Lab 2

Federated Identity

1. This lab extends the guidance in Chapter 4, "*Federated Identity for Web Applications*" and Chapter 6, "*Federated Identity with Multiple Partners*" of the book "*A Guide to Claims–based Identity and Access Control, 2nd Edition*" (<http://msdn.microsoft.com/en-us/library/ff423674.aspx>). It demonstrates how you can make an existing web application use federated identities to allow access to users who authenticate in other security realms. It also demonstrates how you can customize the home realm discovery process to allow users to select the identity provider they want to authenticate with.

# Objectives

After working though this lab, you will understand how claims can be used to implement federated identities in a Web application. You'll learn how to configure trust relationships with other identity providers, and how to control how users select the authentication provider to use. The final optional exercise in this lab shows you how to configure Microsoft Active Directory Federation Services (ADFS) to support a federated identity scenario.

* 1. **Note:** Before you start working with these exercises ensure you have run the dependency checking utility. You must also run Visual Studio as an administrator when opening the solutions in these exercises. If you simply double-click the solution file, Visual Studio may fail to load the projects. See the "*Introduction*" document for information about the dependency checker utility and how to run Visual Studio as an administrator.

This lab contains the following exercises:

* + [Exercise 1](#Ex01): **Federating Adatum and Litware**. In this exercise, you will modify the Adatum a-Order web application to trust the Adatum federation provider, and configure the Adatum federation provider to trust both the Adatum and Litware identity providers.
  + [Exercise 2](#Ex02): **Home Realm Discovery**. In this exercise, you will modify the a-Order web application to send a **whr** parameter to the federation provider. You will then modify the Adatum federation provider to use the value of the **whr** parameter to determine the identity provider the user should authenticate with.
  + [Exercise 3](#Ex03): **Federation with ADFS**. In this optional exercise, you will replace the custom Adatum federation provider with ADFS.

# Exercise 1: Federating Adatum and Litware

In this exercise you will modify the Adatum a-Order web application to trust the Adatum federation provider, and configure the Adatum federation provider to trust both the Adatum and Litware identity providers.

Trusts

Adatum

a-Order

Web application

Adatum

Federation

Provider

Adatum

Identity

Provider

 Litware

Identity

Provider

Trusts

Trusts

This exercise contains the following tasks:

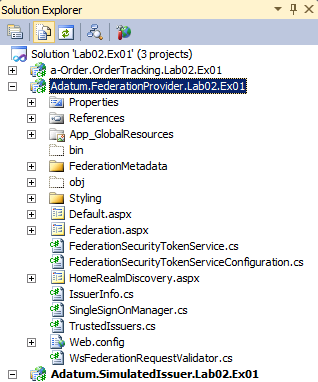
* + [Task 1](#Ex01Task01): Add the Adatum federation provider STS to the solution.
  + [Task 2](#Ex01Task02): Configure the a-Order application to trust the Adatum federation provider.
  + [Task 3](#Ex01Task03): Configure the Adatum federation provider to trust the Litware identity provider.
  + [Task 4](#Ex01Task04): Verify the solution.

You should be able to complete this exercise in approximately 20 minutes.

## Task 1: Add the Adatum federation provider STS to the solution

* 1. In this task you will add a mock Adatum federation provider to the solution and configure it to trust the mock Adatum identity provider. Adatum employees will be authenticated by the Adatum identity provider. This solution requires the federation provider because the a-Order application will also allow access to users from Litware.

To add the mock Adatum federation provider STS to the solution

* 1. In Windows Explorer, copy the **Adatum.FederationProvider.Lab02.Ex01** folder **(including all of its contents)** from the **Lab02-FederatedIdentity\Source\Ex01\Assets** folder to the **Lab02-FederatedIdentity\Source\Ex01\Begin** folder.
     1. The project in the **Adatum.FederationProvider.Lab02.Ex01** folder contains a sample STS that is configured to act as a federation provider — this project was created from the WIF **WCF Security Token Service** project template.
  2. Start Visual Studio as an administrator and open the solution named **Lab02.Ex01.sln** from the **Lab02-FederatedIdentity\Source\Ex01\Begin** folder.
  3. **In Visual Studio Solution Explorer, right-click on the top-level Lab02.Ex01 solution item, point to Add, and click Existing Project. Navigate to the folder Lab02-FederatedIdentity\Source\Ex01\Begin\ Adatum.FederationProvider.Lab02.Ex01, select the file named Adatum.FederationProvider.Lab02.Ex01.csproj, and click Open. The Adatum.FederationProvider.Lab02.Ex01** project now appears in Solution Explorer.
     1. 
  4. In Visual Studio, open the file **TrustedIssuers.cs** from the **Adatum.FederationProvider.Lab02.Ex01** project into the code editor window. Notice how this class defines an **IssuerInfo** object for the Adatum issuer that maps the Adatum home realm identifier to the Adatum identity provider's sign-in page.
     1. C#
     2. public static readonly IssuerInfo Adatum = new IssuerInfo(
     3. "adatum", "adatum.com", "https://localhost/Adatum.SimulatedIssuer.Lab02.Ex01/");
     4. public static readonly IssuerInfo Litware = new IssuerInfo(
     5. "litware", "litware.com", "https://localhost/Litware.SimulatedIssuer.Lab02.Ex01/");
     6. public static IEnumerable<IssuerInfo> GetAll()
     7. {
     8. return new IssuerInfo[] { Adatum, Litware };
     9. }
  5. In Visual Studio, open the file **HomeRealmDiscovery.aspx.cs** from the **Adatum.FederationProvider.Lab02.Ex01** project into the code editor window. In the **Onload** method, add an entry for Adatum to the **TrustedIssuersDropDownList** drop down list as shown in the following highlighted code.
     1. C#
     2. protected override void OnLoad(EventArgs e)
     3. {
     4. **if (!Page.IsPostBack)**
     5. **{**
     6. **this.TrustedIssuersDropDownList.Items.Add(**
     7. **new ListItem("Adatum", TrustedIssuers.Adatum.Realm));**
     8. **}**
     9. string action = this.Request.QueryString[WSFederationConstants.Parameters.Action];
  6. In Visual Studio, open the file **web.config** from the **Adatum.FederationProvider.Lab02.Ex01** project into the code editor window. Add the following highlighted thumbprint entity inside the **trustedIssuers** element. This thumbprint enables the Adatum federation provider to verify that a token is issued by the Adatum identity provider.
     1. XML
     2. <issuerNameRegistry
     3. type="Microsoft.IdentityModel.Tokens.ConfigurationBasedIssuerNameRegistry, …
     4. <trustedIssuers>
     5. **<add thumbprint="f260042d59e14817984c6183fbc6bfc71baf5462" name="adatum" />**
     6. </trustedIssuers>
     7. </issuerNameRegistry>
  7. Close all of the files open in the Visual Studio editor, making sure to save your changes, but keep the solution open so that you are ready to use it in the next task.
  8. You have now completed this task and you should know what changes you need to make to a federation provider created using the WIF template so that it allows a user to select the an identity provider and trusts an identity provider to issue tokens.

## Task 2: Configure the a-Order Application to Trust the Adatum Federation Provider

* 1. In this task you will configure the a-Order application (the relying party) to trust the mock Adatum federation provider. This will enable users to authenticate with the mock Adatum identity provider and use the claims issued by the providers to gain access to the a-Order application.

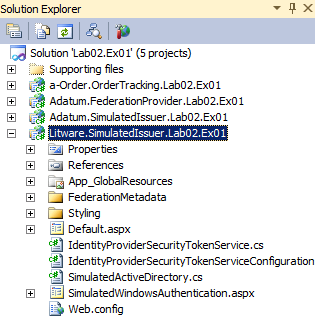
To configure the a-Order application to trust the Adatum Federation Provider

* 1. Continue with the **Lab02.Ex01** solution you used in the previous task.
  2. In Visual Studio, open the file **web.config** from the **a-Order.OrderTracking.Lab02.Ex01** project into the code editor window. Change the value of the **issuer** attribute of the **wsFederation** element to point to the new Adatum federation provider that you added in task 1. The highlighted code in the following snippet illustrates the change you need to make.
     1. XML
     2. <federatedAuthentication>
     3. <wsFederation passiveRedirectEnabled="true"
     4. issuer="**https://localhost/Adatum.FederationProvider.Lab02.Ex01/**"
     5. realm="https://localhost/a-Order.OrderTracking.Lab02.Ex01/" requireHttps="true" />
     6. <cookieHandler requireSsl="true" path="/a-Order.OrderTracking.Lab02.Ex01/" />
     7. </federatedAuthentication>
     8. In a real scenario you would also need to update the **trustedIssuers** section, but in this lab the Adatum mock federation provider and identity provider both use the same certificate.
  3. Close all of the files open in the Visual Studio editor, making sure to save your changes, but keep the solution open so that you are ready to use it in the next task.
  4. You have now completed this task and you should know what changes you need to make to a relying party web application to trust a federation provider.

## Task 3: Configure the Adatum Federation Provider to Trust the Litware Identity Provider

* 1. In this task you will modify the Adatum federation provider that you added to the solution in task 1 so that it trusts the Litware identity provider in addition to the Adatum identity provider. This will complete the federated identity scenario and enable users from both Adatum and Litware to access the a-Order application.

