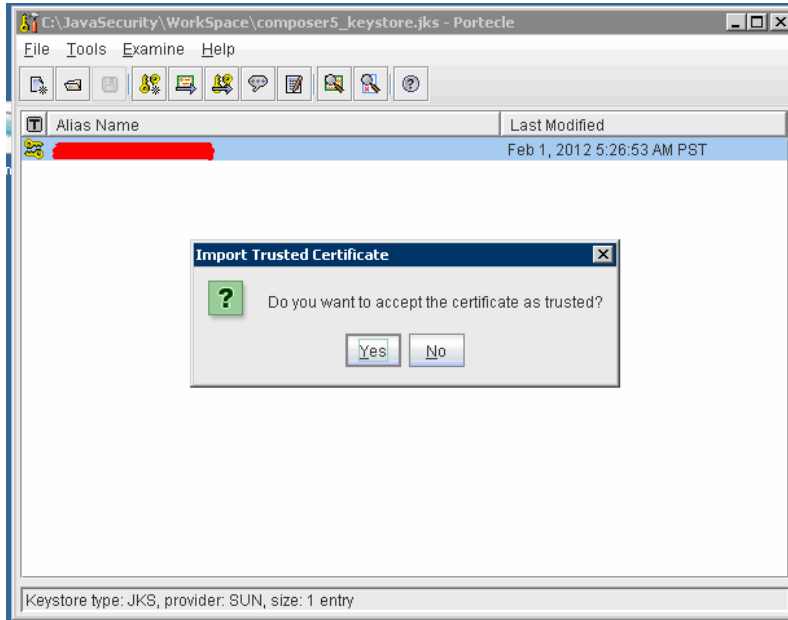
2. Save the key store file with the name composer5\_keystore.jks



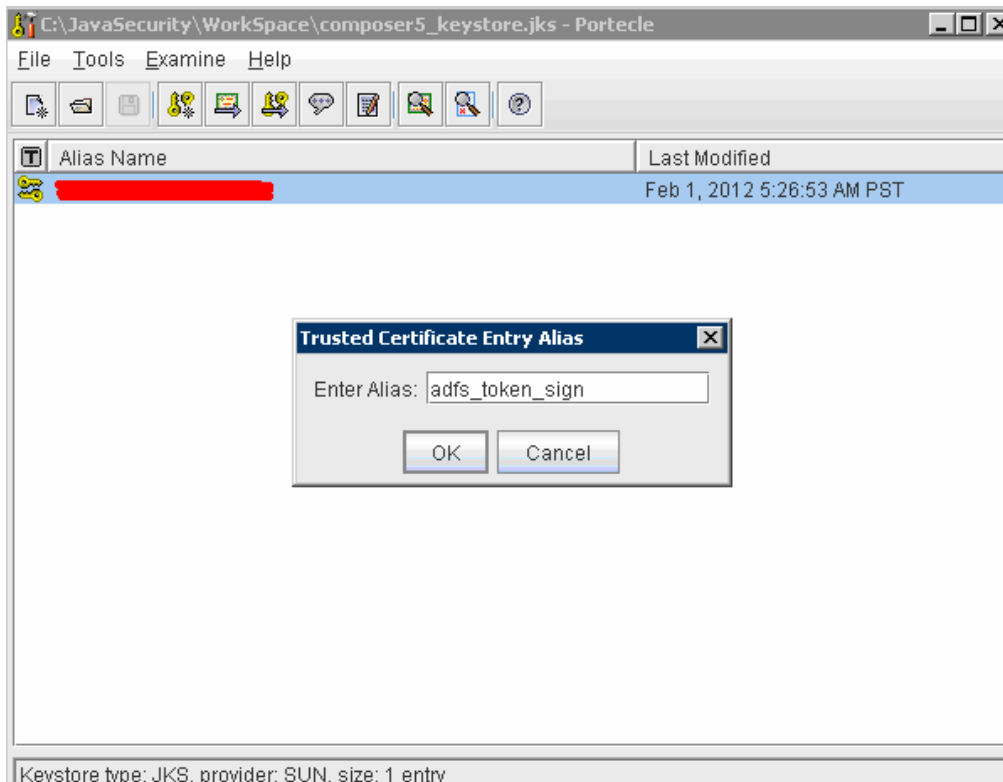
3. Import the ADFS token signing certificate **sys09.jboss.com.cer** on the key store file **composer\_keystore5.jks**. This needs to be used on the web application configuration file **picketlink-idfex.xml** under the tag `KeyProvider`



4. Click the “yes “button to import trusted ADFS Token signing certificate with key store (composer5\_keystore.jks.)

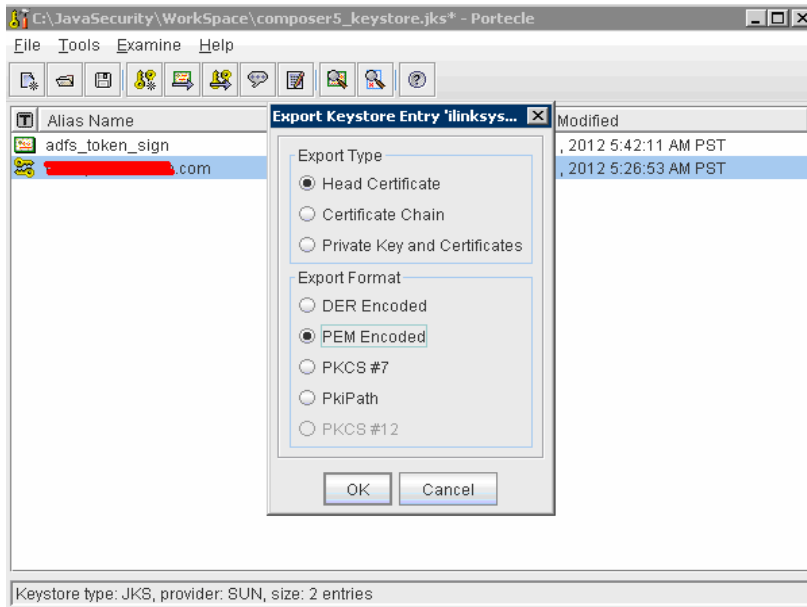


5. After importing the ADFS Token signing certificate into the composer5\_keystore.jks file give the certificate alias name as **adfs\_token\_sign**.

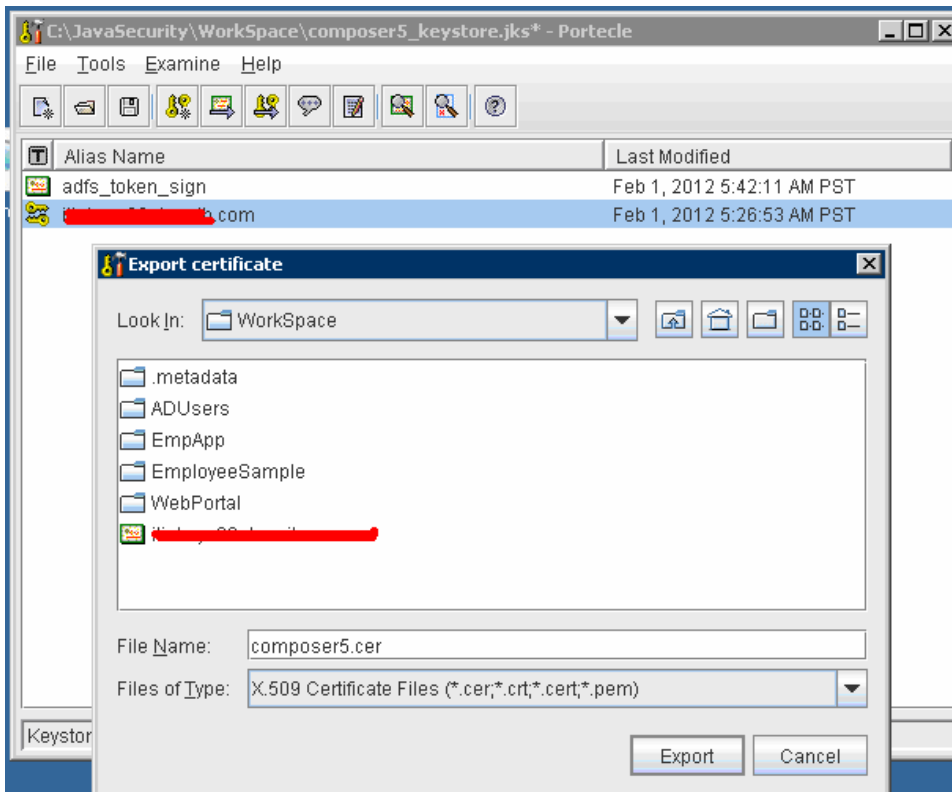


## Export the key store certificate file and then import it into the certificate store used by ADFS

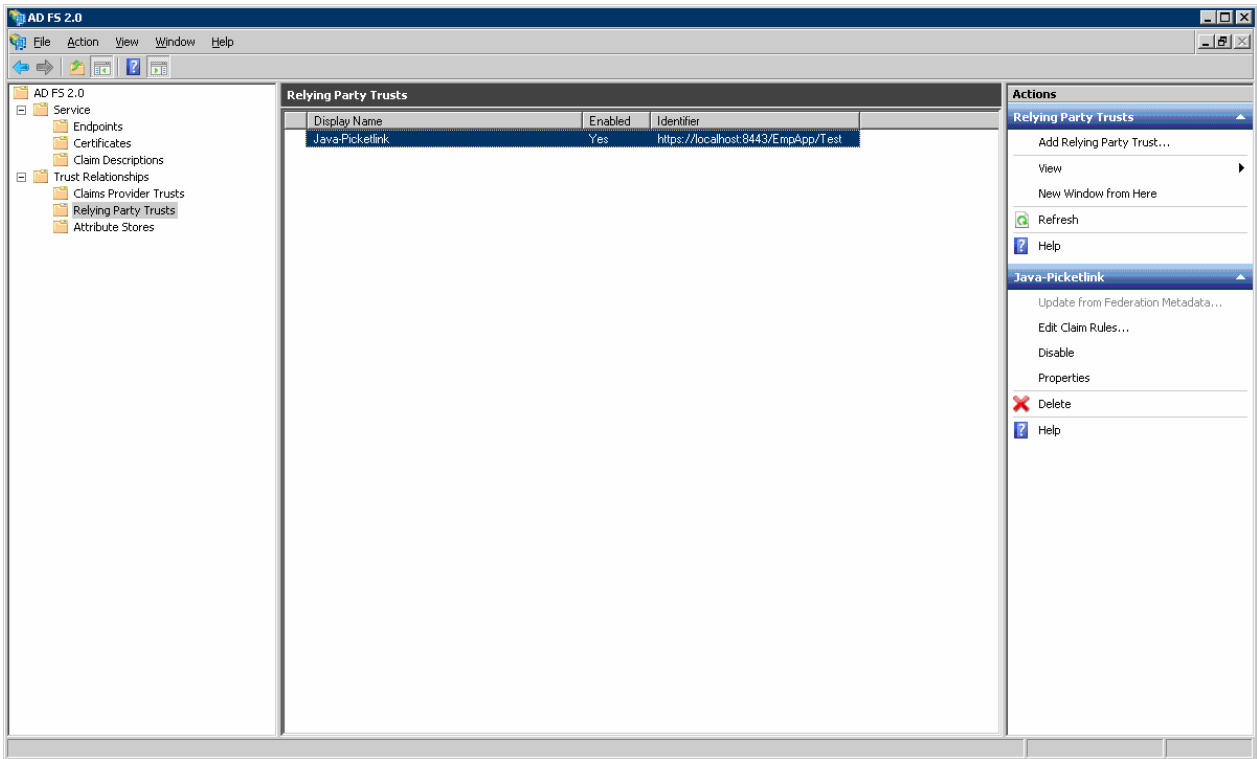
1. Select the generated key value pair name as **sys09.jboss.com** and right click and choose the export option and select the PEM Encoded options to.



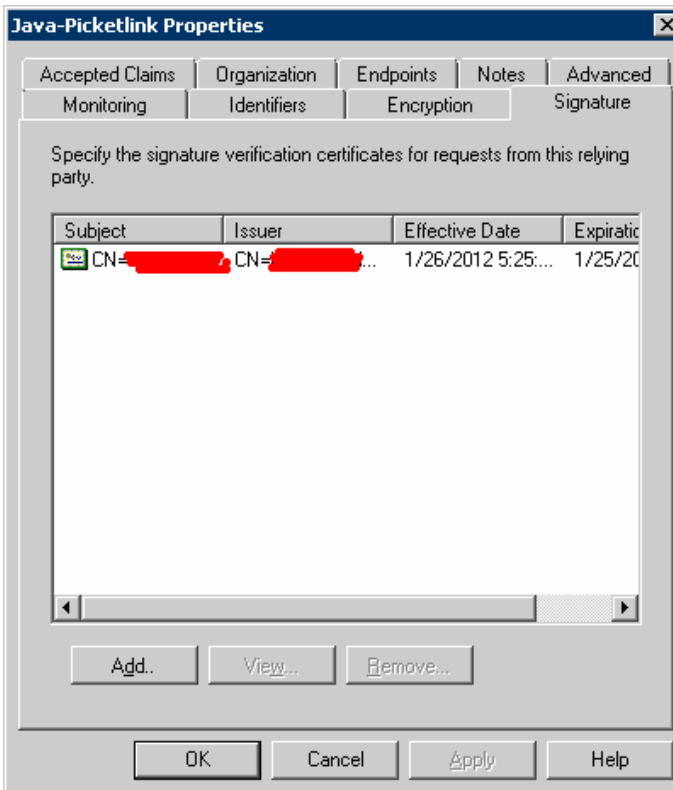
2. Create the certificate name as composer5.cer.



3. Create the ADFS relying party name as Java-Picketlink.



4. Add the composer5.cer certificate file on ADFS relying party name as Java –Picketlink





## Secure or Claims aware Web Application and picket link with the configuration files

See the Picketlink documentation on how to configure your web.xml, jboss-web.xml and picketlink-handler.xml, picketlink-idfex.xml, picketlink-sp-jboss-beans.xml and context.xml.

