Hello Everyone, This is Enayat Meer with ADFS video series 2 with a new fresh set of computers.

ADFS a complete step by step live guide: **please request for this doc if needed**

I did not change any setup advised by Microsoft TechNet for lab testing purposes.

Please go to my YouTube page to see ADFS related videos while I am working on this one from scratch:

Here is the link: https://www.youtube.com/results?search_query=enayat+meer

For your convenience I am going to add all steps here because, I may need to change some steps in order to have accurate result. I am using Hyper-v environment running on a physical server with Intel Quad core i7 CPU, with 56 GB memory, about 8 TB hard drive (4 TB each). Let me show you my physical server:

Step 1: Preinstallation Tasks:

I will use 4 virtual machines for this live lab with complete steps shown in this table:

Computer name (I am keeping suggested Names)	AD FS client/server role	Operating system that I am using	IPv4 settings	DNS settings
adfsclient	Client	Windows 7	IP address:	Preferred:
			192.168.1.1	192.168.1.3
			Subnet mask:	Alternate:
			255.255.255.0	192.168.1.4
adfsweb	Web server	Windows Server 2008 R2 Enterprise (standard can be used as well)	IP address:	Preferred:
			192.168.1.2	192.168.1.4
			Subnet mask:	
			255.255.255.0	
adfsaccount	Federation server and domain controller	Windows Server 2008 R2 Enterprise	IP address:	Preferred:
			192.168.1.3	192.168.1.3
			Subnet mask:	
			255.255.255.0	
adfsresource	Federation server and domain controller	Windows Server 2008 R2 Enterprise	IP address:	Preferred:
			192.168.1.4	192.168.1.4
			Subnet mask:	
			255.255.255.0	

In this guide, A. Datum represents the account partner organization and Trey Research represents the resource partner organization.

Be sure to set both the preferred and alternate Domain Name System (DNS) server settings on the client. If both types of values are not configured as specified, the AD FS scenario will not function correctly.

TCP/IP static configuration is our first step as it is required:

IP configuration:

> Configure the IP addresses as specified in the previous table before you attempt to install AD DS. This helps ensure that DNS records are configured appropriately. Let's configure IP address now as shown on pervious page

As a security best practice, do not run domain controllers as both federation servers and domain controllers in a production environment.

Install AD DS from Server Manager: [validate connectivity first]

Let's install AD DS first as follows on both servers as 2 forests by promoting both servers.

Computer name	Domain Name
adfsaccount	adatum.com
adfsresource	treyresearch.net

Next Step:

Create accounts

After you set up two forests, you start the Active Directory Users and Computers snap-in to create some accounts that you can use to test and verify federated access across both forests. Configure the values in the following table on the adfsaccount computer.

Object to create	Name	User name	Action
Security global group	TreyClaimAppUsers	Not applicable	Not applicable
User	Alan Shen	alansh (alansh acts as the federated user who will be accessing the claims-aware application.)	Make alansh a member of the TreyClaimAppUsers global group.

Join computers to domain as follows:

Computer name	Join to
adfsclient	adatum.com
adfsweb	treyresearch.net

Step 2: Installing AD FS Role Services and Configuring Certificates

Now that you have configured the computers and joined them to the domain, you are ready to install Active Directory Federation Services (AD FS) role services on each of the servers. This step includes the following procedures:

- **Install the Federation Service**
- **Configure IIS to require SSL on both federation servers**
- **Install the AD FS Web Agent**
- **Create, export, and import certificates**

Install the Federation Service

Use the following procedure to install the Federation Service component of AD FS on the adfsaccount computer and the adfsresource computer. After the Federation Service is installed on a computer, that computer becomes a federation server.
This Federation Service installation procedure guides you through the process of creating a new trust policy file, self-signed Secure Sockets Layer (SSL) certificates, and token-signing certificates for each federation server.

To install the Federation Service

1. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.
2. Right-click **Roles**, and then click **Add Roles** to start the Add Roles Wizard.

3. On the **Before You Begin** page, click **Next**.
4. On the **Select Server Roles** page, click **Active Directory Federation Services**. Click **Next** two times.
5. On the **Select Role Services** page, select the **Federation Service** check box. If you are prompted to install additional Web Server (IIS) or Windows Process Activation Service role services, click **Add Required Role Services** to install them, and then click **Next**.
6. On the **Choose a Server Authentication Certificate for SSL Encryption** page, click **Create a self-signed certificate for SSL encryption**, and then click **Next**.
7. On the **Choose a Token-Signing Certificate** page, click **Create a self-signed token-signing certificate**, and then click **Next**.
8. On the **Select Trust Policy** page, click **Create a new trust policy**, and then click **Next** twice.
9. On the **Select Role Services** page, click **Next** to accept the default values.
10. Verify the information on the **Confirm Installation Selections** page, and then click **Install**.
11. On the **Installation Results** page, verify that everything installed correctly, and then click **Close**.

