

OpenSSO WS-Federation How-to

This document describes the sequence of actions required to deploy WS-Federation in OpenSSO.

READ THE ENTIRE DOCUMENT BEFORE YOU START, especially the 'Gotchas' section. The listed steps must be followed in strict order as there are many dependencies in the sequence, some not at all obvious.

In case of any problem, set debug logging to 'message', please zip the contents of the OpenSSO debug and log directories and the webcontainer's log directory and email them to users@opensso.dev.java.net with as detailed a description as possible.

Contents

1 References.....	1
2 Prerequisites.....	1
3 OpenSSO Install/Configuration.....	1
4 ADFS Install/Configuration.....	2
5 OpenSSO as Service Provider.....	2
5.1 ADFS Configuration.....	2
5.2 OpenSSO Configuration.....	2
6 OpenSSO as Identity Provider.....	5
6.1 OpenSSO Configuration.....	5
6.2 ADFS Configuration.....	8
7 Gotchas.....	9

1 References

1	Step-by-Step Guide for Active Directory Federation Services http://www.microsoft.com/downloads/details.aspx?familyid=062F7382-A82F-4428-9BBD-A103B9F27654&displaylang=en
---	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2 Prerequisites

A suitable container (any OpenSSO-supported container will do), **with SSL configured**. ADFS will not POST a WS-Federation RSTR to a non-HTTPS URL.

3 OpenSSO Install/Configuration

- Deploy OpenSSO WAR file – build from CVS or use any nightly after 8/1/07. Ensure that the services are available via SSL - you should now be able to log in securely to OpenSSO via `https://amhost(:amsecureport)/fam/console`
You'll have to accept the new cert in the browser. In IE, ensure you add the root cert, rather than the server cert itself.
- While you're logged in to OpenSSO, create a new test user – Directory Management/Users/New – this will correspond to a user we create later in Active Directory. The UID here has to be the same as the login name in AD.

4 ADFS Install/Configuration

- Setup ADFS as per 'Step-by-Step Guide for Active Directory Federation Services' [1]. Ensure that you can SSO from adfsaccount to adfsweb via adfsresource. Note that you can get by with only 2 machines – combine adfsaccount with adfsclient in the adatum.com domain and adfsresource with adfsweb in the treyresearch.net domain.
- Create a new user in the adatum.com AD domain, with login name the same as the UID you created in OpenSSO: Start/Administrative Tools/Active Directory Users and Computers, right click Users, New/User. Setting a different password here from the user's OpenSSO account is a good idea – it makes it obvious which credentials work where later on. If you have combined adfsaccount and adfsclient, then you will need to be able to login on the domain controller as this new user: after creating the new user, right click, Add to a group, and type 'DomainAdmins' (without the quotes) into the dialog.
- Where I don't mention an option in the ADFS configuration directions below, just leave it with the default.

5 OpenSSO as Service Provider

5.1 ADFS Configuration

- Add a new 'Resource Partner' to adfsaccount.adatum.com:
 - Display name: OpenSSO SP – this can be anything you like.
 - Federation Service URI: urn:federation:mywsfedsp – this MUST be the same as the TokenIssuerName in wsfedsp.xml below.
 - Federation Service endpoint URL:
`https://amhost(:amsecureport)/fam/WSFederationServlet/metaAlias/mywsfedsp`
– the last path component of this URL MUST match metaAlias in wsfedsp.xml below.

5.2 OpenSSO Configuration

- Convert the AD machine's token signing certificate file (e.g. adfsaccount_ts.cer) to PEM format. You can use OpenSSL for this: `openssl x509 -in adfsaccount_ts.cer -inform DER -out adfsaccount_ts.pem -outform PEM`
- Create the following files:

adatum.xml – you will need to paste the PEM-encoded certificate from adfsaccount_ts.pem into the <ns2:X509Certificate> element.