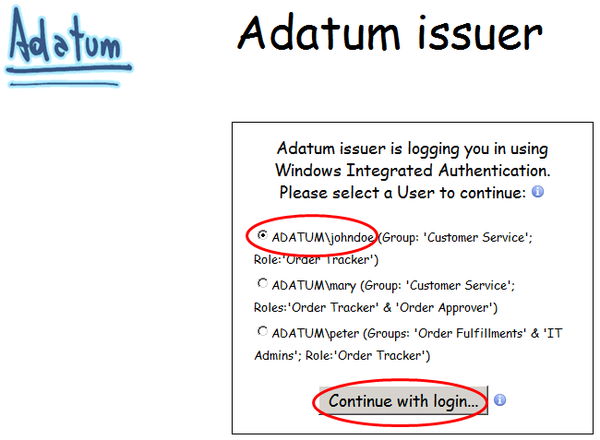
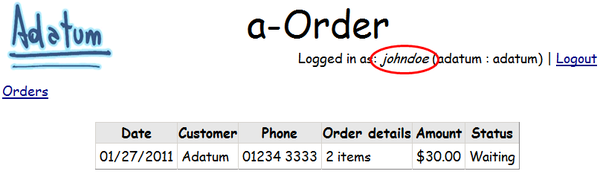
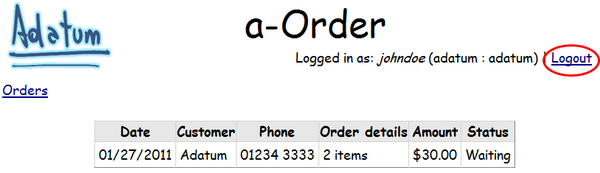
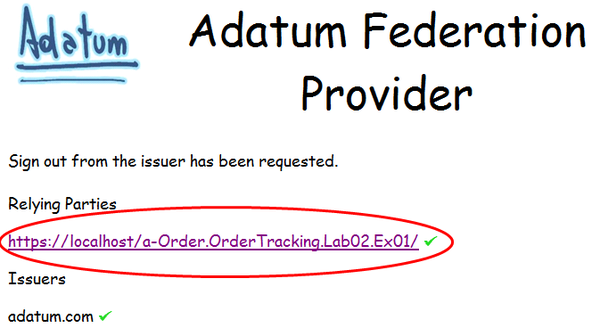
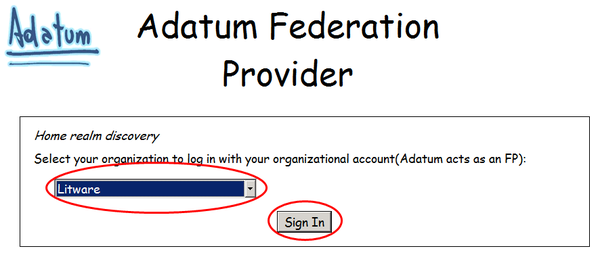
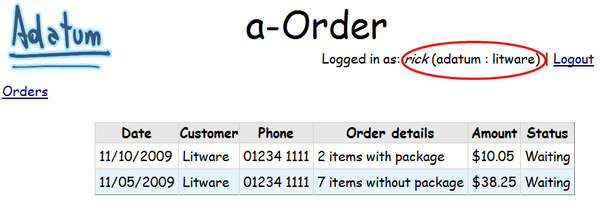
To configure the Adatum federation provider to trust the Litware identity provider

* 1. Continue with the **Lab02.Ex01** solution you used in the previous task.
  2. In Windows Explorer, copy the **Litware.SimulatedIssuer.Lab02.Ex01** folder **(including all of its contents)** from the **Lab02-FederatedIdentity\Source\Ex01\Assets** folder to the **Lab02-FederatedIdentity\Source\Ex01\Begin** folder.
  3. Continue with the **Lab02.Ex01** solution you used in the previous task.
  4. **In Visual Studio Solution Explorer, right-click on the top-level Lab02.Ex01 solution item, point to Add, and click Existing Project. Navigate to the folder Lab02-FederatedIdentity\Source\Ex01\Begin\ Litware.SimulatedIssuer.Lab02.Ex01, select the file named Litware.SimulatedIssuer.Lab02.Ex01.csproj, and click Open. The Litware.SimulatedIssuer.Lab02.Ex01** project now appears in Solution Explorer.
     1. 
  5. In Visual Studio, open the file **TrustedIssuers.cs** from the **Adatum.FederationProvider.Lab02.Ex01** project into the code editor window. Notice how this class defines an **IssuerInfo** object for the Litware issuer that maps the Litware home realm identifier to the Litware identity provider's sign-in page.
     1. C#
     2. public static readonly IssuerInfo Adatum = new IssuerInfo(
     3. "adatum", "adatum.com", "https://localhost/Adatum.SimulatedIssuer.Lab02.Ex02/");
     4. public static readonly IssuerInfo Litware = new IssuerInfo(
     5. "litware", "litware.com", "https://localhost/Litware.SimulatedIssuer.Lab02.Ex02/");
     7. public static IEnumerable<IssuerInfo> GetAll()
     8. {
     9. return new IssuerInfo[] { Adatum, Litware };
     10. }
  6. In Visual Studio, open the file **HomeRealmDiscovery.aspx.cs** from the **Adatum.FederationProvider.Lab02.Ex01** project into the code editor window. In the **Onload** method, add an entry for Litware to the **TrustedIssuersDropDownList** drop down list as shown in the following highlighted code.
     1. C#
     2. protected override void OnLoad(EventArgs e)
     3. {
     4. if (!Page.IsPostBack)
     5. {
     6. this.TrustedIssuersDropDownList.Items.Add(
     7. new ListItem("Adatum", TrustedIssuers.Adatum.Realm));
     8. **this.TrustedIssuersDropDownList.Items.Add(**
     9. **new ListItem("Litware", TrustedIssuers.Litware.Realm));**
     10. }
  7. In Visual Studio, open the file **web.config** from the **Adatum.FederationProvider.Lab02.Ex01** project into the code editor window. Add the following highlighted thumbprint entity inside the **trustedIssuers** element. This thumbprint enables the Adatum federation provider to verify that a token is issued by the Litware identity provider.
     1. XML
     2. <issuerNameRegistry
     3. type="Microsoft.IdentityModel.Tokens.ConfigurationBasedIssuerNameRegistry, …
     4. <trustedIssuers>
     5. <add thumbprint="f260042d59e14817984c6183fbc6bfc71baf5462" name="adatum" />
     6. **<add thumbprint="2f8206711633599286e10db94a5cfb0c5ec0a235" name="litware" />**
     7. </trustedIssuers>
     8. </issuerNameRegistry>
  8. In Solution Explorer, right-click the top-level **Lab02.Ex01** solution item and click **Rebuild solution**. Correct any errors.
  9. Close all of the files open in the Visual Studio editor, making sure to save your changes, but keep the solution open so that you are ready to use it in the next task.
  10. You have now completed this task. The changes that you made in this task were very similar to the changes you made in task 1 because you were adding a new trust relationship to the Adatum federation provider. When you add a new trust relationship to the federation provider, you don' t need to modify the relying party.

## Task 4: Verify the Solution

* 1. In this task you will verify the solution by logging first as a user from Adatum, and then as a user from Litware.

To verify the solution

* 1. Open your web browser and navigate to the URL <https://localhost/a-Order.OrderTracking.Lab02.Ex01/> to run the claims-aware a-Order application.
  2. Because you have not yet authenticated, the a-Order application redirects your browser to the Adatum federation provider where you can select your identity provider. Select **Adatum** in the drop down list box and then click the **Sign In** button.
     1. 
  3. The Adatum federation browser now redirects your browser to the Adatum identity provider. Select the **ADATUM\johndoe** option and then click the **Continue with login…** button to simulate signing in with Windows Integrated Authentication.
     1. 
  4. **You are now redirected back to the original page that you requested in the a-Order application which shows the data you requested and the fact that you are signed-in as johndoe. Notice that you only see Adatum orders.**
     1. 
  5. **Now you will try to access the same application using claims from the Litware identity provider. Click the Logout link to sign-out as johndoe.**
     1. 
  6. **On the sign-out page in the Adatum federation provider, click on the** <https://localhost/a-Order.OrderTracking.Lab02.Ex01/> **hyperlink to return to the a-Order application.**
     1. 
  7. **Because you are signed-out, you return to the home realm discovery page in the Adatum federation provider. In the drop-down list box, select Litware, and then click the Sign In button.**
     1. 
  8. **The Adatum federation provider now redirects you to the Litware identity provider. Click the Click here to continue… button to simulate signing in as Rick from Litware.**
     1. 
  9. **You are now redirected back to the original page that you requested in the a-Order application which shows the data you requested and the fact that you are signed-in as rick from Litware. Notice that you only see Litware orders.**
     1. 
  10. You have now completed this task and this exercise. You have seen how to configure a federated identity environment: users from Adatum and Litware can access the a-Order application using credentials from their own identity providers.
  11. Federated identity is a powerful feature of claims; users do not need to remember additional sets of credentials to access a partner's application, and it is easy for an organization such as Adatum to configure its federation provider to trust issuers from additional partners without the requirement to modify the relying party application.