In the web application we have created the servlet name as Test and in the web.xml mentioned role name as **Enterprise Admins** is a group name only that group member can access the servlet method.

### Web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">
  <display-name>PicketLink Application</display-name>
  <description>
  Just a Test SP
  </description>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>PicketLink Application</web-resource-name>
      <url-pattern>/Test</url-pattern>
      <http-method>POST</http-method>
      <http-method>GET</http-method>
    </web-resource-collection>
    <auth-constraint>
      <role-name>Enterprise Admins</role-name>
    </auth-constraint>
  </security-constraint>
  <security-role>
    <description>
  </description>
    <role-name>Enterprise Admins</role-name>
  </security-role>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>Test</display-name>
    <servlet-name>Test</servlet-name>
    <servlet-class>com.sample.Test</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Test</servlet-name>
    <url-pattern>/Test</url-pattern>
  </servlet-mapping>
</web-app>
```

## Jboss-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
    <security-domain>sp</security-domain>
</jboss-web>
```

## Context.xml

```
<Context>
    <Valve
className="org.picketlink.identity.federation.bindings.tomcat.sp.SPPostFormAuthenticator" />
</Context>
```

## picketlink-sp-jboss-beans.xml

```
<deployment xmlns="urn:jboss:bean-deployer:2.0">
    <application-policy xmlns="urn:jboss:security-beans:1.0" name="sp">
        <authentication>
            <login-module code =
"org.picketlink.identity.federation.bindings.jboss.auth.SAML2LoginModule"
                flag = "required" />
        </authentication>
    </application-policy>
</deployment>
```

## picketlink-handlers.xml

```
<Handlers xmlns="urn:picketlink:identity-federation:handler:config:1.0">
    <Handler
class="org.picketlink.identity.federation.web.handlers.saml2.SAML2LogoutHandler" />
    <Handler
class="org.picketlink.identity.federation.web.handlers.saml2.SAML2IssuerTrustHandler" />
    <Handler
class="org.picketlink.identity.federation.web.handlers.saml2.SAML2AuthenticationHandler">
    </Handler>
</Handlers>
```

## picketlink-idfed.xml

```
<PicketLinkSP xmlns="urn:picketlink:identity-federation:config:1.0"
  ServerEnvironment="tomcat">

  <IdentityURL>https://sys09.jboss.com/adfs/ls/</IdentityURL>

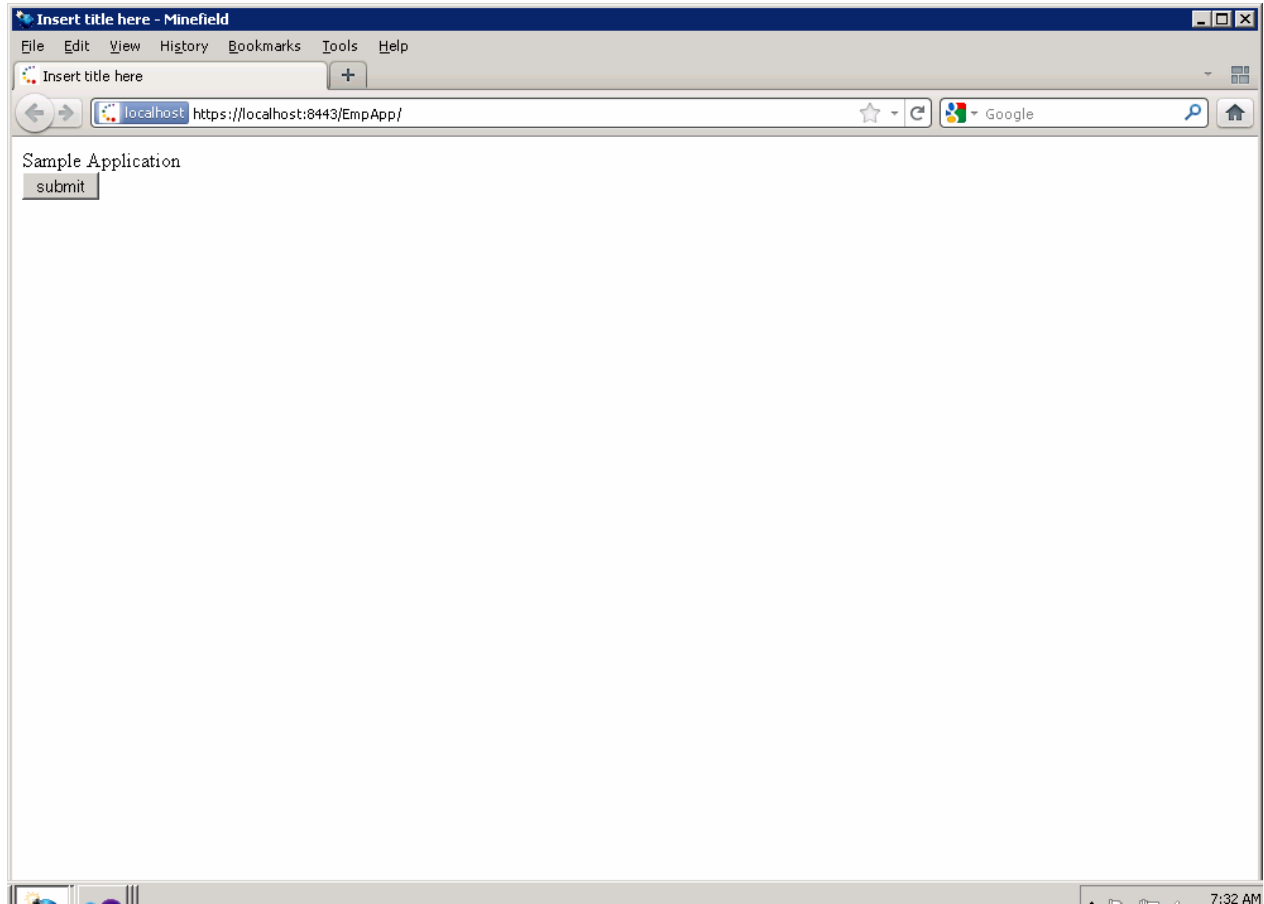
  <ServiceURL>https://localhost:8443/EmpApp/Test</ServiceURL>
  <!-- Only trust IDP SAML Responses from the following IDP domains -->
  <Trust>
    <Domains>localhost</Domains>
  </Trust>
  <KeyProvider
    ClassName="org.picketlink.identity.federation.core.impl.KeyStoreKeyManag
er">
    <!-- Path to keystore of certificates -->
    <Auth Key="KeyStoreURL"
Value="C:\JavaSecurity\composer5_keystore.jks" />
    <Auth Key="KeyStorePass" Value="jbosspass" />
    <!-- Which certificate in the keystore do we use ourself for
signing the
        SAML AuthnRequest to the IDP? -->

    <Auth Key="SigningKeyAlias" Value="sys09.jboss.com" />
    <Auth Key="SigningKeyPass" Value="jbosspass" />
    <!-- Every SAML Response from the IDP is/mustbe signed and the
signing
        must be checked to makeu use the IDP can be trusted
Key=Domain name for which
        this certificate can be used to check the signing
Value=Aliasname in keystore -->
    <ValidatingAlias Key="sys09.jboss.com" Value="adfs_token_sign" />
  </KeyProvider>

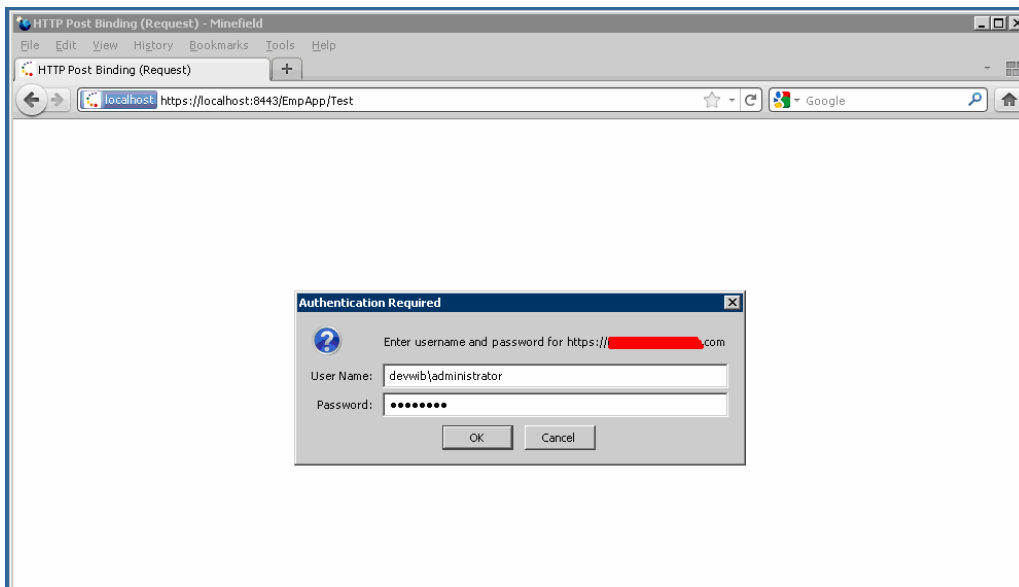
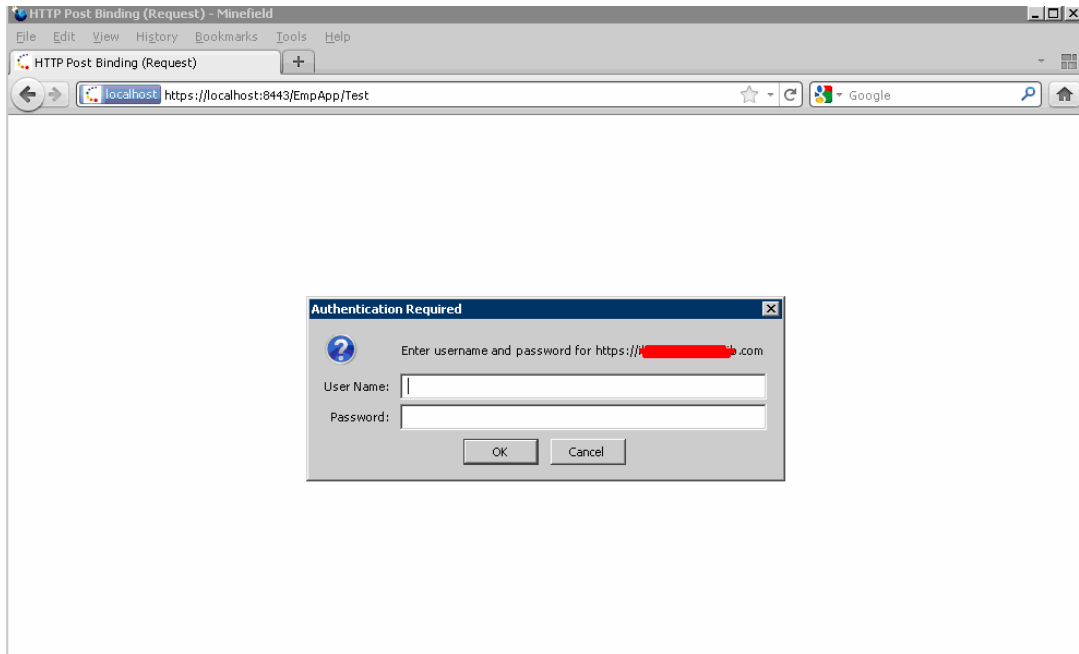
</PicketLinkSP>
```

## Web application Screen shots

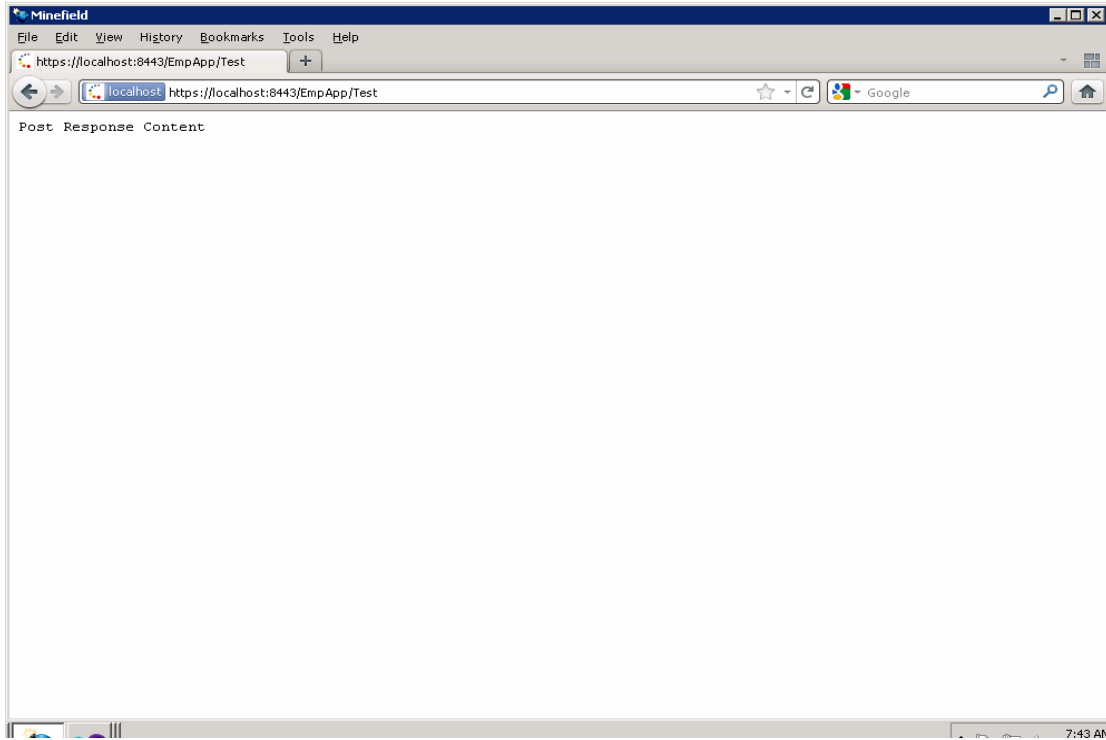
Web application access on the web browser.