NOTE – do not cut and paste the XML from this PDF – Acrobat inserts line breaks all over. Instead, generate template files with `famadm create-metadata-template -u amadmin -f passwordfile -m metadatafile.xml -x extendedfile.xml -i /metaalias -y entityid -c wsfed` (for the identity provider) or `famadm create-metadata-template -u amadmin -f passwordfile -m metadatafile.xml -x extendedfile.xml -s /metaalias -y entityid -c wsfed` (for the service provider)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Federation FederationID="urn:federation:adatum"
```

```
<?xml version='1.0' encoding='utf-8'>
  <TokenSigningKeyInfo>
    <ns1:SecurityTokenReference ns1:Usage="
xmlns:ns1="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <ns2:X509Data
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
        <ns2:X509Certificate>
MIIC0DCCAbygAwIBAgIQ/SQKpB08uqtJ/4BwGzpTODAJBgUrDgMCHQUAMCgxJjAk
BgNVBAMTHUZlZGVyYXRpb24gU2VydmVyIGFkZnNhY2NvdW50MB4XDTA3MDMxNjAw
MTIyMVoXDTA4MDMxNTA2MTIyMVowKDEmMCQGA1UEAxMdRmVkZXJhdGlvbiBTZXJ2
ZXIgaWYwRmM2FjY291bnQwggeEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCk
TBDiJ3KUU1/1Z2k1DFlszlztP+IzuB4TSvUKn75FYQlJfzawuUhGdugoZuUpkWlP
mEWi4HsKtL/cyVN+KmvihbUCPvBWq9k/7J37xXi93r85hEq3KCI+1IIfr354qqWO
tGChPUG258rElcTNIeP2IKIJxhcp7LGeTcryAgSGbj5lT6NrFX4dCM8jyzQjY1xP
UPpgoxb44cIpg8wW/ jnbJMYMHOJfBEK57sXtU3+4sw75757QZA/QSOlkYfBgJa7c
y5kemHENvN/tQiCDcrW9HoVhJnzy0RLLx6QO0MdDSx1R/AXADYkPfh1EsqWnpUoe
Q89Rh2mQKEXh7Zc8rZZJAgMBAAEwCQYFKw4DAh0FAAOCAQEAoivAAoYpG+ka7qTo
28xaJnZrStMGB04faHoVpy9j6CaUsk/tweUzDVVxSvhOYXCNPYkP9yWDUv5md46Q
xvhhfrrQvXHkbbWlX+PAMjBkH9roTT8xyv0i9+anOZd7V46iSl0bcSeSQvaUH2iB2
w+dGEFwKJSNs/8Cl8Ib157Uq6kwKoJvhn6zGb9j9tr4ryoa6UvDB73AYcIg3imX0
tkKZ9/rNkKaV9R9TfykzX5Tgih348FFtSElSp12QaTOTs/Ct2WK5enBz0Dlc83sb
wQ3sBrvlZPP/gwpqW6fQuLnH2tZSF9l0SNNDlWnpDMnM9KqVRLyKI6fXDwnaL7ZB
22xkiw==
        </ns2:X509Certificate>
      </ns2:X509Data>
    </ns1:SecurityTokenReference>
  </TokenSigningKeyInfo>
  <TokenIssuerName>urn:federation:adatum</TokenIssuerName>
  <TokenIssuerEndpoint>
    <ns3:Address
xmlns:ns3="http://www.w3.org/2005/08/addressing">https://adfsaccount.adat
um.com/adfs/ls/</ns3:Address>
  </TokenIssuerEndpoint>
  <TokenTypesOffered>
    <TokenType Uri="urn:oasis:names:tc:SAML:1.1"/>
  </TokenTypesOffered>
  <UriNamedClaimTypesOffered>
    <ClaimType Uri="http://schemas.xmlsoap.org/claims/UPN">
      <DisplayName>UPN</DisplayName>
    </ClaimType>
  </UriNamedClaimTypesOffered>
</Federation>
```