## Running the "End" Solution

1. If you did not complete all of the tasks in this exercise, you can run the "end" solution we provide.

To run the end solution

* 1. Start Visual Studio 2010 as an administrator.
  2. Open the solution named **Lab02.Ex01.End** from the folder **Lab02-FederatedIdentity\Source\Ex01\End**.
  3. Open the **Build** menu and click **Rebuild Solution**.
  4. Open a web browser and navigate to the URL <https://localhost/a-Order.OrderTracking.Lab02.Ex01.End/> to run the a-Order application.
  5. On the home realm discovery page, in the drop-down list, select the **Adatum** issuer to login as a user from Adatum or the **Litware** issuer to login as a user from Litware and click **Continue with login**.

# Exercise 2: Home Realm Discovery

* 1. In this exercise you will modify the a-Order application to prompt unauthenticated users for their email address to use for home realm discovery. The a-Order application will use the domain name from the email address as the value of the **whr** parameter that it sends to the Adatum federation provider; the Adatum federation provider can use the **whr** parameter to determine the identity provider that can authenticate the user, and automatically redirect the browser to the correct identity provider.

This exercise contains the following tasks:

* + [Task 1](#_Task_1:_Modify): Modify the a-Order application to send the **whr** parameter to the federation provider.
  + [Task 2](#_Task_2:_Modify): Modify the Adatum federation provider to use the **whr** parameter.
  + [Task 3](#_Task_3:_Verify): Verify the solution.

You should be able to complete this exercise in approximately 20 minutes.

## Task 1: Modify the a-Order Application to Send the whr Parameter to the Federation Provider

* 1. In this task you will modify the a-Order application to prompt the user for an email address, extract the domain name from the email address, and send it to the Adatum federation provider in the **whr** parameter of the request.

To modify the a-Order application to send the whr parameter to the federation provider

* 1. Start Visual Studio as an administrator and open the solution named **Lab02.Ex02.sln** from the **Lab02-FederatedIdentity\Source\Ex02\Begin** folder.
  2. **In Visual Studio Solution Explorer, right-click on the a-Order.OrderTracking.Lab02.Ex02 project item, point to Add, and click New Item. In the Add New Item dialog, select Web Form, in the Name textbox type Index.aspx, and click Add.**
  3. In the code editor, add a TextBox control named **txtEmail**, a RegularExpressionValidator control to validate that the **txtEmail** control contains a valid email address, and a Button control named **btnSend**. Your new code should look like the highlighted code in the following sample:
     1. ASPX
     2. <body>
     3. <form id="form1" runat="server">
     4. <div>
     5. **<asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>**
     6. **<asp:RegularExpressionValidator ID="valEmailValidator" runat="server"**
     7. **ErrorMessage="Please enter a valid email address"**
     8. **ControlToValidate="txtEmail"**
     9. **ValidationExpression=**
     10. **"\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*">**
     11. **</asp:RegularExpressionValidator>**
     12. **<br />**
     13. **<asp:Button ID="btnSend"**
     14. **runat="server" Text="Send" onclick="txtSend\_Click" />**
     15. </div>
     16. </form>
     17. </body>
  4. In the code editor, switch to design view, and double click in the **btnSend** button to create an event handler for the click event.
  5. Add the highlighted code shown in the sample below to the event handler. This code parses the email address for the user's email domain, constructs a WS-Federation sign-in message, attaches the user's email domain as the **Realm** property of the sign-in message, and sends a redirect message back to the browser.
     1. C#
     2. using System;
     3. using System.Collections.Generic;
     4. using System.Linq;
     5. using System.Web;
     6. using System.Web.UI;
     7. using System.Web.UI.WebControls;
     8. **using Microsoft.IdentityModel.Web;**
     9. **using Microsoft.IdentityModel.Protocols.WSFederation;**
     10. namespace AOrder
     11. {
     12. public partial class Index : System.Web.UI.Page
     13. {
     14. protected void txtSend\_Click(object sender, EventArgs e)
     15. {
     16. **if (this.IsValid)**
     17. **{**
     18. **var homeRealm = this.txtEmail.Text.Split('@')[1];**
     19. **var issuerLocation =**
     20. **FederatedAuthentication.WSFederationAuthenticationModule.Issuer;**
     21. **var realm = FederatedAuthentication.WSFederationAuthenticationModule.Realm;**
     22. **var message = new SignInRequestMessage(new Uri(issuerLocation), realm);**
     23. **string encodedReplyUrl = HttpUtility.UrlEncode(realm);**
     24. **message.Context = string.Format("rm=0&id=passive&ru={0}", encodedReplyUrl);**
     25. **message.CurrentTime = DateTime.Now.ToUniversalTime().ToString(**
     26. **"yyyy'-'MM'-'dd'T'HH':'mm':'ssK");**
     27. **message.HomeRealm = homeRealm;**
     28. **this.Response.Redirect(message.RequestUrl, false);**
     29. **}**
     30. }
     31. }
     32. }
  6. In Solution Explorer expand the **a-Order.OrderTracking.Lab02.Ex02** project, and double-click on the file **Web.config** to open it in the code editor.
  7. Add a new **location** entry as shown in the highlighted section in the following code snippet. This entry enables users to browse to the new **Index.aspx** page without having to authenticate.
     1. XML
     2. </appSettings>
     3. **<location path="Index.aspx">**
     4. **<system.web>**
     5. **<authorization>**
     6. **<allow users="\*" />**
     7. **</authorization>**
     8. **</system.web>**
     9. **</location>**
     10. <location path="FederationMetadata">
  8. Close all of the files open in the Visual Studio editor, making sure to save your changes, but keep the solution open so that you are ready to use it in the next task.
  9. You have now completed this task. The a-Order application now has a new page called Index.aspx that prompts a user for an email address. The page uses the host domain of the email address as the user's home realm and sends this value in the **whr** parameter to the Adatum federation provider. You have also modified the Web.config file in the a-Order application to allow users to visit the new page without authenticating.

## Task 2: Modify the Adatum Federation Provider to Use the whr Parameter

* 1. In this task you will modify the Adatum federation provider to use the **whr** parameter send from the relying party (the a-Order application). The federation provider will use the home realm information in the **whr** parameter to redirect the browser to the correct identity provider for the user.

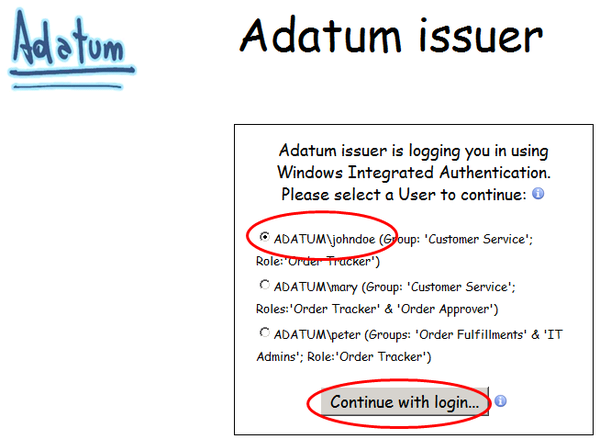
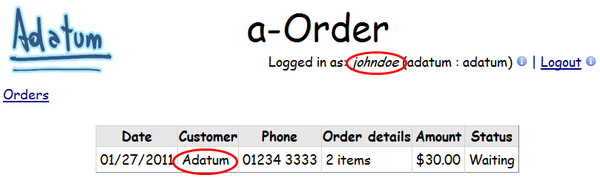
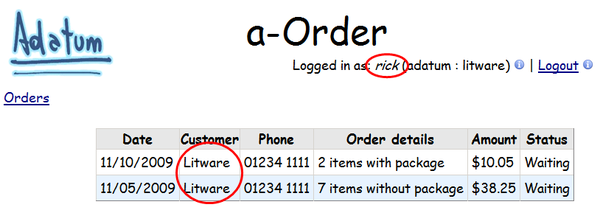
Modify the Adatum federation provider to use the whr parameter

* 1. In Solution Explorer expand the **Adatum.FederationProvider.Lab02.Ex02** project, right-click on the file **HomeRealmDiscovery.aspx**, and click on **View Code** to open it in the code editor.
  2. Add the highlighted code in the following sample to the end of the **OnLoad** method to read and validate the value of the **whr** parameter. The home realm value is valid, redirect the browser to the Federation.aspx page.
     1. C#
     2. base.OnLoad(e);
     3. **var whr = Request.Params["whr"];**
     4. **if (!string.IsNullOrEmpty(whr))**
     5. **{**
     6. **if (TrustedIssuers.IsValidRealm(whr))**
     7. **{**
     8. **this.Response.Redirect("~/Federation.aspx?" + this.Request.QueryString, false);**
     9. **}**
     10. **}**
     11. }
  3. Replace the existing code in **ProcessSignIn** method with the highlighted code in the following sample. If the value of the **whr** is not recognized, this code replaces the **whr** parameter value with the realm that the user selects in the dropdown list box.
     1. C#
     2. public void ProcessSignIn()
     3. {
     4. **var selectedWhr = HttpUtility.UrlEncode(this.TrustedIssuersDropDownList.Text);**
     5. **string queryString = this.Request.QueryString.ToString();**
     6. **// if there is a whr parameter replace it with the user selection**
     7. **if (!string.IsNullOrEmpty(this.Request.QueryString["whr"]))**
     8. **{**
     9. **var whrToReplace = this.Request.QueryString["whr"];**
     10. **queryString = queryString.Replace(whrToReplace, selectedWhr);**
     11. **}**
     12. **else**
     13. **{**
     14. **queryString += "&whr=" + selectedWhr;**
     15. **}**
     16. **this.Response.Redirect("~/Federation.aspx?" + queryString, false);**
     17. }
  4. In Solution Explorer, right-click the top-level **Lab02.Ex02** solution item and click **Rebuild solution**. Correct any errors.
  5. Close all of the files open in the Visual Studio editor, making sure to save your changes, but keep the solution open so that you are ready to use it in the next task.
  6. You have now completed this task. The Adatum federation provider will now try to use the home realm information sent from the relying party application to automatically redirect the browser to the identity provider that can authenticate the user.