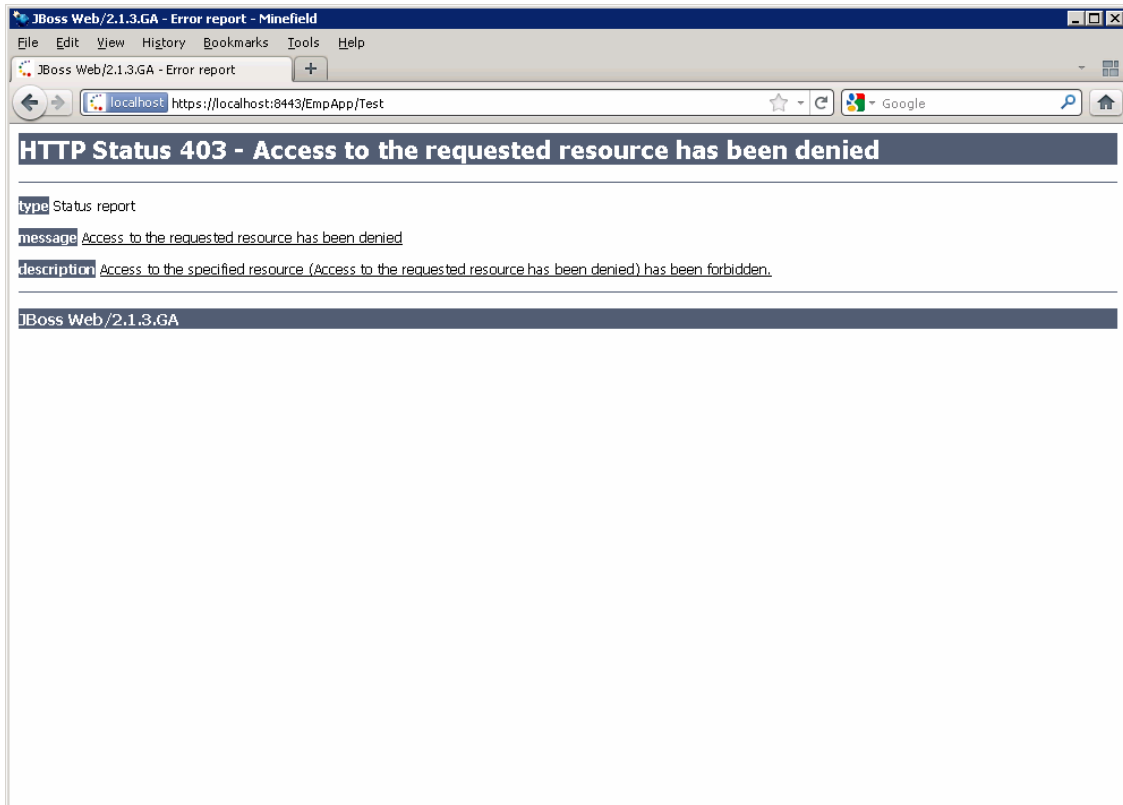
Click submit button it will redirect to ADFS login with the SAML request token which is created by Picketlink and give user credentials when prompted



After successfully login we are redirected to the screen below

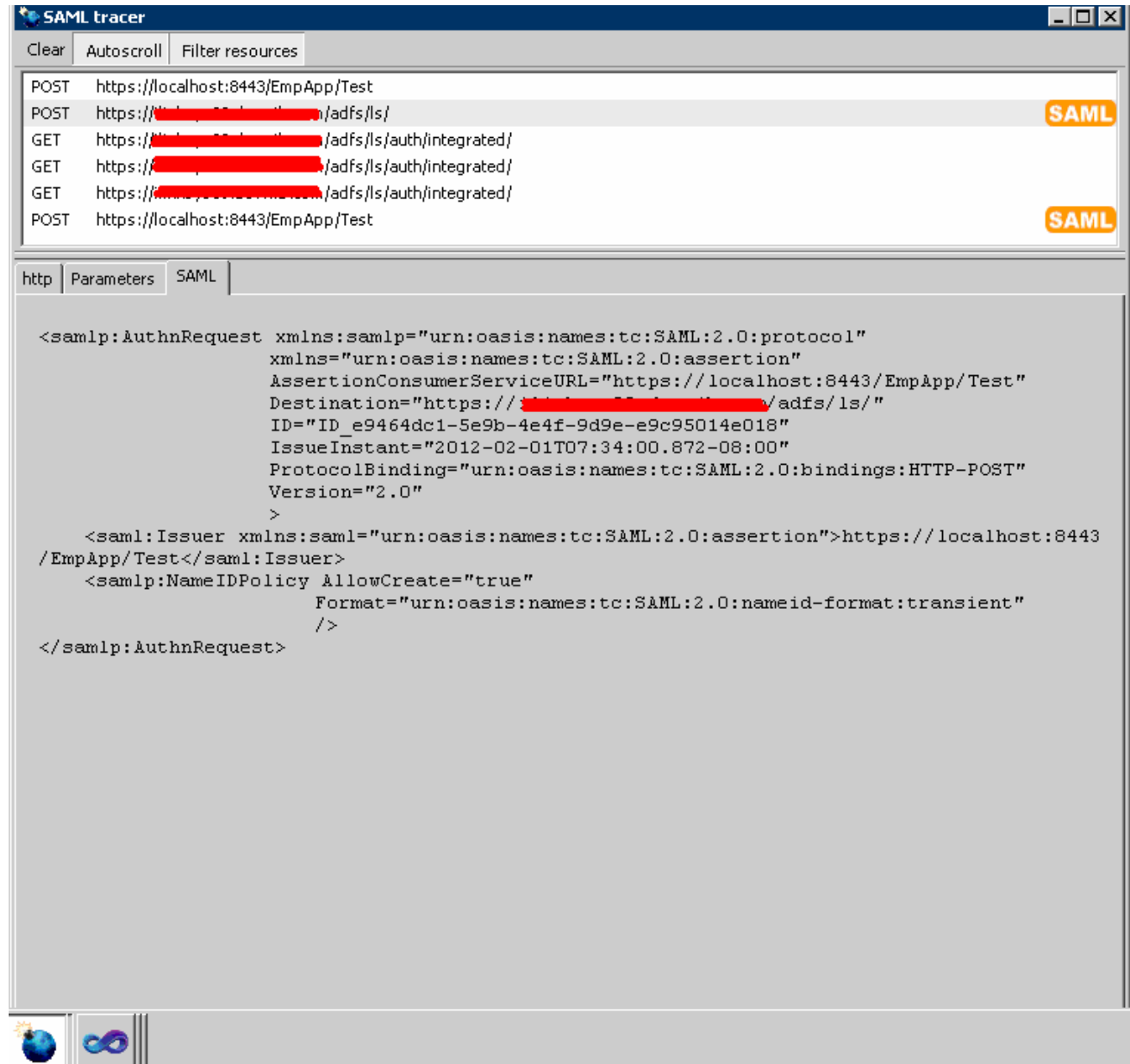


When user doesn't have access we get the error shown below.



## SAML Request Token

Using SAML Tracer plug-in on Firefox web browser to view the Request SAML Token which is created by picketlink and passing through the ADFS Identity provider (IP).



The screenshot shows the SAML Tracer application window. The top toolbar includes 'Clear', 'Autoscroll', and 'Filter resources'. The main list displays several HTTP requests:

- POST https://localhost:8443/EmpApp/Test
- POST https://[redacted]/adfs/ls/ (SAML)
- GET https://[redacted]/adfs/ls/auth/integrated/
- GET https://[redacted]/adfs/ls/auth/integrated/
- GET https://[redacted]/adfs/ls/auth/integrated/
- POST https://localhost:8443/EmpApp/Test (SAML)

The 'SAML' tab is selected, showing the following XML payload:

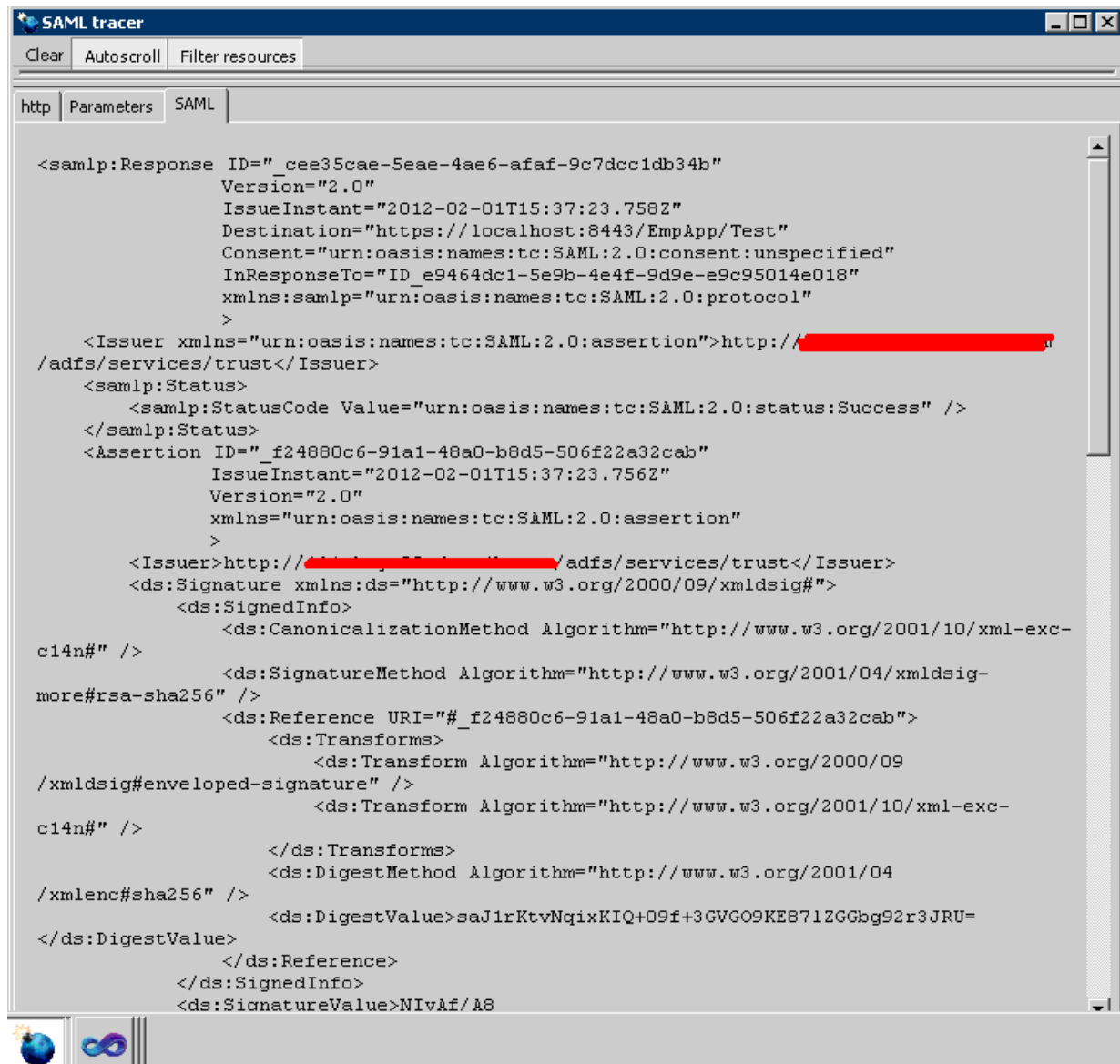
```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
  AssertionConsumerServiceURL="https://localhost:8443/EmpApp/Test"
  Destination="https://[redacted]/adfs/ls/"
  ID="ID_e9464dc1-5e9b-4e4f-9d9e-e9c95014e018"
  IssueInstant="2012-02-01T07:34:00.872-08:00"
  ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  Version="2.0"
  >
  <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://localhost:8443/
  /EmpApp/Test</saml:Issuer>
  <samlp:NameIDPolicy AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
  />
</samlp:AuthnRequest>
```

## SAML Response Token

In Firefox browser go to the menu section (Tools->Add-ons) search and include the Saml Tracer Plug-in and restart the Firefox browser or Install the plug-in using the link <https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/>.

