How to: Configure K2 Services to support Claims

KB Number: KB001426
Date Modified:

Introduction

This article explains how to configure K2 Services to support Claims Authentication. The steps provided for the K2 Web Services configuration are based on the Active Directory Federation Services 2.0 being used as the Secure Token Service, which will be referred to as ADFS from this point forward. Other Secure Token Services can be used however the configuration changes might be different.

Two configuration options are explained in this article:

1. K2 Web Services (WCF and Rest Endpoints)
2. K2 SmartObject Services

Configuring K2 Web Services for Claims support

To configure the K2 Web Services for Claims support, the first action is to generate FederationMetadata.xml which is the K2 Services Endpoints setup to support Claims. The steps below are based on ADFS being used as the Secure Token Service.

Generate FederationMetadata.xml

1. Run "C:\Program Files (x86)\Windows Identity Foundation SDK\v4.0\FedUtil.exe".
2. Select C:\Program Files (x86)\K2 blackpearl\WebServices\K2Services\web.config.
3. Specify http://{Server Name/Host Header}/K2Services/WCF.svc as the Application URI.
4. Click next on the list of endpoints.
5. Use an existing Secure Token Service, enter the **ADFS URL** and click on **Test location**.
6. Close the website once opened and click Next.
7. Choose **Disable certificate chain validation.**
8. Enable Encryption and choose **Select an existing certificate.** Select the required certificate.
9. Click Next to complete the process.

Configure the K2 Services web.config to enable all endpoints

1. Open the web.config located at C:\Program Files (x86)\K2 blackpearl\WebServices\K2Services and edit the endpoints to reflect the changes as illustrated below.
   Locate <service behaviorConfiguration="SourceCode.Services.RestBehavior" name="SourceCode.Services.Rest">

```
<endpoint
address="http://{ServerName/HostHeader}/K2Services/REST.svc/Process"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.IProcessNavigationService"
behaviorConfiguration="SourceCode.Services.RestBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding"  />
<endpoint
address="http://{ServerName/HostHeader}/K2Services/REST.svc/Worklist"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.IWorklistNavigationService"
```

```
behaviorConfiguration="SourceCode.Services.RestBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding" />
<endpoint
address="http://{ServerName/HostHeader}/K2Services/REST.svc/Identity"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.IIdentityService"
behaviorConfiguration="SourceCode.Services.RestBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding"  />
<endpoint address="http://{ServerName/HostHeader}/K2Services/REST.svc/Core"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.ICoreService"
behaviorConfiguration="SourceCode.Services.RestBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding"  />
```

2. Edit the endpoints to reflect the changes as illustrated below.
   Locate <service behaviorConfiguration="SourceCode.Services.SoapBehavior" name="SourceCode.Services.Wcf">

```
<endpoint address="http://{ServerName/HostHeader}/K2Services/WCF.svc/Process"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.IProcessNavigationService"
behaviorConfiguration="SourceCode.Services.SoapBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding"  />
<endpoint
address="http://{ServerName/HostHeader}/K2Services/WCF.svc/Worklist"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.IWorklistNavigationService"
behaviorConfiguration="SourceCode.Services.SoapBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding" />
<endpoint
address="http://{ServerName/HostHeader}/K2Services/WCF.svc/Identity"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.IIdentityService"
behaviorConfiguration="SourceCode.Services.SoapBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding" />
<endpoint address="http://{ServerName/HostHeader}/K2Services/WCF.svc/Core"
binding="ws2007FederationHttpBinding"
contract="SourceCode.Services.ServiceContracts.ICoreService"
behaviorConfiguration="SourceCode.Services.SoapBehavior"
bindingConfiguration="SourceCode.Services.ws2007FederationHttpBinding" />
```

3. Edit the binding name from <binding name= in <ws2007FederationHttpBinding> to <binding name= in <SourceCode.Services.ws2007FederationHttpBinding>.
4. Add the following to the <ws2007FederationHttpBinding> section

```
<issuer
address="https://adfs.{ServerName}/adfs/services/trust/13/usernamemixed" />
```

5. Remove the `jsonOrXmlWebHttp` tag from

```
    <behaviors>
      <endpointBehaviors>
        <behavior name="SourceCode.Services.RestBehavior">
        </behavior>
        <behavior name="SourceCode.Services.SoapBehavior" />
      </endpointBehaviors>
```