## Task 3: Verify the Solution

* 1. In this task, you will verify the solution by entering an email address in the Index.aspx page in the a-Order application. If the email domain is either Litware.com or Adatum.com, you will be redirected to the identity provider's sign-in page. Otherwise, you will see the home realm discovery page in the Adatum federation provider.

To verify the solution

* 1. Open your web browser and navigate to the URL <https://localhost/a-Order.OrderTracking.Lab02.Ex02/Index.aspx> to run the claims-aware a-Order application.
  2. On the Index.aspx page, in the textbox, enter **john@adatum.com**, and click the **Send** button.
     1. 
  3. You will be redirected to the Adatum issuer where you can sign-in as **johndoe**.
     1. 
  4. You will see the Adatum data in the a-Order application, and be signed in as **johndoe**.
     1. 
  5. Click the **Logout** hyperlink.
  6. In your web browser, navigate to the URL <https://localhost/a-Order.OrderTracking.Lab02.Ex02/Index.aspx> again, in the textbox enter **jane@litware.com**, and then click the **Send** button.
     1. 
  7. Notice that only the email domain name has been passed to the Litware identity provider. Click on **Click here to continue** to sign in as Rick.
     1. 
  8. You are now logged into the a-Order application as Rick from Litware.
     1. 
  9. Click the **Logout** hyperlink.
  10. In your web browser, navigate to the URL <https://localhost/a-Order.OrderTracking.Lab02.Ex02/Index.aspx> again, in the textbox enter **bill@contoso.com**, and then click the **Send** button.
      1. 
  11. Because Contoso.com is not a valid realm, you are redirected to the Adatum home realm discovery page to select your identity provider.
      1. 
  12. You have now completed this task and verified that the realm passed from the Index.aspx page in the a-Order application is used by the Adatum federation provider to redirect the user to the correct identity provider. If you enter an email address with an unrecognized email domain, you are redirected to the home realm discovery page instead.
  13. You can use home realm discovery to improve the user experience. If you can determine the user's security realm in the relying party application, you can pass this information to the federation provider so the user is redirected directly to the identity provider, bypassing the home realm discovery page.

## Running the "End" Solution

1. If you did not complete all of the tasks in this exercise, you can run the "end" solution we provide.

To run the end solution

* 1. Start Visual Studio 2010 as an administrator.
  2. Open the solution named **Lab02.Ex02.End** from the folder **Lab02-FederatedIdentity\Source\Ex02\End**.
  3. Open the **Build** menu and click **Rebuild Solution**.
  4. Open a web browser and navigate to the URL <https://localhost/a-Order.OrderTracking.Lab02.Ex02.End/Index.aspx> to run the a-Order application.
  5. If you enter an email address with a domain of either Adatum.com or Litware.com, you will be redirected straight to the correct identity provider. If you enter an email address with any other domain, you will be redirected to the home realm discovery page in the Adatum federation provider.

# Exercise 3: Federation with ADFS

* 1. In this optional exercise, you will replace the mock Adatum federation provider and identity provider projects in the solution and configure ADFS to perform the same roles. This will be a more realistic scenario. The changes to the a-Order application will be just configuration changes so that it points to your ADFS installation instead of the mock issuers in the Visual Studio solution. The remainder of the Lab will show you how to configure ADFS to support the Adatum federated identity scenario.
  2. The Lab Introduction contains instructions for installing and configuring a base ADFS environment. You should ensure that you have read the section, "Information Required for the Lab Exercises," in the Introduction before attempting this Lab Exercise.

This exercise contains the following tasks:

* + [Task 1](#_Task_1:_Modify_1): Add the test user to Active Directory
  + [Task 2](#_Task_2:_Modify_1): Modify the a-Order application to trust ADFS
  + [Task 3](#_Task_2:_Add): Add the Adatum claims rules to ADFS
  + [Task 4](#_Task_3:_Add): Add the a-Order application as a Relying Party in ADFS
  + [Task 5](#_Task_4:_Add): Add a-Order application claims rules to ADFS
  + [Task 6](#_Task_5:_Add): Add the Litware issuer as Claim Trust Provider in ADFS
  + [Task 7](#_Task_7:_Add): Add Litware claims rules to ADFS
  + [Task 8](#_Task_8:_Modify): Modify the Litware issuer
  + [Task 9](#_Task_9:_Verify): Verify the solution

You should be able to complete this exercise in approximately 20 minutes.

## Task 1: Add the test user to Active Directory

* 1. In this task, you will add a test user, John Doe, to Active Directory (AD) to use with the a-Order application. The AD attributes will be passed through to the a-Order application by ADFS as claims.
  2. If you have completed Lab 1, you will have already added this user. If this is the case, you can skip this task.

To add the test user to Active Directory

* 1. On the machine where you have installed Active Directory for use in this Lab, start Active Directory Users and Computers.
  2. Add a new user in your domain with the attributes listed in the following table.

|  |  |
| --- | --- |
| Attribute | Value |
| Display name | johndoe |
| User logon name | johndoe |
| Password | Pa$$w0rd |
| User must change password at next logon | False |

* 1. You have now completed this task to add a new test user called John Doe to Active Directory.

## Task 2: Modify the a-Order Application to Trust ADFS

* 1. In this task, you will modify the a-Order application to trust your ADFS issuer instead of the mock Adatum federation provider. You will also remove the Adatum federation provider and simulated issuer projects from the solution as they are no longer required and will be replaced by ADFS.

To modify the a-Order application to trust ADFS

* 1. Start Visual Studio as an administrator and open the solution named **Lab02.Ex03.sln** from the **Lab02-FederatedIdentity\Source\Ex03\Begin** folder.
  2. In Solution Explorer, right-click on the **Adatum.SimulatedIssuer.Lab02.Ex03** project node, and select **Remove**. Then click **OK**.
  3. In Solution Explorer, right-click on the **Adatum.FederationProvider.Lab02.Ex03** project node, and select **Remove**. Then click **OK**.
  4. In Solution Explorer expand the **a-Order.OrderTracking.Lab02.Ex03** project, and double-click on the file **Web.config** to open it in the code editor.
  5. Using the values for the identifier and thumbprint of your ADFS issuer that you recorded when you read the section "Integrating With Active Directory Federation Services" in the Introduction to these Lab exercises, modify the highlighted values in the **trustedIssuers** section as shown in the following snippet:
     1. XML
     2. <trustedIssuers>
     3. <add thumbprint="{**your-thumbprint}**"
     4. name="{**your-name-identifier}**" />
     5. </trustedIssuers>
  6. Using the value for the issuer address in your ADFS instance that you recorded when you read the section "Integrating With Active Directory Federation Services" in the Introduction to these Lab exercises, modify the **issuer** attribute of the **wsFederation** element as shown in the following snippet:
     1. XML
     2. <federatedAuthentication>

<wsFederation passiveRedirectEnabled="true"

issuer="**https://{issuer-url}/adfs/ls/**"

realm="https://localhost/a-Order.OrderTracking.Lab02.Ex03/"

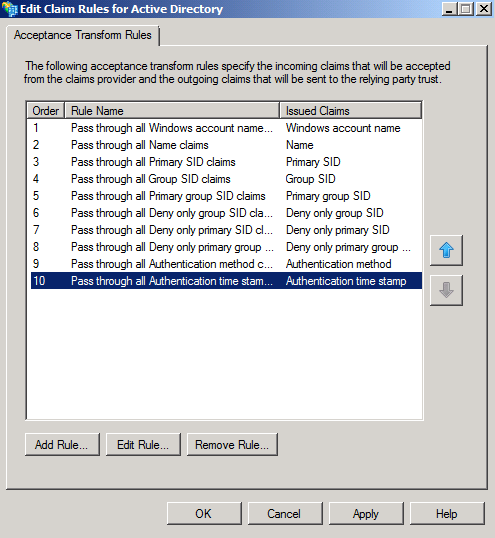
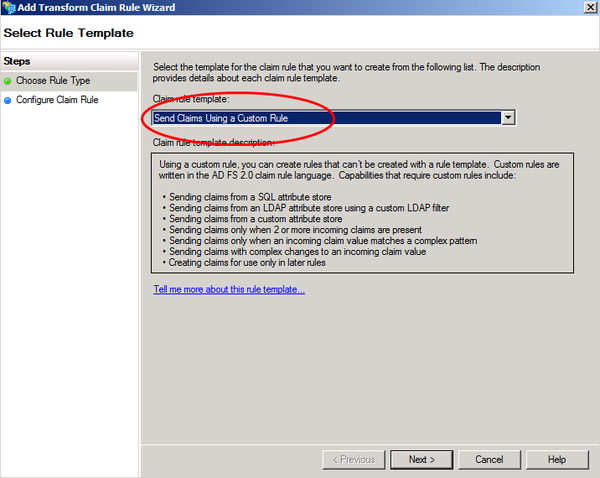
requireHttps="true" />

* + 1. <cookieHandler requireSsl="true" path="/a-Order.OrderTracking.Lab02.Ex03/" />
    2. </federatedAuthentication>
  1. Close all of the files open in the Visual Studio editor, making sure to save your changes, but keep the solution open so that you are ready to use it in the next task.
  2. You have now completed this task to configure the a-Order application to trust your ADFS instance to issue claims instead of using the mock federation provider.