Using SAML Tracer plug-in on Firefox web browser to view the Response SAML Token which is created by ADFS (IP) and picket link validate the user group on the web application after returns back to the web application

Below screen shots to show the SAML response token



```
<samlp:Response ID="_cee35cae-5eae-4ae6-afaf-9c7dcc1db34b"
  Version="2.0"
  IssueInstant="2012-02-01T15:37:23.758Z"
  Destination="https://localhost:8443/EmpApp/Test"
  Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified"
  InResponseTo="ID_e9464dc1-5e9b-4e4f-9d9e-e9c95014e018"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
>
  <Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">http://[redacted]
/adfs/services/trust</Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
  </samlp:Status>
  <Assertion ID="_f24880c6-91a1-48a0-b8d5-506f22a32cab"
    IssueInstant="2012-02-01T15:37:23.756Z"
    Version="2.0"
    xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
  >
    <Issuer>http://[redacted]/adfs/services/trust</Issuer>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha256" />
        <ds:Reference URI="#_f24880c6-91a1-48a0-b8d5-506f22a32cab">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09
/xmlsig#enveloped-signature" />
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2001/04
/xmlenc#sha256" />
          <ds:DigestValue>saJ1rKtvNqixKIQ+09f+3GVGO9KE871ZGGbg92r3JRU=
</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>N1vAf/A8
```



```

SAML tracer
Clear Autoscroll Filter resources

http Parameters SAML

<ds:SignatureValue>NiyAf/A8
/GUM7meS2YcqmcI8KXGlaVuItrHVrxIrlldxlzmg4kpiIPTwOIQeTFYul6+XzdwOVgjQLNJNT1jXFJErfUTDmES
mCUME085dQUjifNXT
/fv5bhduvS94Q+oT+7asqe3hm008EnwZWwrEpmGu9e3wBhriKI7Lm589wn2wUVmP6uqVTqKHH5HgMt21hK
/XvSEGjCxonCOKxAEMduVxm/1sXdVfK6+DEbTs2dVfUtr4r8w6mXi+5UX1K4Q/O385nWYw+CZznYlnXGx1
/gYbnUxkZTgzag3nt8UU3sryze1aDfAA3YerNlMg/ZdS9qVlO1UuB1FuHV4RoZqeQ==</ds:SignatureValue>
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
  <ds:X509Data>
    <ds:X509Certificate>MIIC5jCCAc6gAwIBAgIQEp2IwziL0bh02fQaBNZAgjANBgkqhkiG9w0BAQsFADAvMS0
wKwYDVQQDEyRBREZTIFNpZ25pbmcgLSBpbGlua3N5czA5LmRldndpYi5jb20wHhcNMTIwMTAzMTYwOTQxWhcNMT
MwMTAyMTYwOTQxWjAvMS0wKwYDVQQDEyRBREZTIFNpZ25pbmcgLSBpbGlua3N5czA5LmRldndpYi5jb20wggEiM
AOGCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCp6E0WUUEgiiQrA8EcnedJOSUC7S3M8YzRMfLeZgFssZOHkqd9
yCYgYv+sXHznKmpBOS1DCeUS7blvThlwhlwjOwv3RQN24GkTIdoJwmxmwIveLiCSTjY1jCJdFLMbcNTvaBXR6q
RnGnDsqQB3II+vDlP72W6eumEKM6bC//3k7sOxK8c3zjfbAiTq8OVutCH+3y8yfGqzyhwNzqUj6
/rwWs7t2PvdMv6cLWwXDwnSkiY4h09aL8kiX
/d6HbBm3ZdIN2S2Q1FEUID6+YfdAX1N3Gy1VpzZLdBvhlciiWUTujrgrrZjqCa2Syus2TpxjSqiKGaS4CjnF3eS
SO6rUnAgMBAAEwDQYJKoZIhvcNAQELBQADggEBACx6Z3+NjUpM
/4Heu9+HHNViCr3SA5gHD0cwcMflSOonngLHXvN4gPNyqsPPUqOIaeVwmcQWFFV3DxnVARDRI1K33eqBos6zrS+
tRAU1TOYdcRtFwQPPcPmMBDRbuNGwI/SJotyhpCqFjNxiAlnHooCc5N
/4QRKun8Ukx9+B4KdHXyGd4NGXyZ+Lmg40dZLTZGuia2su
/vHhVEHG10sE4YHitW6XWfKi9+cVM3KeMq1m3K3TiH6EENEI3ygnFhsQXZcxiy2tjC6zWS1+yjZ8DIP8GYGTy3i
AwgPVnLskjhFH1PMMLSt33diTKEum4KvVeVqRuJhPE1huahbGJhGOxY=</ds:X509Certificate>
  </ds:X509Data>
</KeyInfo>
</ds:Signature>
<Subject>
  <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:transient">2KfYQP1JSmVicx50ytqCk1Ryut3aUUtpby/w7UBb1Ek=</NameID>
  <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <SubjectConfirmationData InResponseTo="ID_e9464dc1-5e9b-4e4f-9d9e-
e9c95014e018"
    NotOnOrAfter="2012-02-01T15:42:23.758Z"
    Recipient="https://localhost:8443/EmpApp/Test"
    />
  </SubjectConfirmation>
</Subject>
<Conditions NotBefore="2012-02-01T15:37:23.736Z"
NotOnOrAfter="2012-02-01T16:37:23.736Z"

```

SAML tracer


Clear Autoscroll Filter resources

http Parameters SAML

```

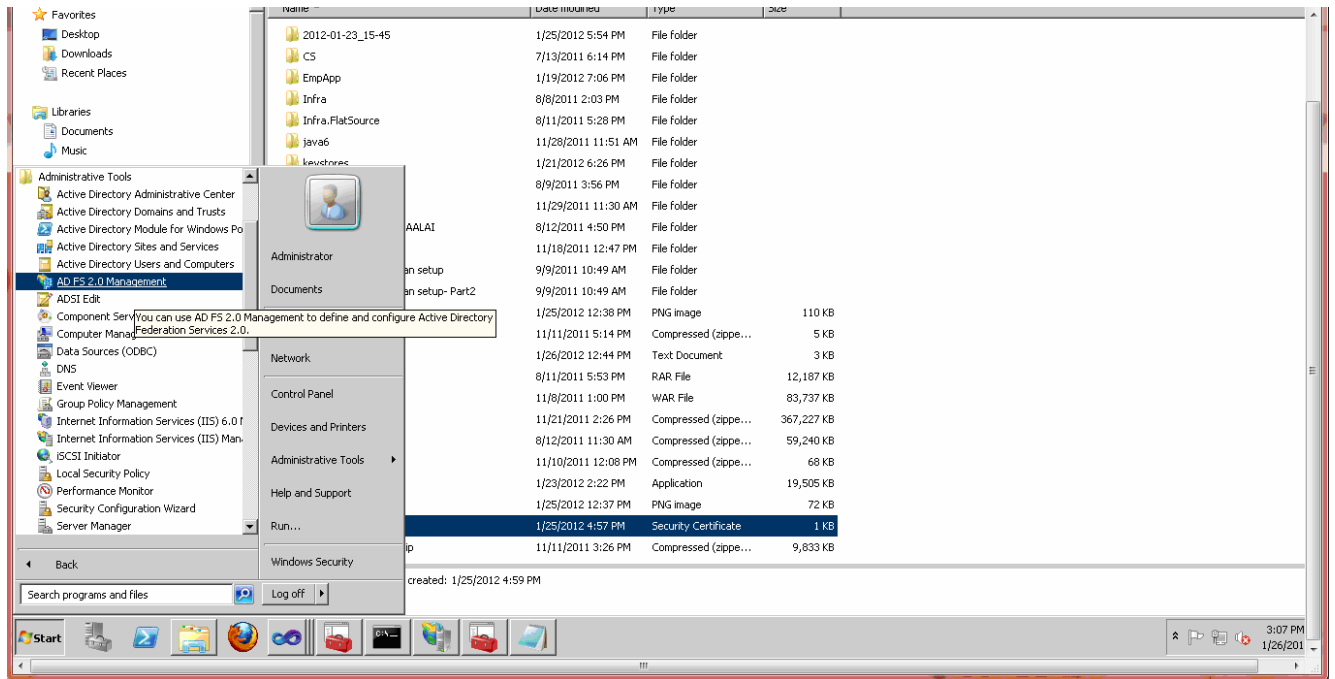
NotOnOrAfter="2012-02-01T15:42:23.758Z"
Recipient="https://localhost:8443/EmpApp/Test"
/>
</SubjectConfirmation>
</Subject>
<Conditions NotBefore="2012-02-01T15:37:23.736Z"
  NotOnOrAfter="2012-02-01T16:37:23.736Z"
  >
  <AudienceRestriction>
    <Audience>https://localhost:8443/EmpApp/Test</Audience>
  </AudienceRestriction>
</Conditions>
<AttributeStatement>
  <Attribute Name="http://schemas.microsoft.com/ws/2008/06/identity/claims
/role">
    <AttributeValue>Domain Admins</AttributeValue>
    <AttributeValue>Domain Users</AttributeValue>
    <AttributeValue>Group Policy Creator Owners</AttributeValue>
    <AttributeValue>WSS_WPG</AttributeValue>
    <AttributeValue>WSS_ADMIN_WPG</AttributeValue>
    <AttributeValue>Schema Admins</AttributeValue>
    <AttributeValue>WSS_RESTRICTED_WPG_V4</AttributeValue>
    <AttributeValue>Enterprise Admins</AttributeValue>
  </Attribute>
  <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims
/surname">
    <AttributeValue>Administrator@devwib.com</AttributeValue>
  </Attribute>
</AttributeStatement>
<AuthnStatement AuthnInstant="2012-02-01T15:37:23.563Z"
  SessionIndex="_f24880c6-91a1-48a0-b8d5-506f22a32cab"
  >
  <AuthnContext>
    <AuthnContextClassRef>urn:federation:authentication:windows</AuthnContextClassRef>
  </AuthnContext>
</AuthnStatement>
</Assertion>
</samlp:Response>

```

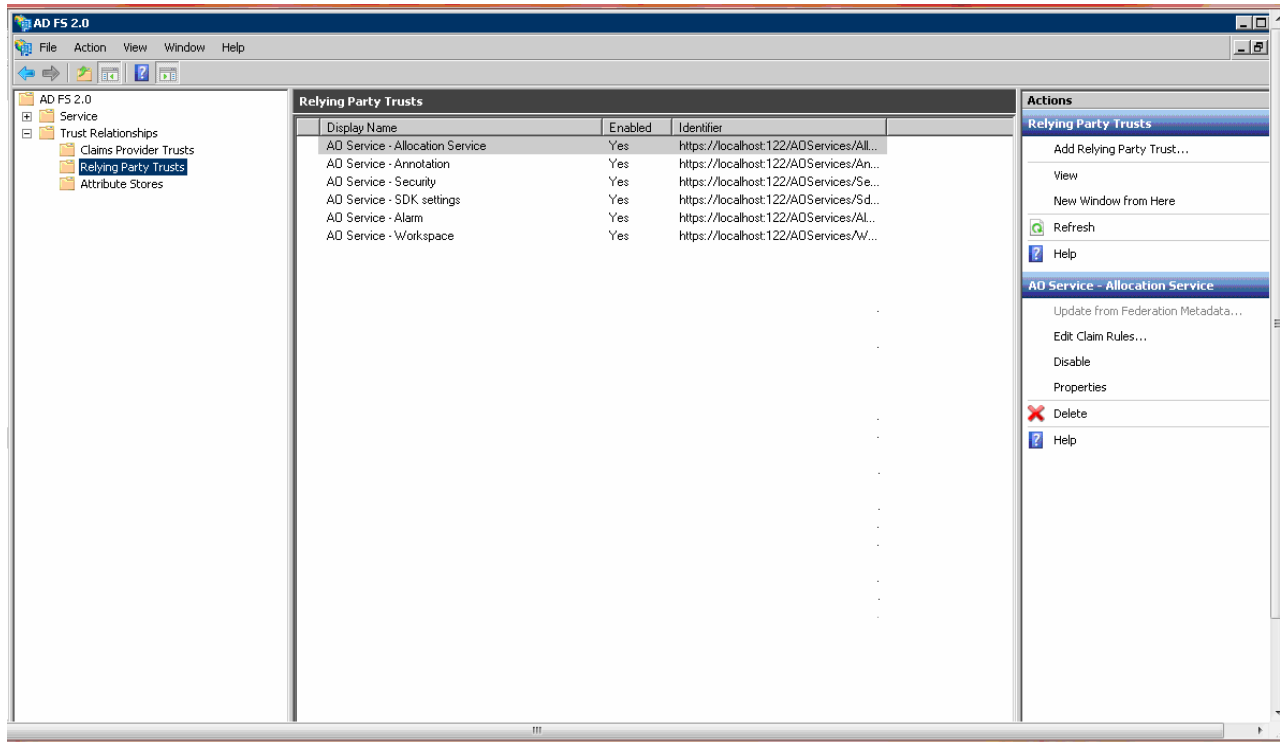


# Add relying party in ADFS 2.0

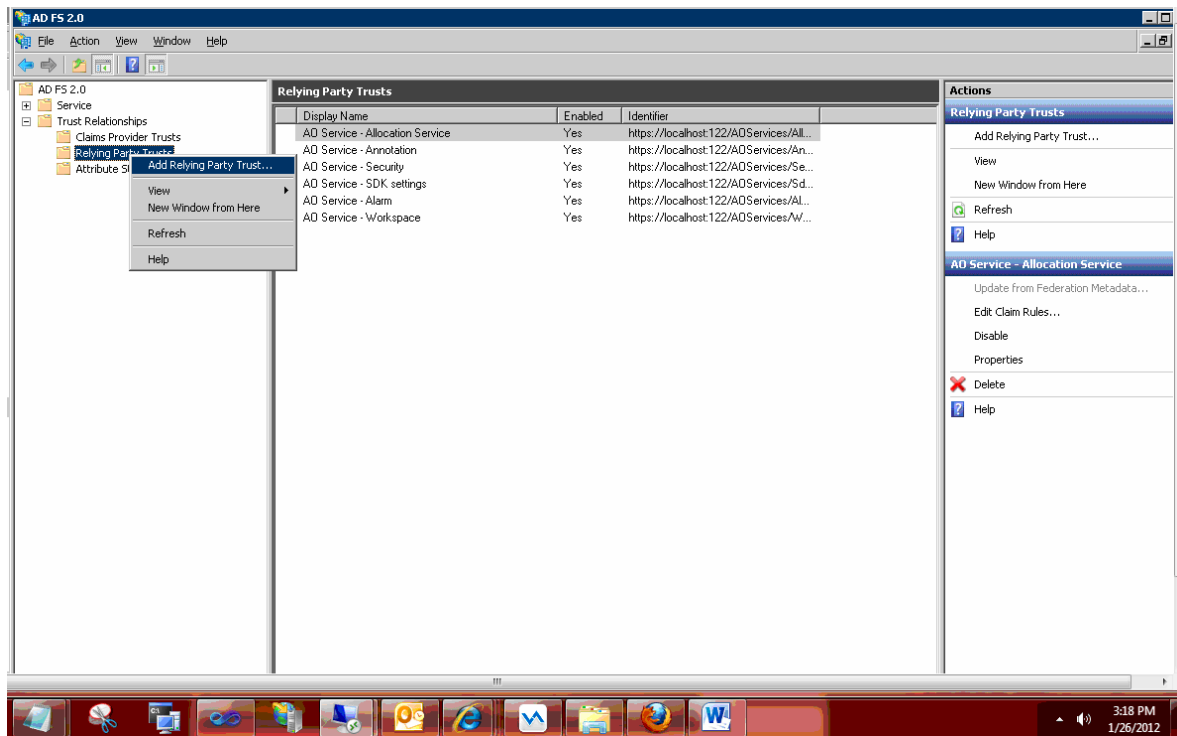
1. Open ADFS 2.0 from Start → Programs → Administrative Tools → ADFS2.0 Management