Configure IIS to require SSL on both federation servers

Use the following procedures to configure Internet Information Services (IIS) to require SSL on the default Web site of both the adfsresource federation server and the adfsaccount federation server.

To configure IIS on the adfsaccount server

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the console tree, double-click **ADFSACCOUNT**, double-click **Sites**, and then click **Default Web Site**.
3. In the **Actions** pane, click **Bindings**.
4. In the **Site Bindings** dialog box, click **Add**.
5. In **Type**, click **https**.
6. Under **SSL certificate**, click **adfsaccount.adatum.com**, click **OK**, and then click **Close**.
7. In the center pane, double-click **SSL Settings**, and then select the **Require SSL** check box.
8. Under **Client certificates**, click **Accept**, and then click **Apply**.

To configure IIS on the adfsresource server

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the console tree, double-click **ADFSRESOURCE**, double-click **Sites**, and then click **Default Web Site**.
3. In the center pane, double-click **SSL Settings**, and then select the **Require SSL** check box.
4. Under **Client certificates**, click **Accept**, and then click **Apply**.

Install the AD FS Web Agent

Use the following procedure to install the claims-aware Web Agent on the Web server (adfsweb).

To install the AD FS Web Agent

1. Click **Start**, point to **Administrative Tools**, and then click **Server Manager**.
2. Right-click **Roles**, and then click **Add Roles** to start the Add Roles Wizard.
3. On the **Before You Begin** page, click **Next**.

4. On the **Select Server Roles** page, click **Active Directory Federation Services**. Click **Next** two times.
5. On the **Select Role Services** page, select the **Claims-aware Agent** check box. If you are prompted to install additional Web Server (IIS) or Windows Process Activation Service role services, click **Add Required Role Services** to install them, and then click **Next**.
6. On the **Web Server (IIS)** page, click **Next**.
7. On the **Select Role Services** page, in addition to the preselected check boxes, select the **Client Certificate Mapping Authentication** and **IIS Management Console** check boxes, and then click **Next**.
   The **Client Certificate Mapping Authentication** check box installs the components that IIS must have to create a self-signed server authentication certificate that is required for this server.
8. After you verify the information on the **Confirm Installation Selections** page, click **Install**.
9. On the **Installation Results** page, verify that everything installed correctly, and then click **Close**.

Create, export, and import certificates

The most important factor in setting up the Web server and the federation servers successfully is creating and exporting the required certificates appropriately. Because you previously used the Add Roles Wizard to create the server authentication certificate for both of the federation servers, all you have to do now is create the server authentication certificate for the adfsweb computer. This section includes the following procedures:

- Create a server authentication certificate for adfsweb
- Export the token-signing certificate from adfsaccount to a file
- Export the adfsresource server authentication certificate to a file
- Import the server authentication certificate for adfsresource to adfsweb

In a production environment, certificates are obtained from a certification authority (CA). For the purposes of the test lab deployment in this guide, self-signed certificates are used.

Create a server authentication certificate for adfsweb

Use the following procedure on the Web server (adfsweb) to create a self-signed server authentication certificate.

To create a server authentication certificate for adfsweb

Export the token-signing certificate from adfsaccount to a file

Use the following procedure on the account federation server (adfsaccount) to export the token-signing certificate from adfsaccount to a file.

To export the token-signing certificate from adfsaccount to a file

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Right-click **Federation Service**, and then click **Properties**.
3. On the **General** tab, click **View**.
4. On the **Details** tab, click **Copy to File**.
5. On the **Welcome to the Certificate Export Wizard** page, click **Next**.
6. On the **Export Private Key** page, click **No, do not export the private key**, and then click **Next**.
7. On the **Export File Format** page, click **DER encoded binary X.509 (.CER)**, and then click **Next**.
8. On the **File to Export** page, type **d:\adfsaccount_ts.cer**, and then click **Next**.

NOTE: The adfsaccount token-signing certificate will be imported to adfsresource later when the Account Partner Wizard prompts you for the Account Partner Verification Certificate. (See Step 4: Configuring the Federation Servers.) At that time, you access adfsresource over the network to obtain this file.