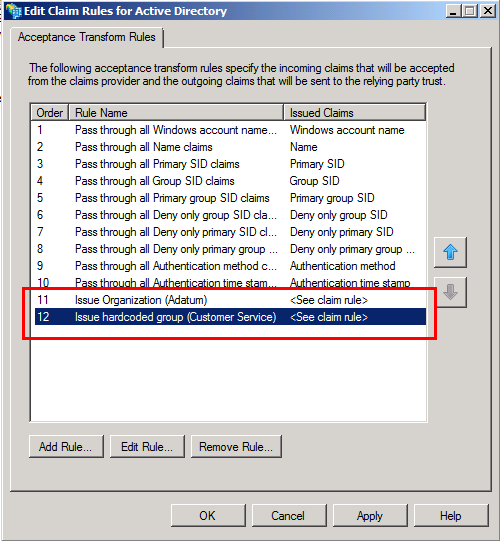
## Task 3: Add the Adatum Claims Rules to ADFS

* 1. In this task you will modify the Active Directory claims provider trust in ADFS to issue some sample claims to work with the a-Order application. You will use the AD FS 2.0 Management tool to add the rules that you can use to test the a-Order application.
  2. If you have completed Lab 1, you will have already added these rules. If this is the case, you can skip this task.

To add the Adatum claims rules to ADFS

* 1. On the machine where you have installed ADFS, start the AD FS 2.0 Management tool.
  2. Expand **Trust Relationships**, then click on **Claims Provider Trusts**, then right-click on **Active Directory**, and then select **Edit Claim Rules**.
     1. 
  3. Now you can add the additional rules that you need to test the scenario. In the **Edit Claim Rules for Active Directory** dialog, click **Add Rule.** Then, in **Add Transform Claim Rule Wizard** dialog, in the **Claim rule template** dropdown list, select **Send Claims Using a Custom Rule**, and then click **Next**.
     1. 
  4. On the **Configure Rule** page of the wizard, in the **Claim rule name** textbox, type **Issue Organization (Adatum)**. In the **Custom rule** textbox, add the following rule:
     1. => issue(Type = "http://schemas.adatum.com/claims/2009/08/organization", Value = "Adatum");
  5. On the **Configure Rule** page of the wizard, click **Finish**.
  6. Using the information in the following table, repeat steps 3, 4 and 5 to add the Group rule to ADFS.

|  |  |
| --- | --- |
| Claim rule name | Custom rule |
| Issue hardcoded group (Customer Service) | => issue(Type = "http://schemas.xmlsoap.org/claims/group", Value = "Customer Service"); |

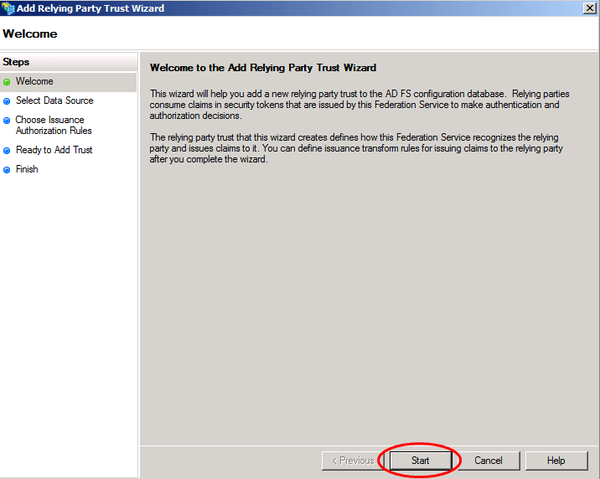
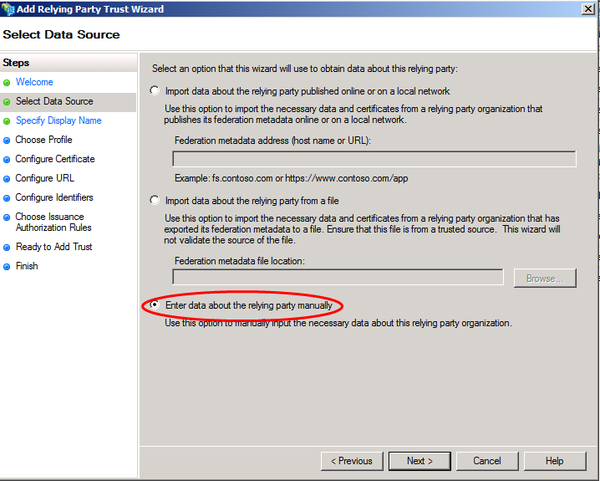
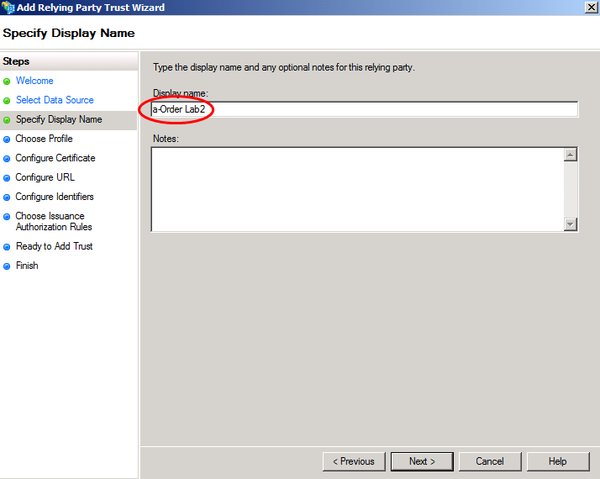
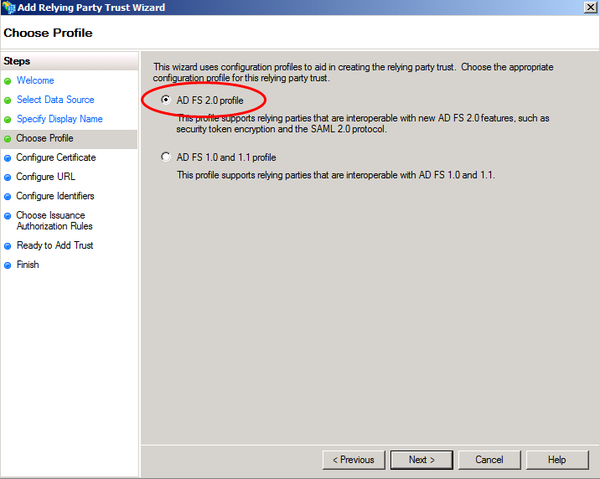
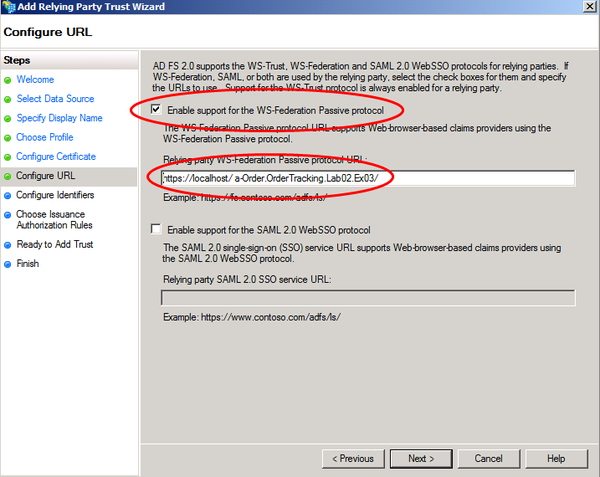
* 1. When you have finished, the **Edit Claim Rules for Active Directory** dialog should look like the following screenshot. Click **OK** to close the dialog.
     1. 

You have now completed this task to add the claims rules to the Active Directory claims provider trust in ADFS that you can use to test the a-Order application.

## Task 4: Add the a-Order Application as a Relying Party in ADFS

* 1. In this task, you will modify the ADFS configuration to define the a-Order application as a relying party.