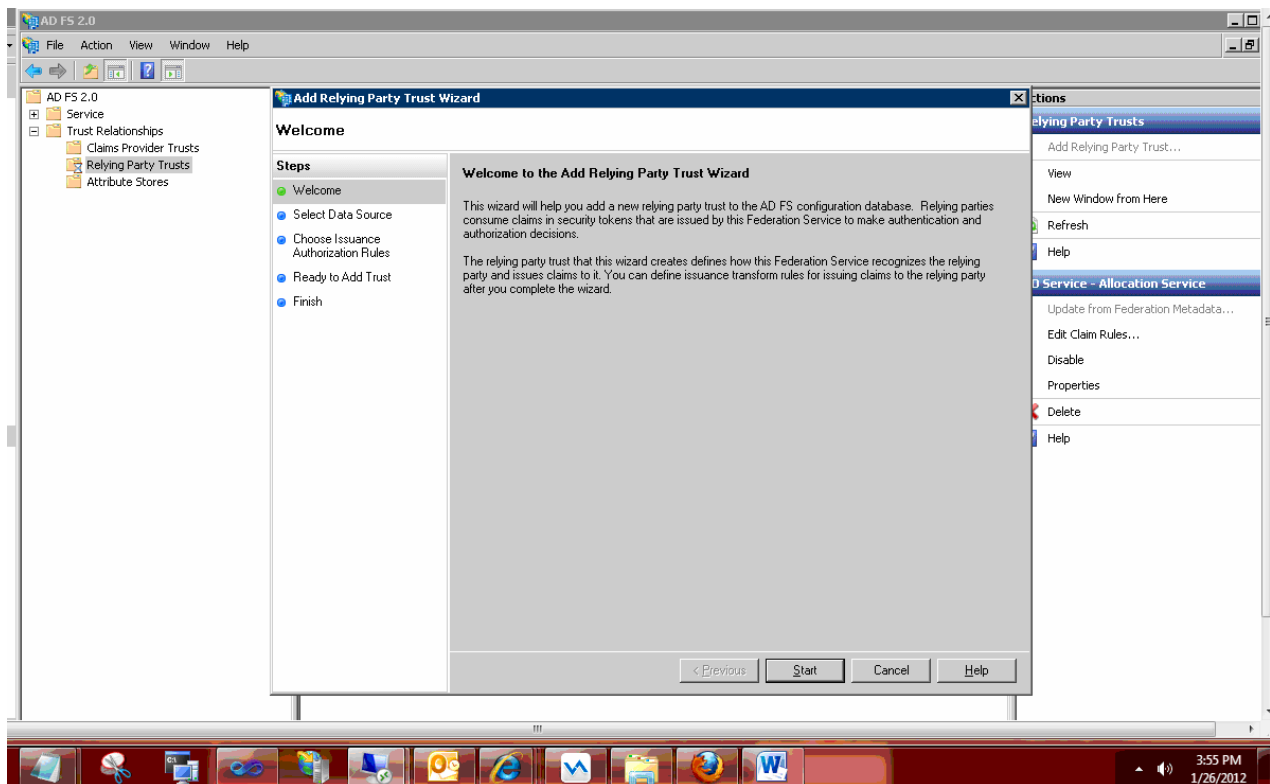
2. Select Relying Party trust from left pane



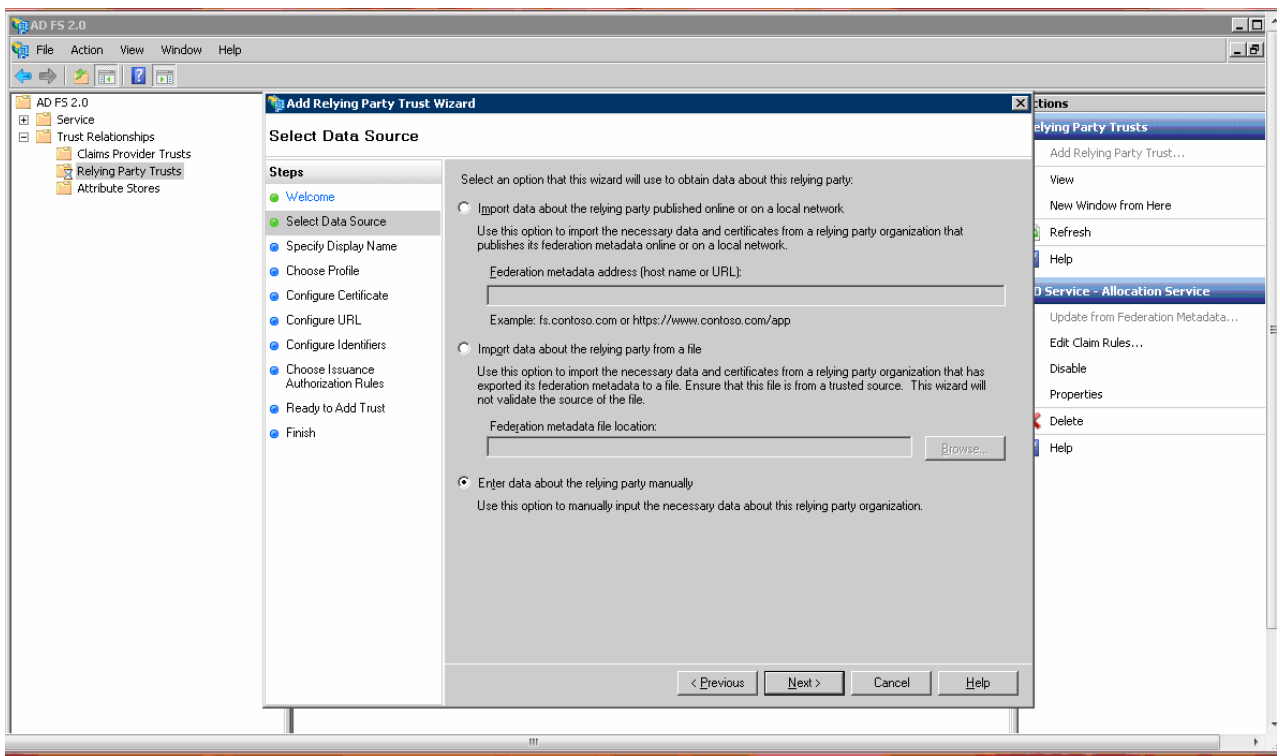
3. Right click on relying party Trust and select “Add Relying party Trust” menu.



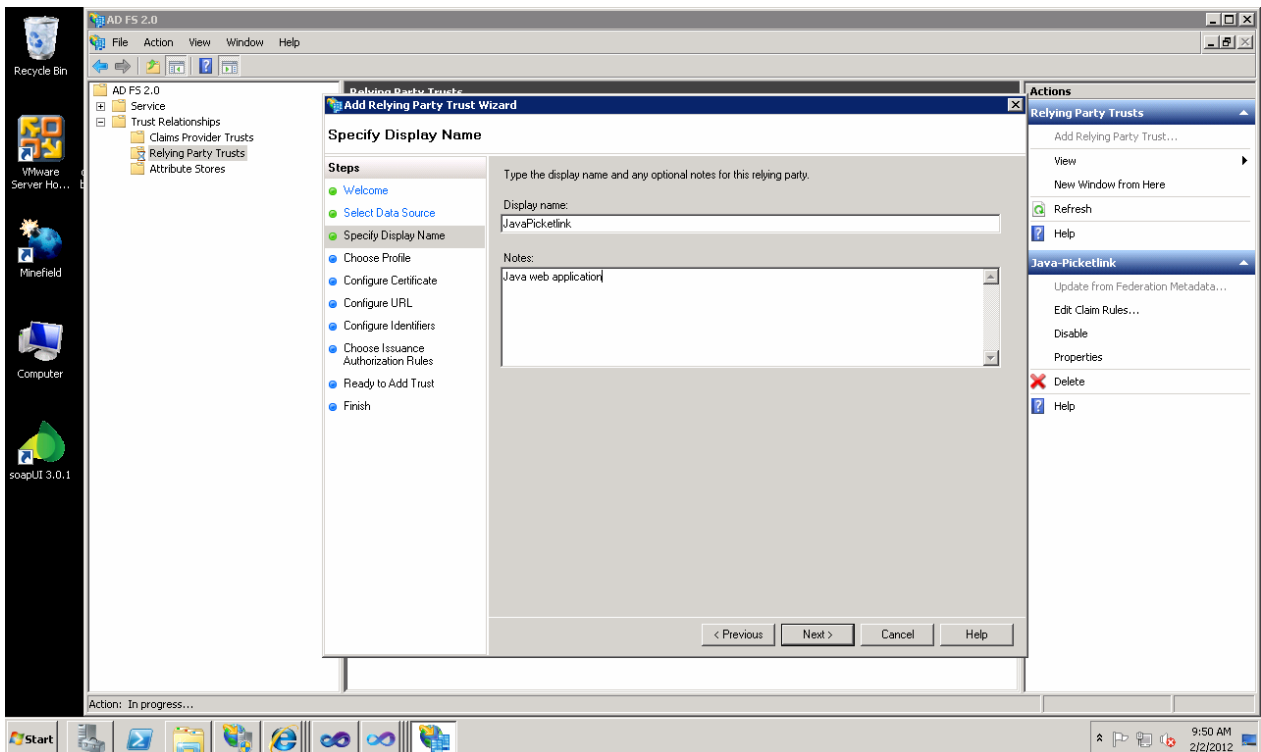
4. Click Start button



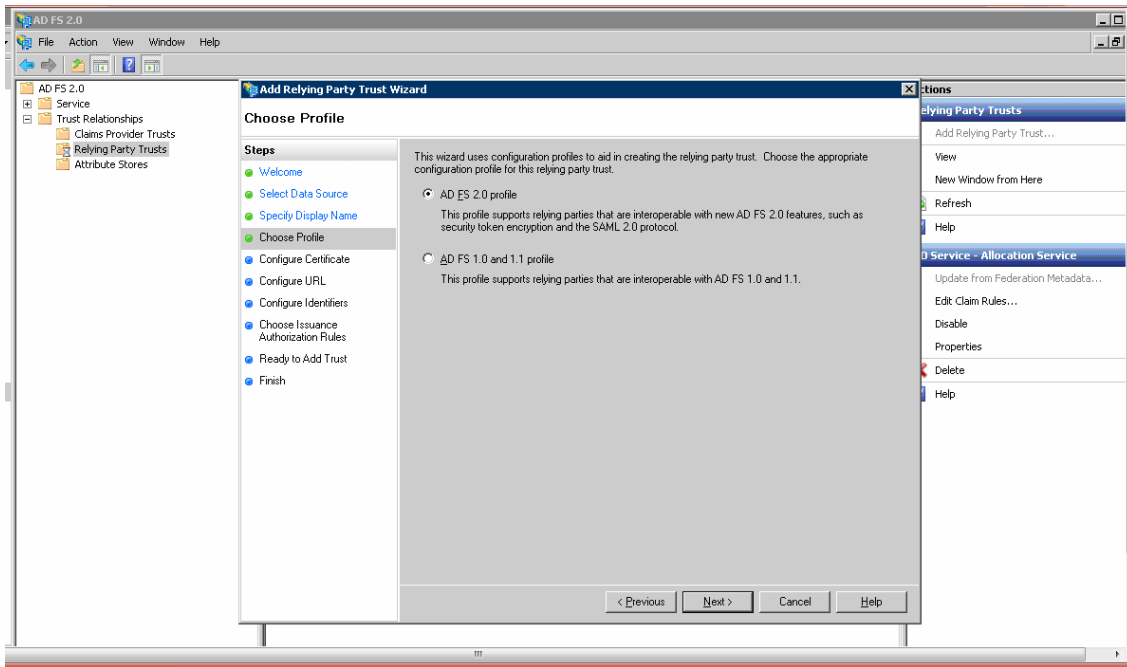
5. Select the 3rd option and Click Next button.



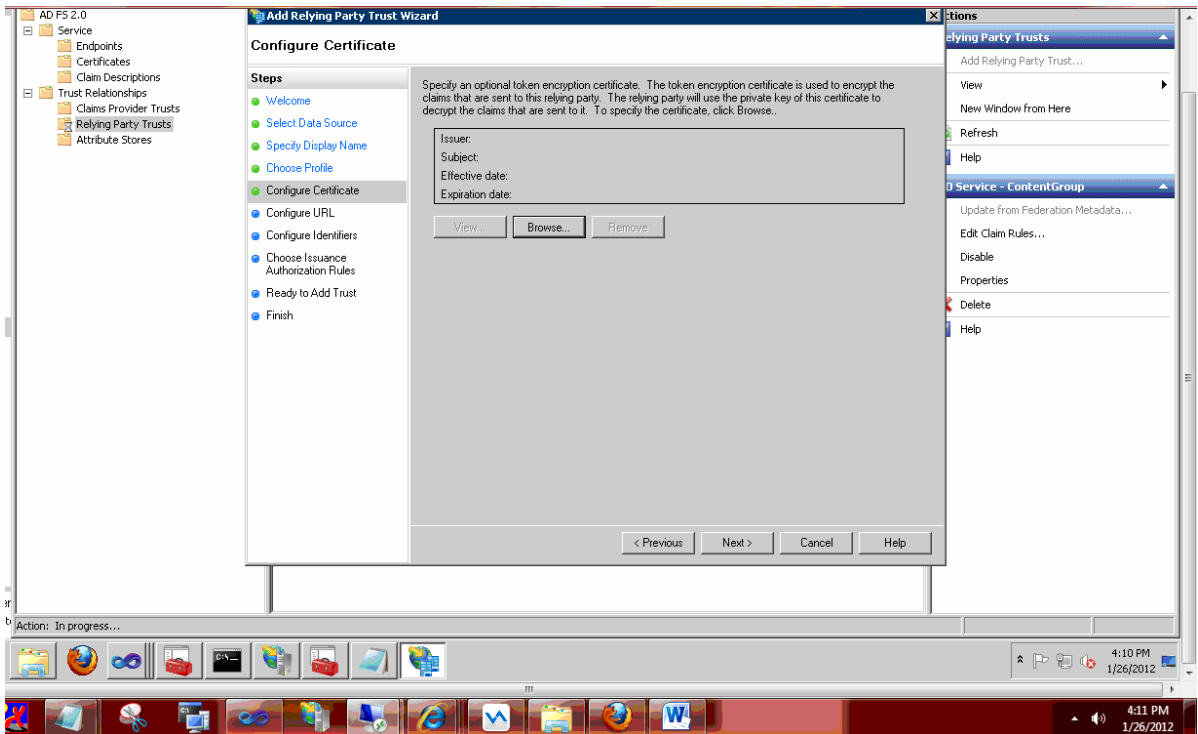
6. Enter Display name and description and click Next button.