9. On the **Completing the Certificate Export Wizard**, click **Finish**.

Export the adfsresource server authentication certificate to a file

Import the server authentication certificate for adfsresource to adfsweb

To import the server authentication certificate for adfsresource, perform the following procedure on the Web server (adfsweb).

To import the server authentication certificate for adfsresource to adfsweb

1. Click **Start**, click **Run**, type **mmc**, and then click **OK**.
2. Click **File**, and then click **Add/Remove Snap-in**.
3. Select **Certificates**, click **Add**, click **Computer account**, and then click **Next**.
4. Click **Local computer: (the computer this console is running on)**, click **Finish**, and then click **OK**.

5. In the console tree, double-click the **Certificates (Local Computer)** icon, double-click the **Trusted Root Certification Authorities** folder, right-click **Certificates**, point to **All Tasks**, and then click **Import**.
6. On the **Welcome to the Certificate Import Wizard** page, click **Next**.
7. On the **File to Import** page, type **\\adfsresource\d$\adfsresource.pfx**, and then click **Next**.

NOTE:

You may have to map the network drive to obtain the adfsresource.pfx file. You can also copy the adfsresource.pfx file directly from adfsresource to adfsweb, and then point the wizard to that location.

8. On the **Password** page, type the password for the adfsresource.pfx file, and then click **Next**.
9. On the **Certificate Store** page, click **Place all certificates in the following store**, and then click **Next**.
10. On the **Completing the Certificate Import Wizard** page, verify that the information you provided is accurate, and then click **Finish**.

Step 3: Configuring the Web Server

Applies To: Windows Server 2008 R2
This step includes procedures for setting up a claims-aware application on the Web server (adfsweb). You can use the following procedures to configure Internet Information Services (IIS) and the claims-aware application:

- Configure IIS on the Web server
- Create and configure the claims-aware application

Administrative credentials
To perform all the procedures in this step, log on to adfsweb with the local Administrator account.

Configure IIS on the Web server

Use the following procedure to configure IIS on the Web server (adfsweb).

To configure IIS on the Web server

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the console tree, double-click **ADFSWEB**, double-click **Sites**, and then click **Default Web Site**.
3. In the **Actions** pane, click **Bindings**.
4. In the **Site Bindings** dialog box, click **Add**.
5. In **Type**, click **https**.
6. Under **SSL certificate**, click **adfsweb**, click **OK**, and then click **Close**.
7. In the center pane, double-click **SSL Settings**, select the **Require SSL** check box.
8. Under **Client certificates**, click **Accept**, and then click **Apply**.

Create and configure the claims-aware application

Use the following procedure to configure the Web server (adfsweb) to host a sample claims-aware application.

To create and configure the claims-aware application

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the console tree, double-click **ADFSWEB**, double-click **Sites**, right-click **Default Web Site**, and then click **Add Application**.
3. In the **Add Application** dialog box, in **Alias**, type **claimapp**.
4. Click **Select**, select **Classic .NET AppPool** in the drop-down menu, and then click **OK**.
5. Click the **...** button, and then highlight the **d:\inetpub\wwwroot** folder.
6. Click **Make New Folder**, name the folder claimapp, click **OK**, and then click **OK** again.

NOTE: Do not use capital letters in the claimapp folder name. If this folder name contains capital letters, users must also use capital letters when they type the address of the Web site.

7. Create the three files that make up the sample claims-aware application by using the procedures in Appendix: Creating the Sample Claims-Aware Application. After you create the files, copy them into the d:\inetpub\wwwroot\claimapp folder.

Step 4: Configuring the Federation Servers

Now that you have installed Active Directory Federation Services (AD FS) and you have configured the Web server for the sample claims-aware application, next you configure the Federation Service on the federation servers for both the A. Datum Corporation and Trey Research. In this step, you:

- Make the Federation Service for Trey Research aware of the claims-aware application.
- Add account stores and group claims to the appropriate Federation Service.
- Configure each of the group claims so that they map to an Active Directory Domain Services (AD DS) group in the appropriate forest.

This step includes the following tasks:

- Configure the Federation Service for A. Datum Corporation
- Configure the Federation Service for Trey Research
- Create both sides of the federated trust using import and export functionality

Administrative credentials

To perform all the procedures in this step, log on to the adfsaccount computer and the adfsresource computer with the Administrator account for the domain.