```
<FederationConfig xmlns="urn:sun:fm:wsfederation:1.0:federationconfig"
  xmlns:fm="urn:sun:fm:wsfederation:1.0:federationconfig"
  hosted="0"
  FederationID="urn:federation:adatum">
  <IDPSSOConfig metaAlias="/adatumidp">
    <Attribute name="DisplayName">
      <Value>Adatum Corp</Value>
    </Attribute>
  </IDPSSOConfig>
</FederationConfig>
```

```
</IDPSSOConfig>
</FederationConfig>
```

wsfedsp.xml – you will need to change the hostname and port in the `<ns3:Address>` element to match your configuration.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Federation FederationID="mywsfedsp"
xmlns="http://schemas.xmlsoap.org/ws/2006/12/federation">
  <TokenIssuerName>urn:federation:mywsfedsp</TokenIssuerName>
  <TokenIssuerEndpoint>
    <ns3:Address
xmlns:ns3="http://www.w3.org/2005/08/addressing">https://patlinux.red.ip
lanet.com:8443/fam/WSFederationServlet/metaAlias/mywsfedsp</ns3:Address>
  </TokenIssuerEndpoint>
</Federation>
```

wsfedsp.xml – you will need to change the hostname and port in the `HomeRealmDiscoveryService` attribute to match your configuration.

```
<FederationConfig xmlns="urn:sun:fm:wsfederation:1.0:federationconfig"
xmlns:fm="urn:sun:fm:wsfederation:1.0:federationconfig"
hosted="1" FederationID="mywsfedsp">
  <SPSSOConfig metaAlias="/mywsfedsp">
    <Attribute name="displayName">
      <Value>My Open Federation Service Provider</Value>
    </Attribute>
    <Attribute name="AccountRealmSelection">
      <Value>cookie</Value>
    </Attribute>
    <Attribute name="AccountRealmCookieName">
      <Value>amWSFederationAccountRealm</Value>
    </Attribute>
    <Attribute name="HomeRealmDiscoveryService">
      <Value>http://patlinux.red.ipplanet.com:8180/fam/RealmSelectio
n/metaAlias/mywsfedsp</Value>
    </Attribute>
    <Attribute name="spAccountMapper">
      <Value>com.sun.identity.wsfederation.plugins.DefaultADFSPartn
erAccountMapper</Value>
    </Attribute>
    <Attribute name="spAttributeMapper">
      <Value>com.sun.identity.wsfederation.plugins.DefaultSPAttribu
teMapper</Value>
    </Attribute>
  </SPSSOConfig>
</FederationConfig>
```

- Create a circle of trust and import the newly created metadata using famadm:

```
famadm create-circle-of-trust -u amadmin -w password -t cot1
```

```
famadm import-entity -u amadmin -w password -m adatum.xml -x adatumx.xml -t
```

```
cotl -c wsfed
```

```
famadm import-entity -u amadmin -w password -m wsfedsp.xml -x wsfedsp.xml -t  
cotl -c wsfed
```

- If all is well, you should be able to go to `https://amhost(:amsecureport)/fam/WSFederationServlet/metaAlias/mywsfedsp?goto=https://amhost(:amsecureport)/fam` and be forwarded to the realm selection page. Click 'Proceed' and you'll see a few redirections in the browser's address bar before landing at the user's OpenSSO page. If you do this from outside the Window domain, you'll get an HTTPbasic auth-style username/password dialog. Enter the test user's AD credentials and you should be in. The realm selection process sets a persistent cookie; if you go to the above URL a second time, you should not be prompted for a realm – you should be redirected straight to the OpenSSO user page.
- You should now be able to configure a policy agent with the WS-Federation servlet as its login URL – for Java EE agents, set

```
com.sun.identity.agents.config.login.url[0]=https://amhost(:amsecureport)/fam/WSFederationServlet/metaAlias/mywsfedsp
```

for web agents, set

```
com.sun.am.policy.am.login.url=https://amhost(:amsecureport)/fam/WSFederationServlet/metaAlias/mywsfedsp
```

Now, when accessing the resource protected by the policy agent, you should be authenticated via WS-Federation – that is, have seamless access from the Windows desktop.