To add the a-Order application as a Relying Party in ADFS

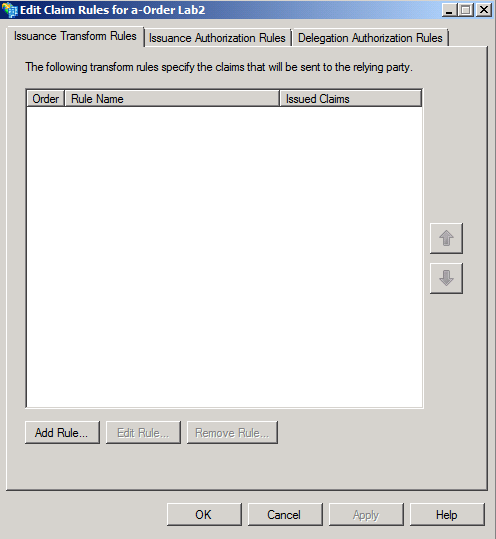
* 1. On the machine where you have installed ADFS, start the AD FS 2.0 Management tool.
  2. Expand **Trust Relationships**, then right-click on **Relying Party Trusts**, and then select **Add Relying Party Trust**.
  3. In the **Add Relying Party Trust Wizard** click **Start**.
     1. 
  4. On the **Select Data Source** page of the wizard, select **Enter data about the relying party manually** and click **Next**.
     1. 
  5. On the **Specify Display Name** page of the wizard, type **a-Order Lab2** in the **Display name** textbox, then click **Next**.
     1. 
  6. On the **Choose Profile** page of the wizard, ensure that the **AD FS 2.0 profile** option is selected, and then click **Next**.
     1. 
  7. On the **Configure Certificate** page of the wizard, click **Next**.
  8. On the **Configure URL** page of the wizard, ensure that **Enable support for the WS-Federation Passive protocol** is checked, in the **Relying party WS-Federation Passive protocol URL** text box type **https://localhost/a-Order.OrderTracking.Lab02.Ex03/**, and then click **Next**.
     1. 
  9. On the **Configure Identifiers** page of the wizard, click **Next**.
  10. On the **Choose Issuance Authorization Rules** page of the wizard, click **Next**.
  11. On the **Ready to Add Trust** page of the wizard, click **Next**.
  12. On the **Finish** page of the wizard, click **Close**. The **Edit Claim Rules for a-Order Lab2** dialog will open ready for you to complete the next task.

You have now completed this task to define the a-Order application as a relying party in ADFS.

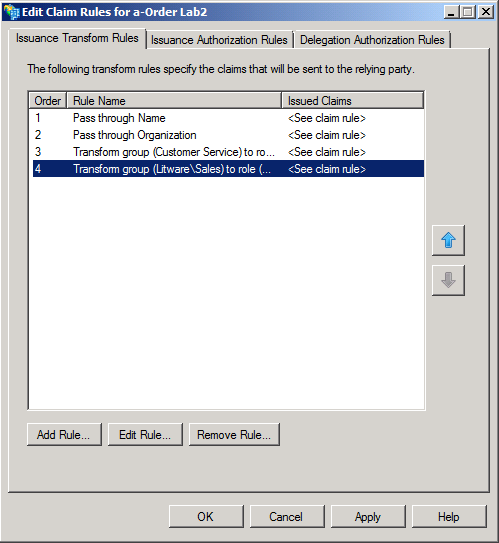
## Task 5: Add a-Order Application Claims Rules to ADFS

* 1. In this task, you will add the claims transformation rules to the **a-Order Lab2** relying party trust in ADFS. These rules will map the claims from the **Active Directory** claims provider trust to a set of claims that the a-Order application will recognize.

To add a-Order application claims rules to ADFS

* 1. If you do not have the **Edit Claim Rules for a-Order Lab2** dialog open from the previous task, then on the machine where you have installed ADFS, start the AD FS 2.0 Management tool. Expand **Trust Relationships**, then click on **Relying Party Trusts**, then right-click on **a-Order Lab2**, and then select **Edit Claim Rules**. You should be able to see the dialog shown in the following screenshot:
     1. 
  2. Now you can add the claim transformation rules you need to test the scenario. In the **Edit Claim Rules for a-Order Lab2** dialog, click **Add Rule.** Then, in **Add Transform Claim Rule Wizard** dialog, in the **Claim rule template** dropdown list, select **Send Claims Using a Custom Rule**, and then click **Next**.
  3. On the **Configure Rule** page of the wizard, in the **Claim rule name** textbox, type **Pass through Name**. In the **Custom rule** textbox, add the following rule:
     1. c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"] => issue(claim = c);
  4. On the **Configure Rule** page of the wizard, click **Finish**.
  5. Using the information in the following table, repeat steps 2, 3, and 4 to add the remaining rules to ADFS.

|  |  |
| --- | --- |
| Claim rule name | Custom rule |
| Pass through Organization | c:[Type == "http://schemas.adatum.com/claims/2009/08/organization"] => issue (claim = c); |
| Transform group (Customer Service) to role (Order Tracker) | c:[Type == "http://schemas.xmlsoap.org/claims/group", Value =~ "^(?i)Customer\ Service$"]  => issue(Type = "http://schemas.microsoft.com/ws/2008/06/identity/claims/role", Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = "Order Tracker", ValueType = c.ValueType); |
| Transform group (Litware\Sales) to role (Order Tracker) | c:[Type == "http://schemas.xmlsoap.org/claims/group", Value =~ "^(?i)Litware\\Sales$"]  => issue(Type = "http://schemas.microsoft.com/ws/2008/06/identity/claims/role", Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = "Order Tracker", ValueType = c.ValueType); |

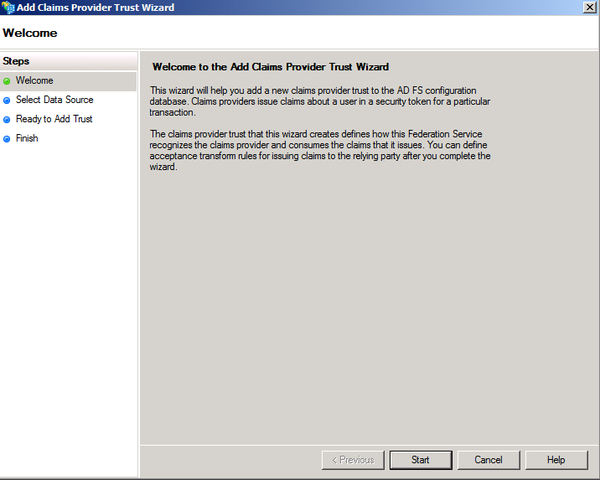
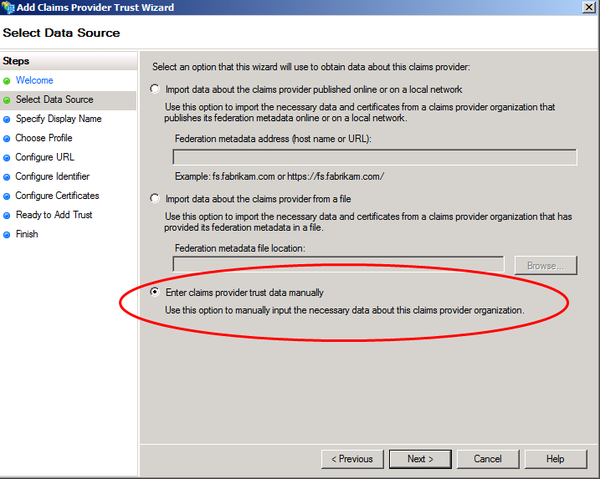
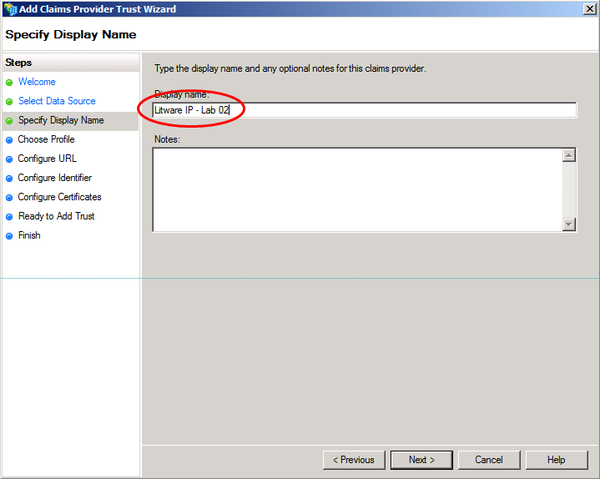
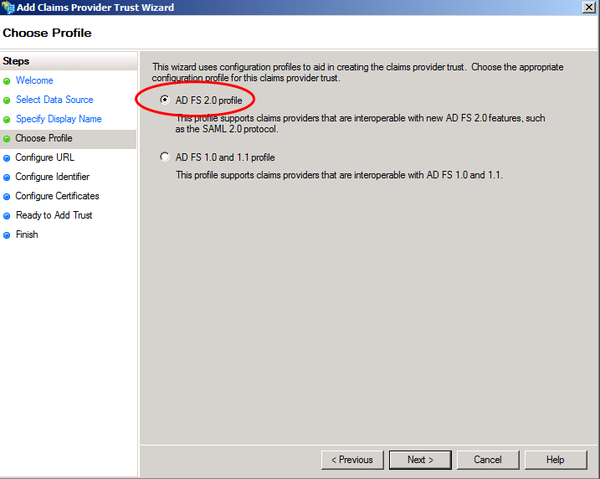
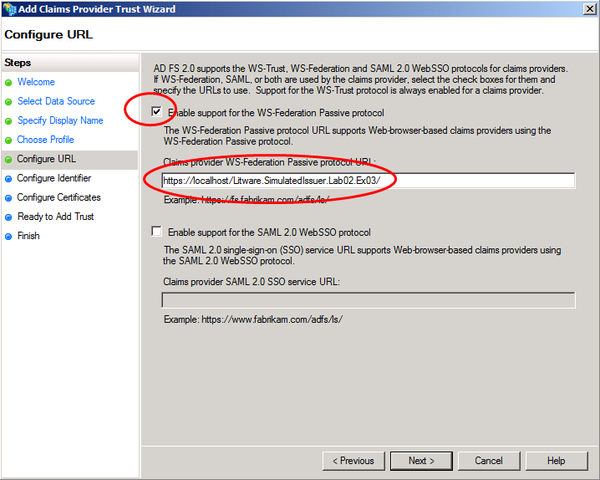
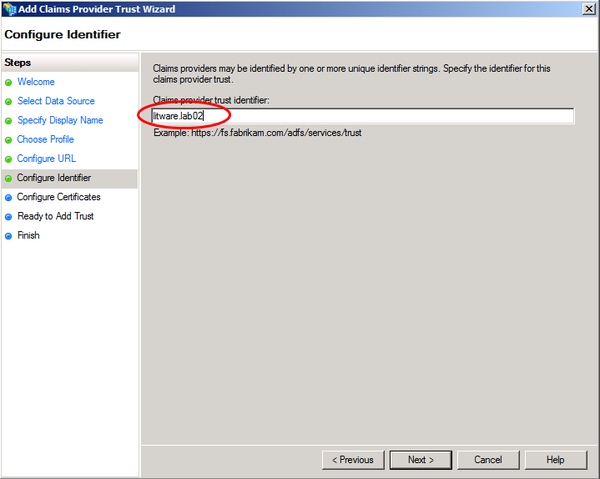
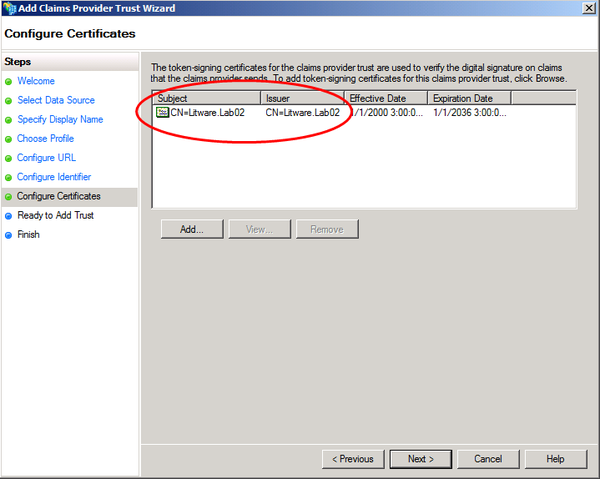
* 1. When you have finished, the **Edit Claim Rules for a-Order Lab2** dialog should look like the following screenshot. Click **OK** to close the dialog.
     1. 