6. Set <serviceMetadata httpGetEnabled=”false”> to true for both “SourceCode.Services.RestBehavior” and “SourceCode.Services.SoapBehavior”

```
      <behaviors>
        <serviceBehaviors>
          <behavior name="SourceCode.Services.RestBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
          </behavior>
          <behavior name="SourceCode.Services.SoapBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
          </behavior>
        </serviceBehaviors>
      </behaviors>
```

7. Copy the federatedServiceHostConfiguration:
   From <behavior name="SourceCode.Services.RestBehavior">
   To <behavior name="SourceCode.Services.SoapBehavior">
   and set the name attribute to “SourceCode.Services.Wcf”

```
<behavior name="SourceCode.Services.SoapBehavior">
          <federatedServiceHostConfiguration name="SourceCode.Services.Wcf"/>
          <serviceMetadata httpGetEnabled="true" />
          <serviceDebug includeExceptionDetailInFaults="true" />
```

8. Copy the <serviceCredentials>
   From <behavior name="SourceCode.Services.RestBehavior">
   To <behavior name="SourceCode.Services.SoapBehavior">

```
<serviceCredentials>
            <!--Certificate added by FedUtil.
Subject='CN={ServerName/HostHeader}, OU=IT, O={}, L=Redmond, S=WA, C=US',
Issuer='CN={}, DC={}, DC=com'.-->
            <serviceCertificate findValue="{Value}"
storeLocation="LocalMachine" storeName="My" x509FindType="FindByThumbprint"
/>
          </serviceCredentials>
```

9. Add saveBootstrapTokens="true" to <microsoft.identityModel> -> <service name="SourceCode.Services.Rest" >

```
    <microsoft.identityModel>
        <service name="SourceCode.Services.Rest" saveBootstrapTokens="true">
```

10. Change the audienceUri under <service name="SourceCode.Services.Rest" to

```
<add value="http://{ServerName/HostHeader}/K2Services/REST.svc" />
```

11. Remove the tag if not in development, but then make sure the signing ADFS certificate is trusted and 100%, else check MicrosoftIdentity SVC logging

```
<certificateValidation certificateValidationMode="None" />
```

12. Copy the complete <service name="SourceCode.Services.Rest"> section and paste it in the K2 Services web.config file then change the <service name= to "SourceCode.Services.Wcf" and the audienceUri to

```
<add value="http://{Server Name/Host Header}/K2Services/WCF.svc" />
```

13. If required these changes can be applied to the other K2 Services in the web.config located in C:\Program Files (x86)\K2 blackpearl\WebServices\K2Services.
14. Add the following to the K2HostServer.exe.Config - <configuration> section

```
<sourcecode.security.claims>
  <issuers>
    <issuer name="ADFS"
thumbprint="C4E84A89C565A6A49DF98ED672CEF5D3813DA5A7"/>
  </issuers>
  <claimTypeMappings>
    <claimTypeMapping securityLabel="K2">
      <identityProviderClaim originalIssuer="ADFS"
claimType="http://schemas.xmlsoap.org/claims/CommonName"
claimValue="ADFS" />
      <identityClaim originalIssuer="ADFS"
claimType="http://schemas.microsoft.com/ws/2008/06/identity/claims/window
saccountname" />
    </claimTypeMapping>
  </claimTypeMappings>
</sourcecode.security.claims>
```

NOTE: These are client specific. This is only an example of the claims used in this scenario. Make sure the thumbprint is correct. It should match the thumbprint found in the trustedissuers sections in the Microsoft.identitymodel section.

15. Make sure the following is in <configuration> -> <configSections>

```
<section name="sourcecode.security.claims"
type="SourceCode.Security.Claims.ClaimsConfigurationSectionHandler,
SourceCode.Security.Claims, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=16a2c5aaaa1b130d" />
```

Configure ADFS

ADFS now needs to be configured to be able to use the FederationMetadata.xml. Follow these steps:

1. Open ADFS 2.0 Management.
2. Add a new Relaying Party Trust under Trust Relationships.
3. Select to Import data from the relaying party from a file and select the generated FederationMetadata.xml from the following location C:\Program Files (x86)\K2 blackpearl\WebServices\K2Services\FederationMetadata\2007-06).
4. Specify a Display Name for example **K2WebServices**.
5. Permit all users to access this relaying party.
6. Click Next and select to create claim rules.