6 OpenSSO as Identity Provider

6.1 OpenSSO Configuration

- Create a token signing certificate in a Java Keystore on the OpenSSO machine:

```
keytool -keystore keystore.jks -genkey -dname "CN=amhost" -alias mywsfedidp
```

Specify the same password for the keystore and key. You can put the keystore anywhere, since you will need to specify the full path in `AMConfig.properties` (see below).

- You must encrypt the keystore/key password. The easiest way to do this is to go to `https://amhost(:amsecureport)/fam/encode.jsp` and enter the password. Create two files, `.storepass` and `.keypass`, whose only content is the encrypted password.
- Edit `FederationConfig.properties` (this will be in the configuration directory that you specified when installing OpenSSO) and edit the following lines:

```
com.sun.identity.saml.xmlsig.keystore=/path/to/keystore.jks  
com.sun.identity.saml.xmlsig.storepass=/path/to/.storepass  
com.sun.identity.saml.xmlsig.keypass=/path/to/.keypass
```

- Export the token signing cert in DER format:

```
keytool -keystore keystore.jks -export -alias mywsfedidp -file cert.der
```

Copy cert.der to the adfsresource machine.

- Create the following files:

treyresearch.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Federation FederationID="treyresearch"
xmlns="http://schemas.xmlsoap.org/ws/2006/12/federation">
  <TokenIssuerName>urn:federation:treyresearch</TokenIssuerName>
  <TokenIssuerEndpoint>
    <ns3:Address
xmlns:ns3="http://www.w3.org/2005/08/addressing">https://adfsresource.tre
yresearch.net/adfs/ls/</ns3:Address>
  </TokenIssuerEndpoint>
</Federation>
```

treyresearch.xml

```
<FederationConfig xmlns="urn:sun:fm:wsfederation:1.0:federationconfig"
  hosted="false" FederationID="treyresearch">
  <SPSSOConfig>
  </SPSSOConfig>
</FederationConfig>
```

wsfedidp.xml – you will need to change the hostname and port in the <ns3:Address> element to match your configuration.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Federation FederationID="mywsfedidp"
xmlns="http://schemas.xmlsoap.org/ws/2006/12/federation">
  <TokenSigningKeyInfo>
    <ns1:SecurityTokenReference ns1:Usage=""
xmlns:ns1="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <ns2:X509Data
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
        <ns2:X509Certificate>MIIB7jCCAVcCBEaK0dswDQYJKoZIhvcNAQEE
BQAwPjEZMBcGA1UEChMQU3VuIElpy3Jvc3lzdGVtczEhMB8GA1UEAxMYcGF0bGludXgucmVh
LmlwbGFuZlZlY29tMB4XDTA3MDcwMzIyNDY1MVVoXDE3MDYzMDIyNDY1MVowPjEZMBcGA1UEC
hMQU3VuIElpy3Jvc3lzdGVtczEhMB8GA1UEAxMYcGF0bGludXgucmVhLmlwbGFuZlZlY29tM
IGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDIAOaHI4mkqqPbKfOwDgQhhRfKEfTO3MstF
qD2SxaNoeFmtjpvPNDQcWzYC3gtXPaj8df8ITJ891fBLqQHUWepZECdRgcBHX3ZDKJ3YUffX9
22G12Tz40iigaJ5U9FWUmDW1i0zv5IaqZzdHNQx4BAkQk53wi2zcBmp2Jpzn3LFwIDAQABMA
0GCSqGSIb3DQEBAQUAA4GBALrnlD7r9NGCVn6fxtEWGItV0VFADT9TkIkI5fw3AsnjiJ02V+o
QHPXpv8nZ7lTyy/OYhGNpPLi8LEVcRt+DqtM8goL+cMJE24kHatPXOZqy1gOIzqTfZpkMNMoe
xQIfqjrZWkYXWad9cmZW1H7L+4o1WZvpNw42rM4zkHY+nS</ns2:X509Certificate>
      </ns2:X509Data>
    </ns1:SecurityTokenReference>
  </TokenSigningKeyInfo>
  <TokenIssuerName>urn:federation:mywsfedidp</TokenIssuerName>
  <TokenIssuerEndpoint>
    <ns3:Address
```

```

xmlns:ns3="http://www.w3.org/2005/08/addressing">https://patlinux.red.ip
lanet.com:8443/fam/WSFederationServlet/metaAlias/mywsfedidp</ns3:Address>
  </TokenIssuerEndpoint>
  <TokenTypesOffered>
    <TokenType Uri="urn:oasis:names:tc:SAML:1.1"/>
  </TokenTypesOffered>
  <UriNamedClaimTypesOffered>
    <ClaimType Uri="http://schemas.xmlsoap.org/claims/UPN">
      <DisplayName>UPN</DisplayName>
    </ClaimType>
  </UriNamedClaimTypesOffered>
</Federation>