You have now completed this task and added a set of claim transformation rules for the a-Order relying party application in ADFS.

## Task 6: Add the Litware Issuer as Claim Trust Provider in ADFS

* 1. In this task, you will add a claims provider trust for the Litware IP to the ADFS configuration. This will enable users to authenticate with Litware to gain access to the a-Order application.

To add the Litware issuer as Claim Trust Provider in ADFS

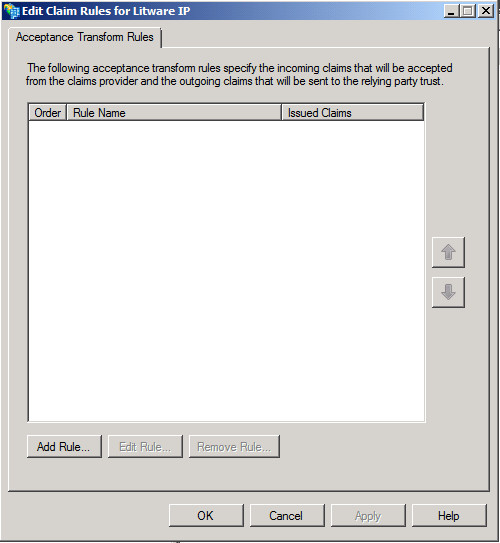
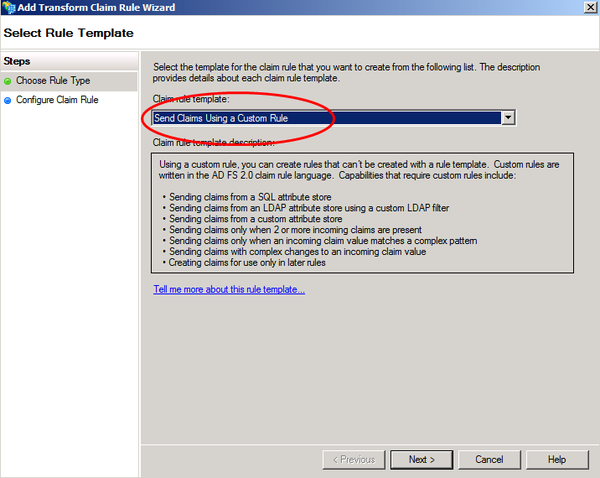
* 1. On the machine where you have installed ADFS, start the AD FS 2.0 Management tool. Expand **Trust Relationships**, then right-click on **Claims Provider Trusts**, and then select **Add Claims Provider Trust**. The **Add Claims Provider Trust Wizard** will launch. Click **Start**.
     1. 
  2. On the **Select Data Source** page of the wizard, click **Enter claims provider trust data manually**, and then click **Next**.
     1. 
  3. On the **Specify Display Name** page, in **Display name** type **Litware IP – Lab 02**. Then click **Next**.
     1. 
  4. On the **Choose Profile** page, select **AD FS 2.0 Profile**, and then click **Next**.
     1. 
  5. On the **Configure URL** page, select the **Enable support for the WS-Federation Passive** protocol check box. Under **Claims provider WS-Federation Passive protocol URL**, type **https://localhost/Litware.SimulatedIssuer.Lab02.Ex03/**, and then click **Next**.
     1. 
  6. On the **Configure Identifier** page, under **Claims provider trust identifier**, type **litware.lab02**, and then click **Next**.
     1. 
  7. On the **Configure Certificates** page, click **Add** to locate the **litware.lab02.cer** file and add it to the list of certificates, and then click **Next**. If you receive a message, "**The certificate key length is less than 2048 bits. Certificates with key sizes less than 2048 bits might present security risk and are not recommended. Do you want to continue?**" click **Yes** to continue.
     1. You can find the **litware.lab02.cer** file in the **\Lab02-FederatedIdentity\Source\Ex03\Assets** folder. If ADFS is installed on a different machine from where the Lab files are located, you will need to copy the file from the machine where the Lab source files are located to the machine where you have ADFS installed.
     2. This is a test certificate — a certificate for use in production should have a key-size of 2048 bits or more.
     3. 
  8. On the **Ready to Add Trust** page, click **Next** to save your claims provider trust information.
  9. On the **Finish** page, click **Close**. This action automatically displays the **Edit Claim Rules** dialog box that you will use in the next task.

You have now completed this task to configure the Litware issuer as a claims provider in ADFS.

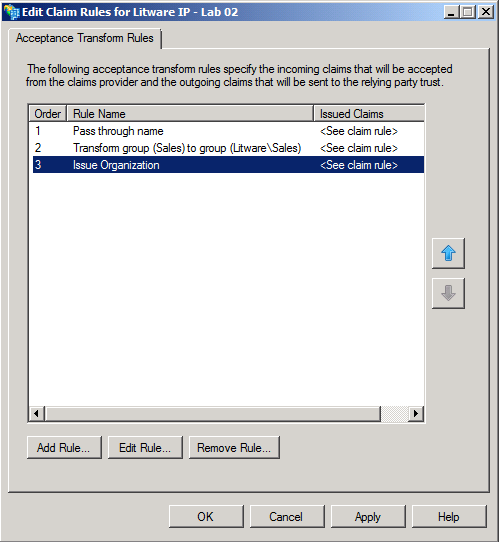
## Task 7: Add Litware Claims Rules to ADFS

* 1. In this task, you will modify the Litware IP claims provider trust in ADFS to accept claims from the Litware issuer. You will use the AD FS 2.0 Management tool to add the rules that you can use to test the a-Order application.

To add Litware claims rules to ADFS

* 1. If you do not have the **Edit Claim Rules for Litware IP** dialog open from the previous task, then on the machine where you have installed ADFS, start the AD FS 2.0 Management tool. Expand **Trust Relationships**, then click on **Claims Provider Trusts**, then right-click on **Litware IP – Lab 02**, and then select **Edit Claim Rules**. You should be able to see the dialog shown in the following screenshot.
     1. 
  2. Now you can add the rules you need to test the scenario. In the **Edit Claim Rules for Litware IP** dialog, click **Add Rule.** Then, in **Add Transform Claim Rule Wizard** dialog, in the **Claim rule template** dropdown list, select **Send Claims Using a Custom Rule**, and then click **Next**.
     1. 
  3. On the **Configure Rule** page of the wizard, in the **Claim rule name** textbox, type **Pass through Name**. In the **Custom rule** textbox, add the following rule:
     1. c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"] => issue (claim = c);
  4. On the **Configure Rule** page of the wizard, click **Finish**.
  5. Using the information in the following table, repeat steps 2, 3 and 4 to add the remaining rules to ADFS.

|  |  |
| --- | --- |
| Claim rule name | Custom rule |
| Transform group (Sales) to group (Litware\Sales) | c:[Type == "http://schemas.xmlsoap.org/claims/group", Value =~ "^(?i)Sales$"]  => issue(Type = "http://schemas.xmlsoap.org/claims/group", Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = "Litware\Sales", ValueType = c.ValueType); |
| Issue Organization | => issue(Type = "http://schemas.adatum.com/claims/2009/08/organization", Value = "Litware"); |

* 1. When you have finished, the **Edit Claim Rules for Litware IP – Lab02** dialog should look like the following screenshot. Click **OK** to close the dialog.
     1. 