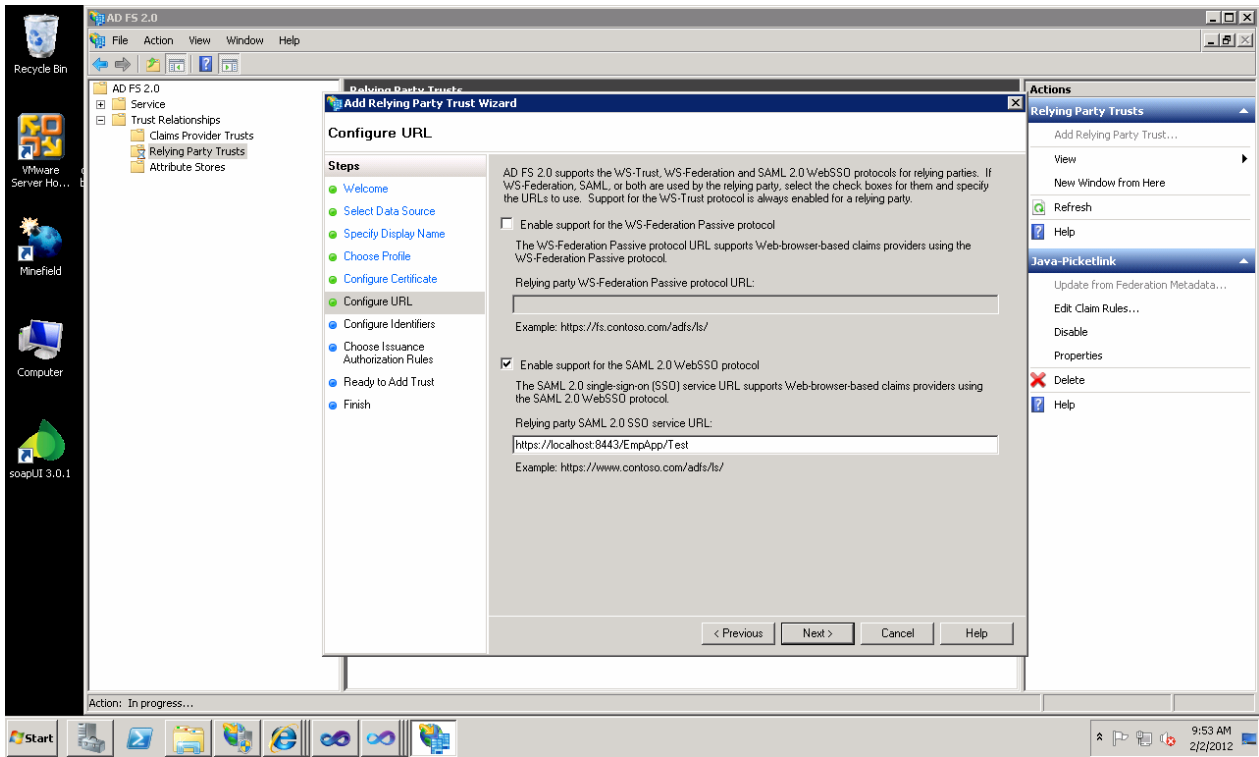
7. Select the ASFS 2.0 Profile option.



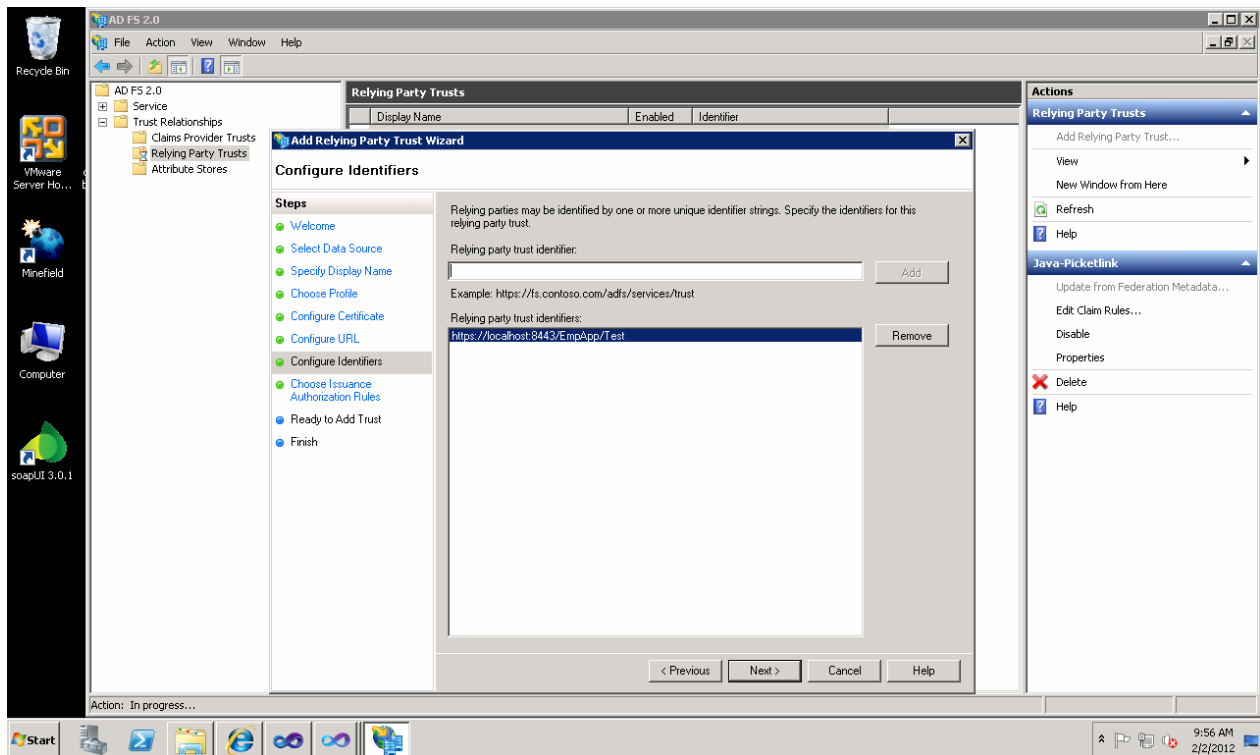
8. Browse the “composer5.cer” and click Next.



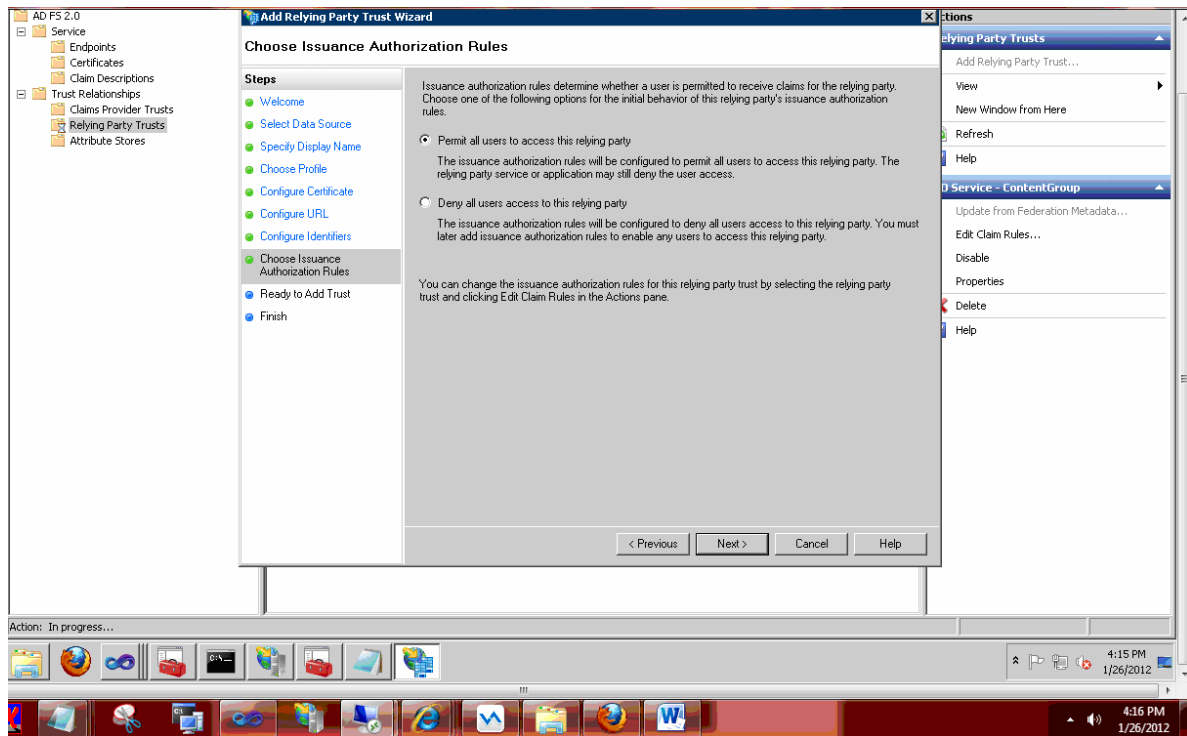
9. Select the “Enable support for SAML 2.0” option and Click Next button.



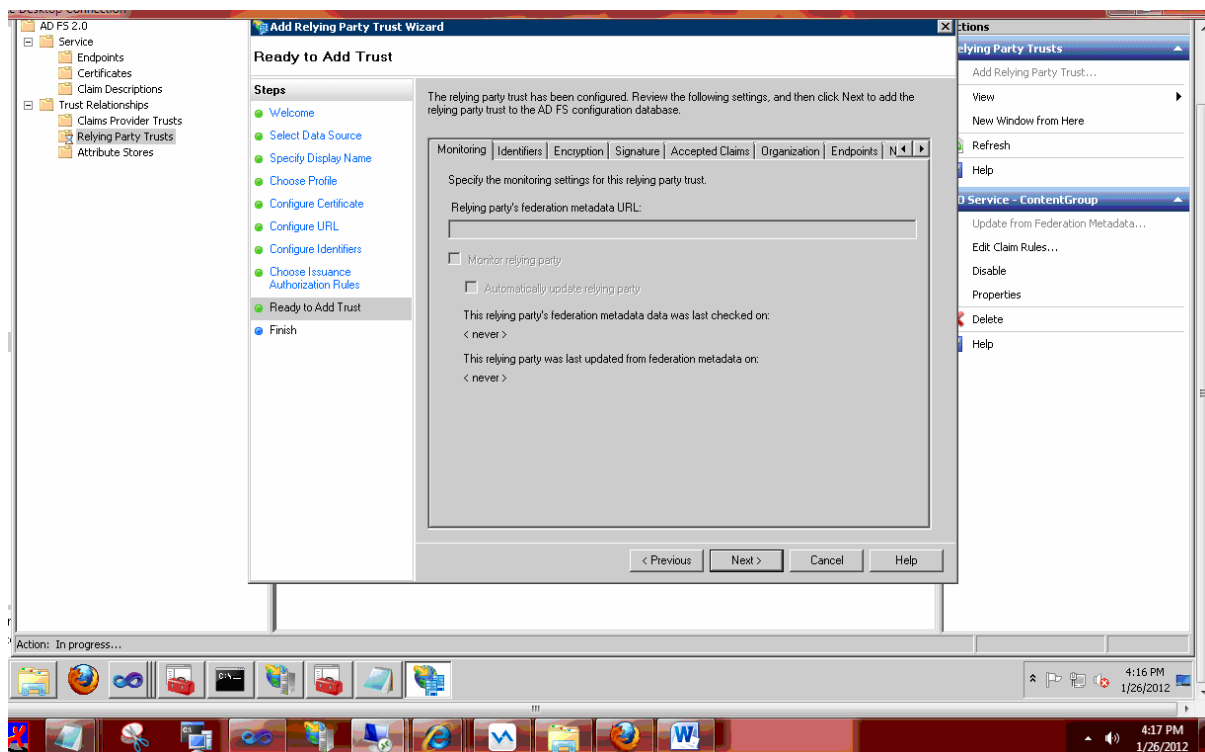
10. Add the Relying party service URL and click Next button.