```

wsfedidpx.xml – you will need to change the hostname and port in the HomeRealmDiscoveryService attribute to match your configuration.

```

<FederationConfig xmlns="urn:sun:fm:wsfederation:1.0:federationconfig"
  xmlns:fm="urn:sun:fm:wsfederation:1.0:federationconfig"
  hosted="1" FederationID="mywsfedidp">
  <IDPSSOConfig metaAlias="/mywsfedidp">
    <Attribute name="displayName">
      <Value>My Open Federation Identity Provider</Value>
    </Attribute>
    <Attribute name="upnDomain">
      <Value>iplanet.com</Value>
    </Attribute>
    <Attribute name="signingCertAlias">
      <Value>mywsfedidp</Value>
    </Attribute>
    <Attribute name="assertionEffectiveTime">
      <Value>600</Value>
    </Attribute>
    <Attribute name="idpAccountMapper">
      <Value>com.sun.identity.wsfederation.plugins.DefaultIDPAccountMapper</Value>
    </Attribute>
    <Attribute name="idpAttributeMapper">
      <Value>com.sun.identity.wsfederation.plugins.DefaultIDPAttributeMapper</Value>
    </Attribute>
  </IDPSSOConfig>
</FederationConfig>

```

- Import the newly created metadata using famadm:

```

famadm import-entity -u amadmin -w password -m treyresearch.xml -x
treyresearchx.xml -t cot1 -c wsfed