7. The claims that will be created is configured in K2HostServer.exe.config `<sourcecode.security.claims>` and should match.
8. View the properties of the created Relay party trust and replace the current identifier with the following identifiers:
   http://{ServerName/HostHeader}/K2Services/REST.svc
   http://{ServerName/HostHeader}/K2Services/WCF.svc
9. Run an IISRESET.

Configuring K2 SmartObject Services for Claims support

Follow the steps below to configure the K2 SmartObject Services for Claims support. Information on enabling and configuring SmartObject Services can be found here.

Generate FederationMetadata.xml

1. Run the FedUtil.exe from C:\Program Files (x86)\Windows Identity Foundation SDK\v4.0\FedUtil.exe
2. Select the K2HostServer.exe.Config from C:\Program Files (x86)\K2 blackpearl\Host Server\Bin\K2HostServer.exe.config
3. Specify http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Active Directory as the Application URI
4. Use an existing Secure Token Service, enter the **ADFS URL** and click on **Test location.**
5. Close the website once opened and click Next.
6. Select **Disable certificate chain validation.**
7. Enable Encryption and select the **Select an existing certificate** option. Select the required certificate.
8. Click Next to complete the process.

Configure SmartObject Web Services for Claims support

1. Modify the **K2HostServer.exe.config** located in C:\Program Files (x86)\K2 blackpearl\Host Server\Bin\K2HostServer.exe.config.
2. Change `enableEndpoints` to "`true`" in the `<configuration>` section

```
<smoServices enableEndpoints="true" enableEvents="true"
enableCrossDomainPolicy="false" specialCharacterReplacement="_"
scheme="http" server="{Server}" port="{Port}"
serviceRoot="SmartObjectServices">
```

3. Change **excluded all** to false under

```
<configuration>
        <smoServices>
          <managedEndpoints>
            <static>
              <endpoints />
            </static>
            <excluded all="true" />
          </managedEndpoints>
        </smoServices>
```

4. Edit the WCF and REST bindings as illustrated below

```
<wcf binding="ws2007FederationHttpBinding"
bindingConfiguration="ws2007FederationHttpBinding_WCF" />
<rest binding=" ws2007FederationHttpBinding"
bindingConfiguration="ws2007FederationHttpBinding_Rest" />
```

5. Edit the binding name in <ws2007FederationHttpBinding>to

```
<binding name="ws2007FederationHttpBinding_Config">
```

6. Add the following to the <ws2007FederationHttpBinding> section

```
<issuer
address="https://adfs.{ServerName/HostHeader}/adfs/services/trust/1
3/usernamemixed" />
```

7. Add saveBootstrapTokens="true" as indicated below

```
<microsoft.identityModel>
  <service saveBootstrapTokens="true">
    <audienceUris>
```

8. Add saveBootstrapTokens="true"to

```
<microsoft.identityModel>
  <service>
    <audienceUris>
      <add
value="http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Active
Directory" />
      <add value="
http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Exchange" />
      <add value="
http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Task
Allocation" />
      <add value="
http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Workflow
Reports/Workflow General" />
      <add value="
http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Active
Directory" />
      <add value="
http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Exchange" />
      <add value="
http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Task
Allocation" />
      <add value="
http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Workflow
Reports/Workflow General" />
```

9. Remove the tag if not in development, but then make sure the signing ADFS certificate is trusted and 100%, else check MicrosoftIdentity SVC logging

```
<certificateValidation certificateValidationMode="None" />
```

10. Restart the K2 server.
11. Ensure that the `<sourcecode.security.claims>` section exists in the `<configuration>` section of the K2HostServer.exe.Config, as described in the previous K2 Web Services section.
12. Update the certificate values (servicecertificate and trusted issuers) to client values.

Configure ADFS

ADFS now needs to be configured to be able to use the FederationMetadata.xml. Follow these steps:

1. Open ADFS Management.
2. Add a new Relaying Party Trust under Trust Relationships.
3. Select to Import data from the relaying party from a file and select the generated FederationMetadata.xml from the following location (C:\Program Files (x86)\K2 blackpearl\Host Server\Bin\FederationMetadata\2007-06)
4. Specify a Display Name for example K2SMOEndpoints.
5. Permit all users to access this relaying party.
6. Click next and choose to create claim rules.
7. The claims that will be created is configured in K2HostServer.exe.config `<sourcecode.security.claims>` and should match.
8. View the properties of the created Relay party trust and replace the current identifier with these identifiers:
   http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Active Directory
   http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Exchange
   http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Task Allocation
   http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Workflow Reports/Workflow General
   http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Active Directory
   http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Exchange
   http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Task Allocation
   http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Workflow Reports/Workflow General
9. Run an IISRESET.