## 11. Select the Permit all user Option.

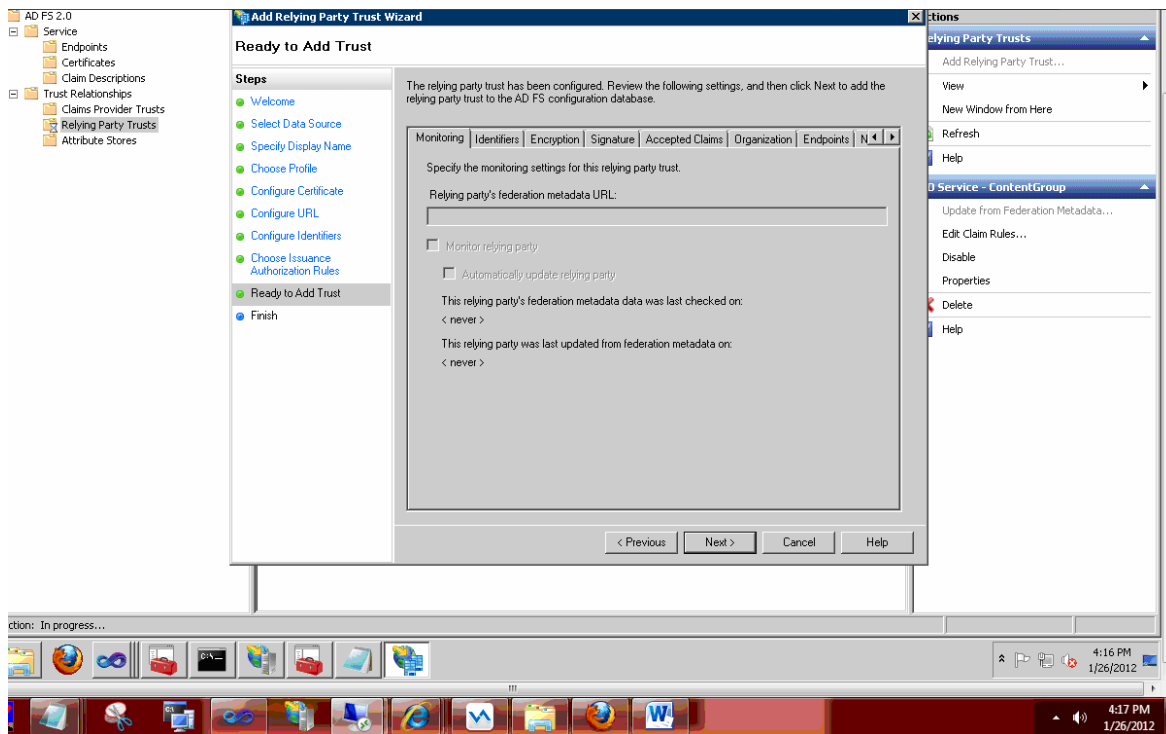


## 12. Click Next

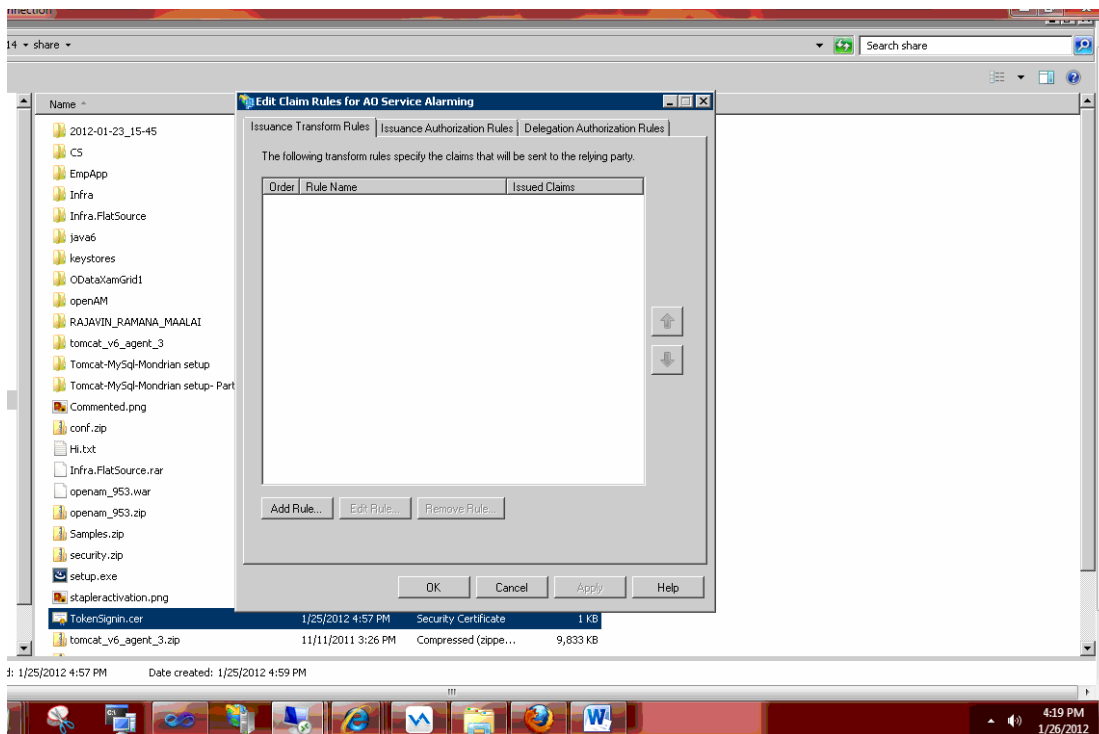




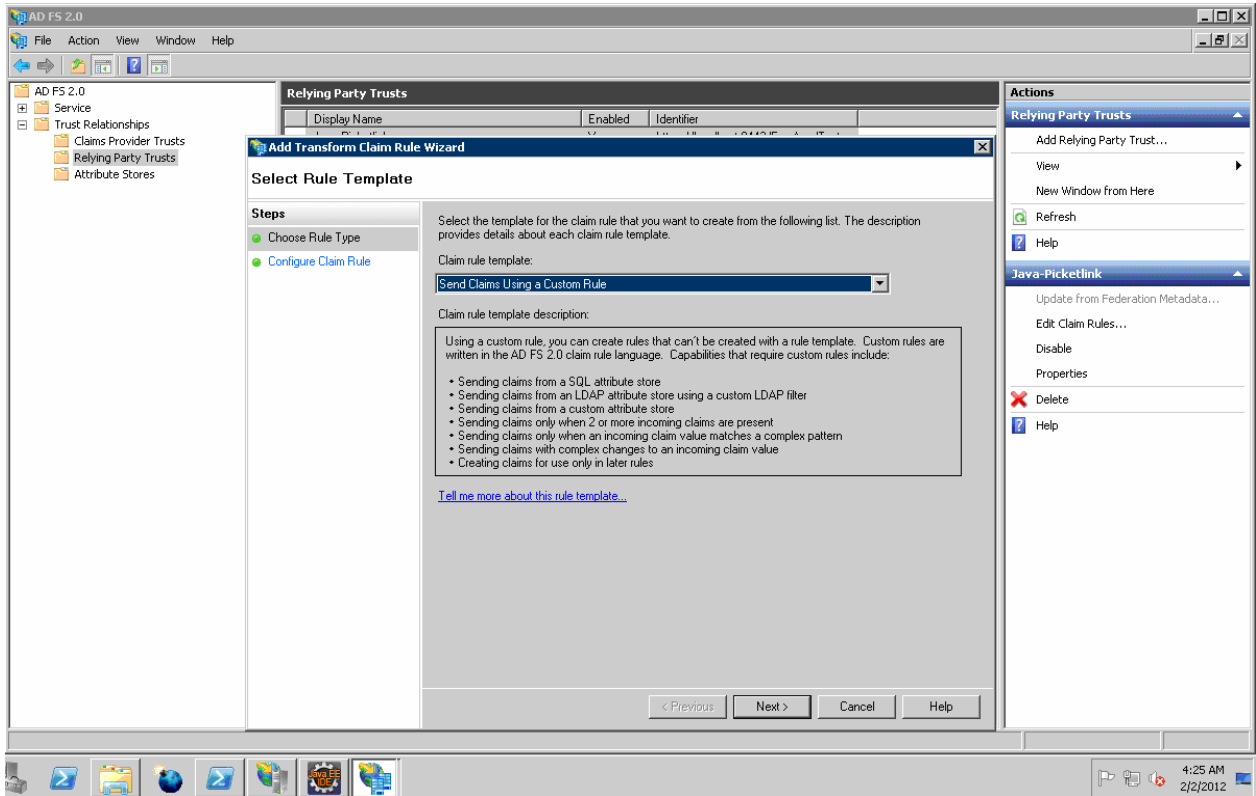
### 13. Click “Next” button and click Close Button



### 14. Now this will open a new window called as “Edit Claims Rules”

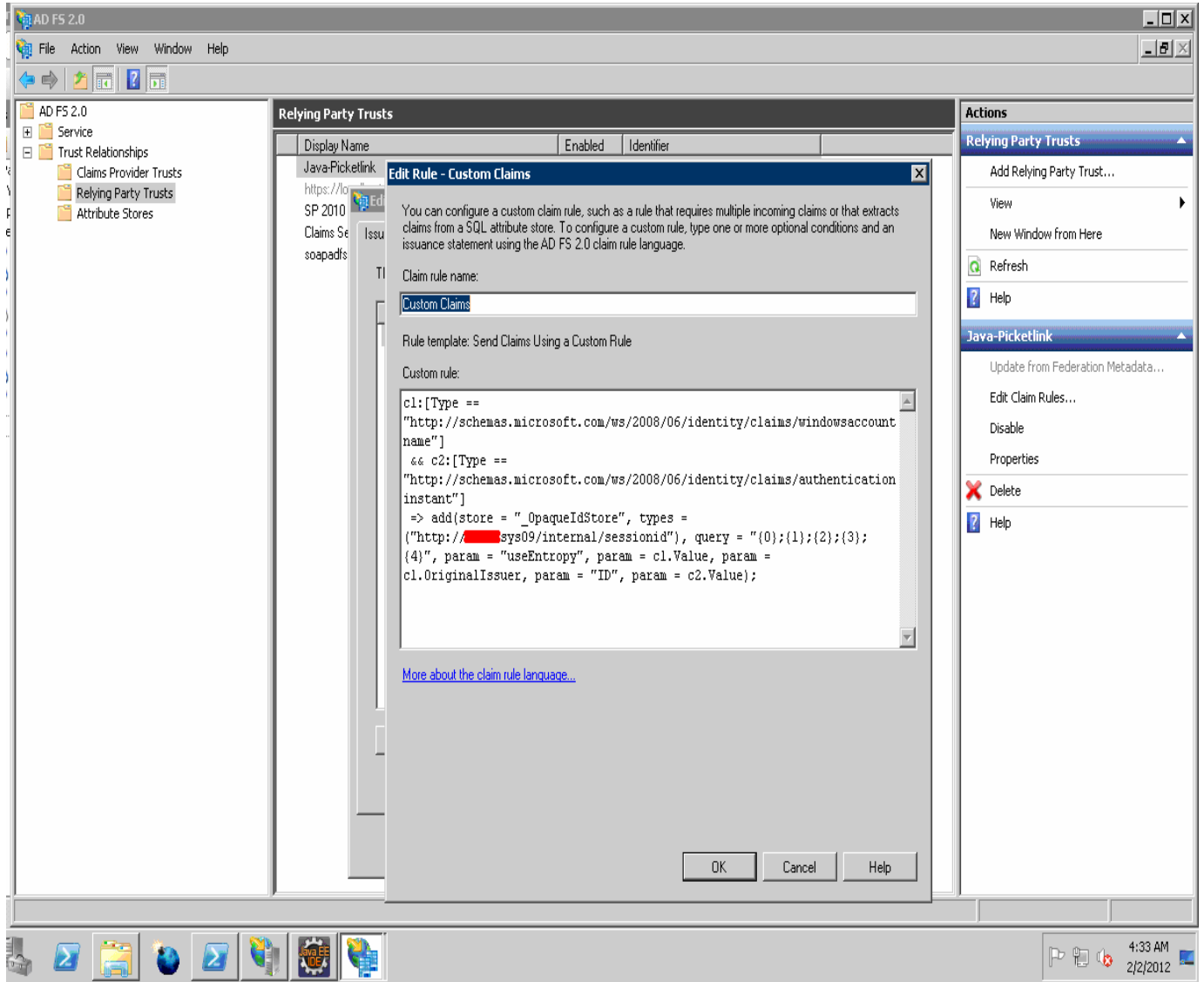


- Click "Add Rule" button and select the "Send Claims using Custom Rule" and Click Next Button.

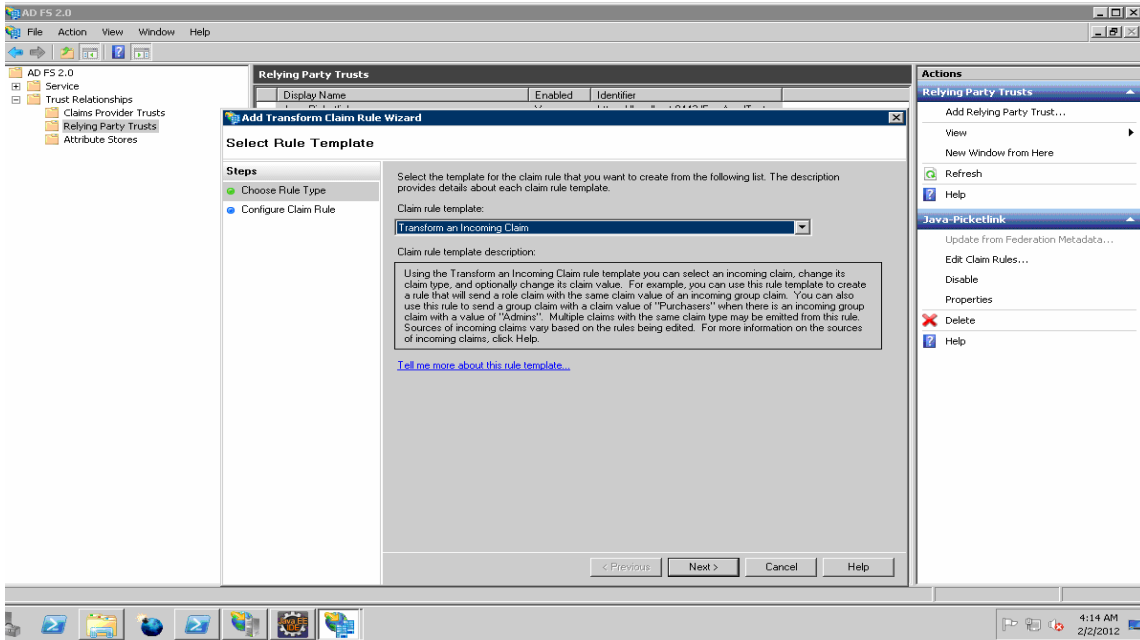


- Mention claims rule name and add the custom rule.

```
c1:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname"]
&& c2:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant"]
=> add(store = "_OpaqueldStore", types = ("http://sys09/internal/sessionid"), query =
"{0};{1};{2};{3};{4}", param = "useEntropy", param = c1.Value, param = c1.OriginalIssuer,
param = "ID", param = c2.Value);
```