You have now completed this task to add the Litware IP claims rules to the Litware IP claims provider trust in ADFS that you can use to test the a-Order application.

## Task 8: Modify the Litware Issuer

* 1. In this task, you will modify the Litware issuer to work with your ADFS instance.

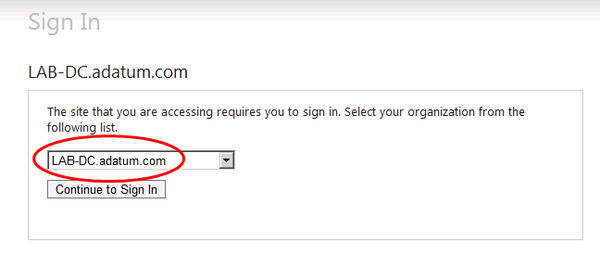
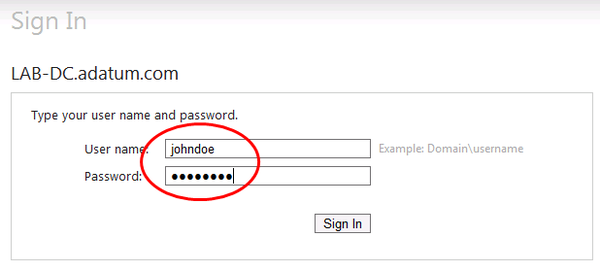
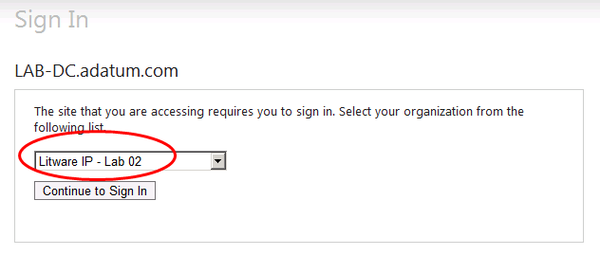
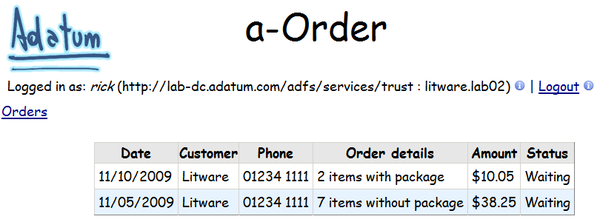
To modify the Litware Issuer

* 1. Start Visual Studio as an administrator and open the solution named **Lab02.Ex03.sln** from the **Lab02-FederatedIdentity\Source\Ex03\Begin** folder.
  2. In Visual Studio, open the file **IdentityProviderSecurityTokenService.cs** from the **Litware.SimulatedIssuer.Lab02.Ex03** project into the code editor window.
  3. Modify the constructor to add details of your ADFS instance to the dictionary. Modify the following highlighted code, to replace the **{issuer-url}** with the host name of your ADFS server.
     1. C#
     2. public IdentityProviderSecurityTokenService(
     3. SecurityTokenServiceConfiguration configuration) : base(configuration)
     4. {
     5. this.RelyingParties = new Dictionary<string, string>();
     6. this.RelyingParties.Add(
     7. "http://**{issuer-url}**/adfs/services/trust", "https://**{issuer-url}**/adfs/ls/");
     8. }
     9. You should enter the **hostname** in all lower-case characters.
  4. In Solution Explorer, right-click the top-level **Lab02.Ex03** solution item and click **Rebuild solution**. Correct any errors.
  5. Close all of the files open in the Visual Studio editor, making sure to save your changes, but keep the solution open so that you are ready to use it in the next task.
  6. You have now completed this task to modify the Litware issuer to work with your ADFS instance.

## Task 9: Verify the Solution

* 1. In this task you will verify the solution by visiting the a-Order application, authenticating first as John Doe from Adatum, and then as Rick from Litware.
  2. ADFS uses a cookie to record your preferred identity provider: to avoid being "locked" to one identity provider during testing you should use your browser's **InPrivate** browsing feature that ignores all persistent cookies. In Internet Explorer 9, you can activate **InPrivate** browsing by clicking the **Tools** icon, then pointing to **Safety**, and then click **InPrivate Browsing**. In Internet Explorer 8, click **Safety** on the toolbar, and then click **InPrivate Browsing**.
  3. If you are running this Lab on the same machine as the one where you have installed ADFS, then you will receive a warning message "The security certificate presented by this website was issued for a different website's address" when your browser is redirected to ADFS. You should click **Continue to this website** to continue.

To verify the solution

* 1. Open your web browser, activate **InPrivate** browsing, and navigate to the URL <https://localhost/a-Order.OrderTracking.Lab02.Ex03/> to run the claims-aware a-Order application. Because you have not authenticated, you will be immediately redirected to ADFS.
  2. If you see a **Certificate Error: Navigation Blocked** page, click **Continue to this website (not recommended)**. You will see the ADFS home realm discovery page. There will be two entries in the drop-down listbox, select the one that matches the name of your Active Directory domain (it may not match exactly the name shown in the following screenshot.) Then click **Continue to Sign In**.
     1. 
  3. On the Sign In page enter your domain name and **johndoe** as the user name, and **Pa$$w0rd** as the password. Then click **Sign In**.
     1. 
     2. If you don't see the ADFS sign in page, you should check your ADFS configuration. See the section "*Integrating with Active Directory Federation Services*" in the *Introduction* document for these labs for details.
  4. You will see the a-Order page with a list of Adatum items and your login details.
     1. 
  5. If you see a **Certificate Error: Navigation Blocked** page, click **Continue to this website (not recommended)**. You will see the ADFS home realm discovery page. There will be two entries in the drop-down listbox, select **Litware IP.** Then click **Continue to Sign In**.
     1. 
     2. If you are not using **InPrivate** browsing you will be redirected immediately to the Adatum sign-in page, by-passing this page. This is because ADFS has saved your preferred identity provider in a cookie. To see the home realm discovery page you will need to clear your browser's cookie cache.
  6. If you see a **Certificate Error: Navigation Blocked** page, click **Continue to this website (not recommended)**. On the Litware Issuer page, click **Click here to continue** to log in as Rick from Litware.
     1. 
  7. You will see the a-Order page with a list of Litware items and your login details.
     1. 

You have now completed this task to verify that ADFS functions as a federation provider for the a-Order application.

* 1. If you are using Windows Integrated Authentication, the Logout will not appear to work. This is because ADFS will silently sign the user in again using Windows Integrated Authentication the next time he tries to access the resource.
  2. You can use ADFS as part of your federated identity infrastructure. Adatum users can sign-in with their AD credentials, and Litware uses can sign-in with their corporate credentials. Both sets of users get access to their data in the a-Order application.

## Running the "End" Solution

1. If you did not complete all of the tasks in this exercise, you can run the "end" solution we provide.

To run the end solution

* 1. You must complete task 1 to add a test user to your Active Directory.
  2. You must complete task 2 to configure the a-Order application to trust your ADFS instance.
  3. You can run the PowerShell script, **Add-Adatum-Rules.ps1** in the Lab02-FederatedIdentity\Source\Ex03\Assets folder to add the Adatum claims rules to ADFS.
  4. You can run the PowerShell script, **Add-aOrder-RelyingParty.ps1** in the Lab02-FederatedIdentity\Source\Ex03\Assets folder to add the a-Order application as a relying party in ADFS.
  5. You can run the PowerShell script, **Add-aOrder-Rules.ps1** in the Lab02-FederatedIdentity\Source\Ex03\Assets folder to add the a-Order application claims rules to ADFS.
  6. You can run the PowerShell script, **Add-Litware-ClaimsProviderTrust.ps1** in the Lab02-FederatedIdentity\Source\Ex03\Assets folder to add the Litware issuer as a claims trust provider in ADFS.
     1. To run this script, you must first **cd** to the **Lab02-FederatedIdentity\Source\Ex03\Assets** folder at the **PowerShell** prompt.
  7. You can run the PowerShell script, **Add-Litware-Rules.ps1** in the Lab02-FederatedIdentity\Source\Ex03\Assets folder to add the Litware claims rules to ADFS.
  8. You must complete task 8 to configure the Litware issuer to work with your ADFS instance.
  9. You can follow the steps in task 9 to verify your solution.
     1. You should begin by browsing to <https://localhost/a-Order.OrderTracking.Lab02.Ex03.End/> in step 1 of task 9.