Locate the endpoints

1. Open Internet Explorer and browse to http://{ServerName/HostHeader:Port}/SmartObjectServices/endpoints/endpoints.xml
2. The list of endpoints will display and services should be consumable in Visual Studio on:
   - http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Active Directory
   - http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Exchange
   - http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Task Allocation
   - http://{ServerName/HostHeader:Port}/SmartObjectServices/wcf/Workflow Reports/Workflow General
   - http://{ServerName/HostHeader:Port}/SmartObjectServices/rest/Active Directory/AD User/Create

Retrieve logs and diagnostics on WCF Services

To get logs and diagnostics on the WCF Services, add the following to the relevant .config file and adjust paths as needed:

```
<system.diagnostics>
  <sources>
    <source name="Microsoft.IdentityModel" switchValue="Verbose,ActivityTracing">
      <listeners>
        <add type="System.Diagnostics.DefaultTraceListener" name="Default">
          <filter type="" />
        </add>
        <add name="IdentityModelTraceListener">
          <filter type="" />
        </add>
      </listeners>
    </source>
    <source name="System.ServiceModel" switchValue="Verbose,ActivityTracing"
propagateActivity="true">
      <listeners>
        <add type="System.Diagnostics.DefaultTraceListener" name="Default">
          <filter type="" />
        </add>
        <add name="ServiceModelTraceListener">
          <filter type="" />
        </add>
      </listeners>
    </source>
    <source name="System.ServiceModel.MessageLogging" switchValue="Warning,
ActivityTracing">
      <listeners>
        <add type="System.Diagnostics.DefaultTraceListener" name="Default">
          <filter type="" />
        </add>
        <add name="ServiceModelMessageLoggingListener">
          <filter type="" />
        </add>
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add initializeData="C:\Program Files (x86)\K2
blackpearl\WebServices\K2Services\K2Web_identlog.svclog"
 type="System.Diagnostics.XmlWriterTraceListener, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
 name="IdentityModelTraceListener" traceOutputOptions="Timestamp">
      <filter type="" />
    </add>
    <add initializeData=" C:\Program Files (x86)\K2
blackpearl\WebServices\K2Services\K2Web_tracelog.svclog"
 type="System.Diagnostics.XmlWriterTraceListener, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
 name="ServiceModelTraceListener" traceOutputOptions="Timestamp">
        <filter type="" />
      </add>
      <add initializeData=" C:\Program Files (x86)\K2
blackpearl\WebServices\K2Services\K2Web_messages.svclog"
 type="System.Diagnostics.XmlWriterTraceListener, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
 name="ServiceModelMessageLoggingListener" traceOutputOptions="Timestamp">
        <filter type="" />
      </add>
    </sharedListeners>
  <trace autoflush="true" />
```

```
</system.diagnostics>
```

Troubleshooting

Below is a list of possible errors that might occur during this configuration.

1. ID3082: The request scope is not valid or is unsupported.
    a. Check URI's in webconfig and in ADFS 2.0 (ex http://{ServerName/HostHeader}/K2Services/WCF.svc)

2. 401 (Anonymous and NTLM,Negotiate)
    a. WebService\K2_SMO_Endpoint Rejecting it

3. FedUtils: ID1032 A WCF application federated to a security token service requires an application certificate
    a. Must supply a certificate

4. MSIS3127: The specified request failed
    a. ADFS Relay Party Trust not enabled

5. ID3242: The security token could not be authenticated or authorized
    a. // check microsoft.identitymodel SVC trace logs
    The X.509 certificate CN=ADFS Signing - adfs.{ServerName/HostHeader} is not in the trusted people store. The X.509 certificate CN=ADFS Signing - adfs.{ServerName/HostHeader} chain building failed. The certificate that was used has a trust chain that cannot be verified. Replace the certificate or change the certificateValidationMode. A certificate chain processed, but terminated in a root certificate which is not trusted by the trust provider.