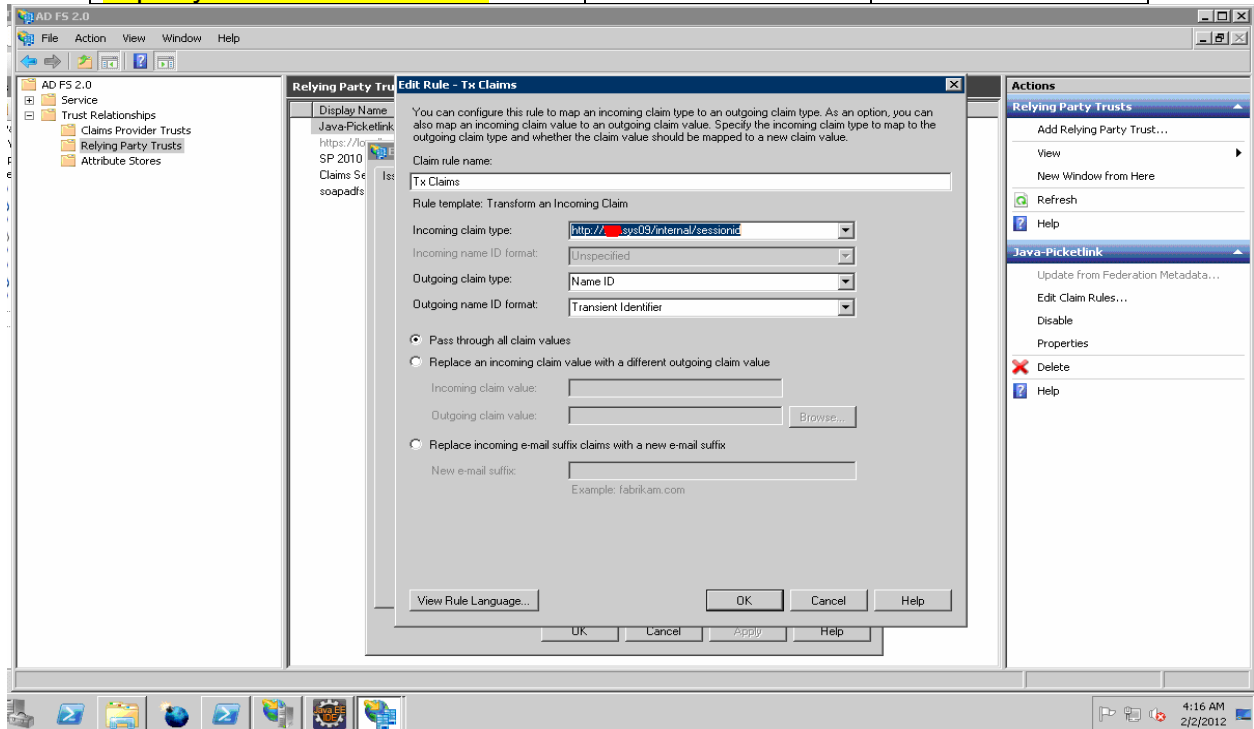


17. Click “Add Rule” button and select the “Transform Incoming as Claims” and Click Next Button.

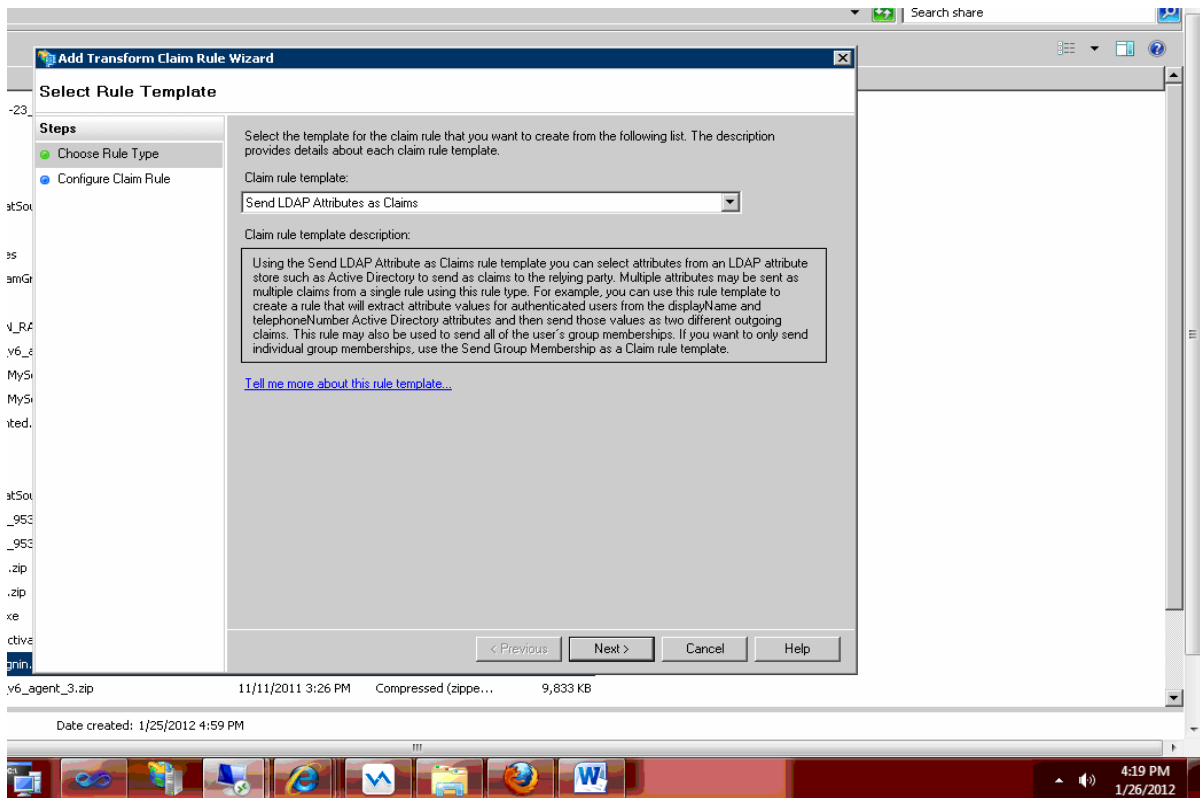


18. Enter the Following and Click Finish Button.

Incoming claim type.	Outgoing claim type.	Outgoing name Id format
<b>http://sys09/internal/sessionid</b>	Name ID	Transient Identifier



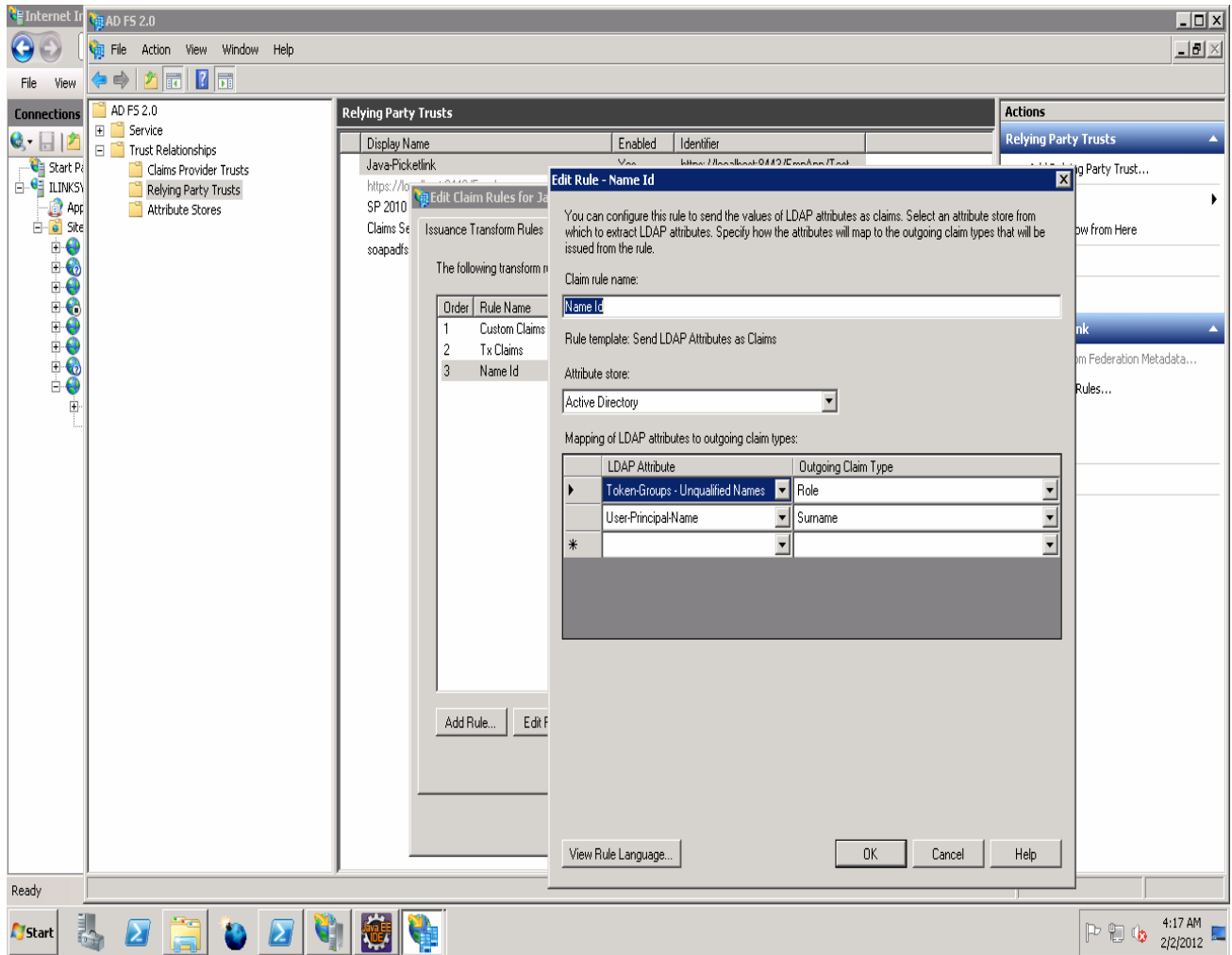
19. Click “Add Rule” button and select the “Send LDAP attribute as claim” and Click Next Button



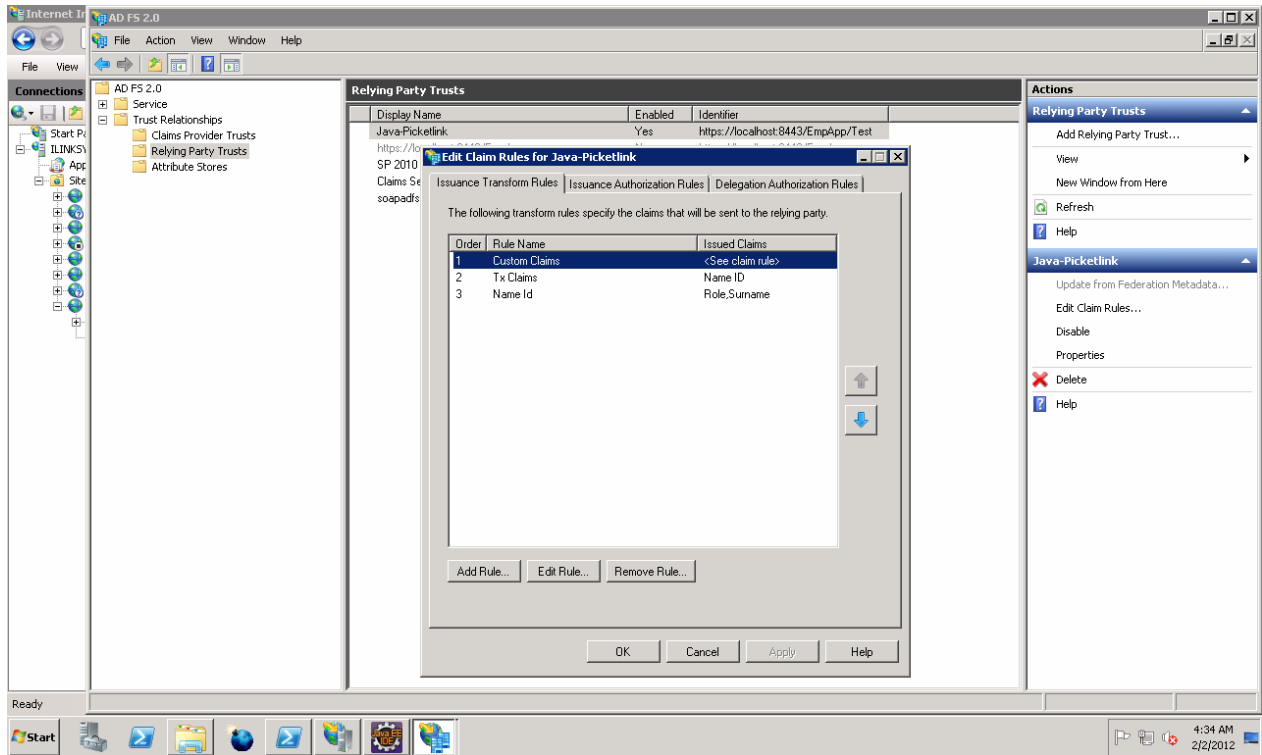
20. Enter the Following and Click Finish Button.

- Claim Rule name : Some Name
- Select “Active Directory” from Attribute store

LDAP Attribute	Outgoing claim type.
Token Groups – Unqualified Name	Role
UserPrincipalName	Surname



21. Click Apply and OK button in “Edit Claim Rule” window.



## Reference Links

1. <https://sourceforge.net/projects/portecle>
2. <http://community.jboss.org/wiki/HowtoconfigurePicketlinkonTomcatwithMicrosoftADFSv2>
3. <https://community.jboss.org/thread/153501>
4. <http://blogs.msdn.com/b/card/archive/2010/02/17/name-identifiers-in-saml-assertions.aspx>
5. <https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/>