```

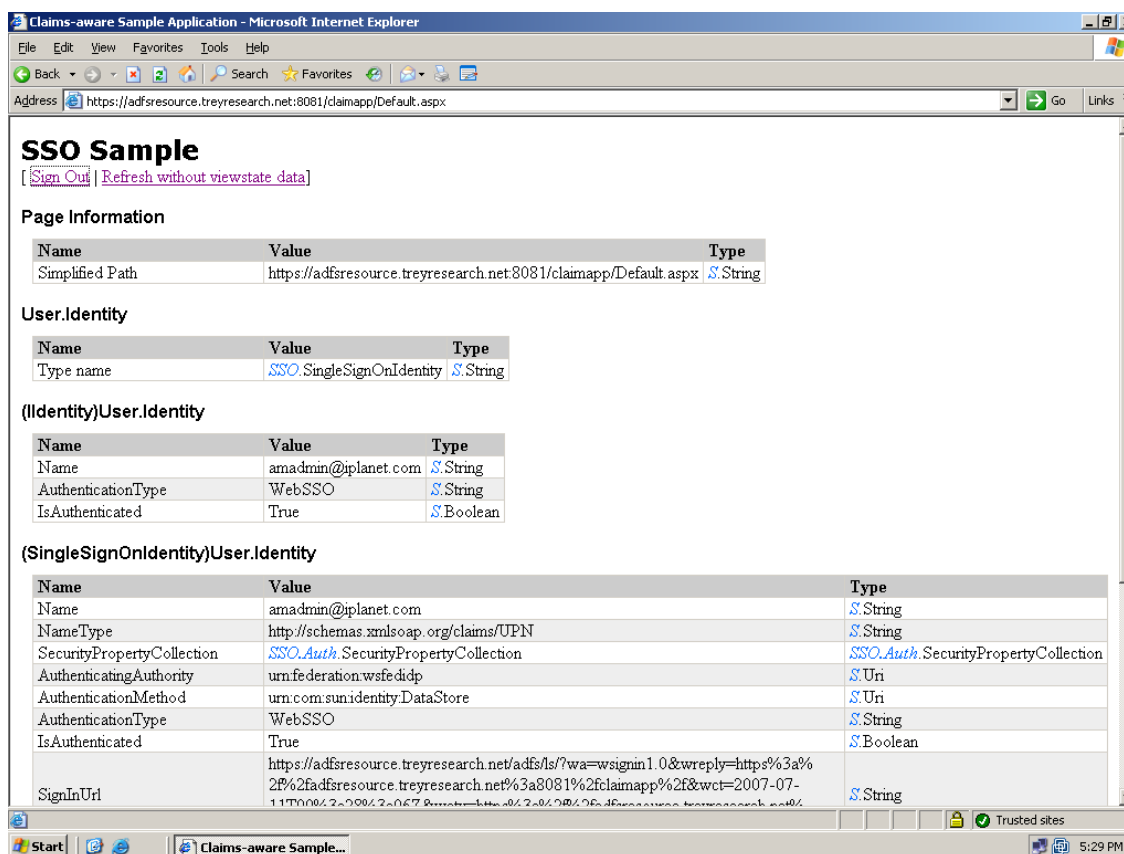
```

famadm import-entity -u amadmin -w password -m wsfedidp.xml -x wsfedidpx.xml
-t cot1 -c wsfed

```

6.2 ADFS Configuration

- Add a new 'Account Partner' to `adsresource.treyresearch.net`:
 - Display name: `OpenSSO IdP` – this can be anything you like.
 - Federation Service URI: `urn:federation:mywsfedidp` – this MUST match the `TokenIssuerName` in `wsfedidp.xml` below.
 - Federation Service endpoint URL:
`https://amhost(:amsecureport)/fam/WSFederationServlet/metaAlias/mywsfedidp` – the last path component of this URL MUST match `metaAlias` in `wsfedidpx.xml` below.
 - Account Partner Verification Certificate: import the OpenSSO token signing certificate that you copied to the `adsresourcemachine`.
- Delete all cookies in your browser and go to the sample claims-aware application - `https://adsweb.treyresearch.net:8081/claimapp/` - you should see the OpenSSO IdP listed in the dropdown list. Select the OpenSSO IdP and you should be redirected to the standard OpenSSO login screen. After logging in, you should be redirected back to the sample application and see output similar to this:



- You can click the 'Sign Out' link to do a single logout. You can check that you are logged out by trying the `https://adfsweb.treyresearch.net:8081/claimapp/` URL again. You should be redirected to the OpenSSO login page, demonstrating that neither ADFS or OpenSSO have an active session for the browser.
- Again, the realm choice is stored in a persistent cookie. If you close and restart the browser then

return to <https://adfsweb.treyresearch.net:8081/claimapp/>, you should proceed to the OpenSSO login page directly.

7 Gotchas

- The ADFS box's clock needs to be at the same time as the OpenSSO box. Configuring NTP everywhere is a good solution.
- If you do a desktop SSO from a browser window, then try to do another one in quick succession from the same window, ADFS will fail with an error stating that you have tried to request two tokens from the same browser session within a 20 second window. Just close the browser window(s) and try again – this deletes the non-persistent cookies.
- The logo jpg on the realm selection and login pages looks a bit crap in IE. It seems to have transparent areas top and bottom.