Configure the Federation Service for A. Datum Corporation

This section includes the following procedures:

- Configure the A. Datum trust policy
- Create a group claim for the claims-aware application
- Add and configure an AD DS account store

Configure the A. Datum trust policy

Use the following procedure on the adfsaccount computer to configure the trust policy for the Federation Service for A. Datum Corporation.

To configure the A Datum trust policy

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. In the console tree, double-click **Federation Service**, right-click **Trust Policy**, and then click **Properties**.
3. On the **General** tab, in **Federation Service URI**, type **urn:federation:adatum**.[value is case sensitive]
4. In the **Federation Service endpoint URL** text box, verify that **https://adfsaccount.adatum.com/adfs/ls/** appears.
5. On the **Display Name** tab, in **Display name for this trust policy**, type **A. Datum** (replace any value that may already exist in this field with A. Datum), and then click **OK**.

Create a group claim for the claims-aware application

Use the following procedure to create a group claim that will be used to authenticate to the treyresearch.net forest.

To create a group claim for the claims-aware application

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **My Organization**, right-click **Organization Claims**, point to **New**, and then click **Organization Claim**.
3. In the **Create a New Organization Claim** dialog box, in **Claim name**, type **Trey ClaimApp Claim**.
4. Ensure that **Group claim** is selected, and then click **OK**.

Add and configure an AD DS account store

Use the following procedures to add an AD DS account store to the Federation Service for A. Datum Corporation.

- Add an AD DS account store
- Map a global group to the group claim for the claims-aware application

Add an AD DS account store

Use the following procedure to add an AD DS account store.

To add an AD DS account store

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **My Organization**, right-click **Account Stores**, point to **New**, and then click **Account Store**.
3. On the **Welcome to the Add Account Store Wizard** page, click **Next**.
4. On the **Account Store Type** page, ensure that **Active Directory Domain Services** is selected, and then click **Next**.

NOTE: You can have only one AD DS store that is associated with a Federation Service. If the AD DS option is not available, an AD DS store has already been created for this Federation Service.

5. On the **Enable this Account Store** page, ensure that the **Enable this account store** check box is selected, and then click **Next**.
6. On the **Completing the Add Account Store Wizard** page, click **Finish**.

Map a global group to the group claim for the claims-aware application

Use the following procedure to map an AD DS global group to the Trey ClaimApp Claim group claim.

To map a global group to the group claim for the claims-aware application

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **My Organization**, double-click **Account Stores**, right-click **Active Directory**, point to **New**, and then click **Group Claim Extraction**.
3. In the **Create a New Group Claim Extraction** dialog box, click **Add**, type **treyclaimappusers**, and then click **OK**.
4. Ensure that the **Map to this Organization Claim** menu displays **Trey ClaimApp Claim**, and then click **OK**.

Configure the Federation Service for Trey Research

This section includes the following procedures:

- Configure the Trey Research trust policy
- Create a group claim for the claims-aware application
- Add an AD DS account store

- Add and configure a claims-aware application

Configure the Trey Research trust policy

Use the following procedure on the adfsresource computer to configure the trust policy for the Federation Service in Trey Research.

To configure the Trey Research trust policy

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. In the console tree, double-click **Federation Service**, right-click **Trust Policy**, and then click **Properties**.
3. On the **General** tab, in **Federation Service URI**, type **urn:federation:treyresearch**.

NOTE: This value is case sensitive.

4. In the **Federation Service endpoint URL** text box, verify that **https://adfsresource.treyresearch.net/adfs/ls/** appears.
5. On the **Display Name** tab, in **Display name for this trust policy**, type **Trey Research** (replace any value that may already exist in this field with Trey Research), and then click **OK**.

Create a group claim for the claims-aware application

Use the following procedure to create a group claim that will be used to make authorization decisions for the sample claims-aware application on behalf of users in the adatum.com forest.

To create a group claim for the claims-aware application

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **My Organization**, right-click **Organization Claims**, point to **New**, and then click **Organization Claim**.
3. In the **Create a New Organization Claim** dialog box, in **Claim name**, type **Adatum ClaimApp Claim**.
4. Ensure that **Group claim** is selected, and then click **OK**.

Add an AD DS account store

Use the following procedure to add an AD DS account store to the Federation Service for Trey Research.

To add an AD DS account store

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **My Organization**, right-click **Account Stores**, point to **New**, and then click **Account Store**.
3. On the **Welcome to the Add Account Store Wizard** page, click **Next**.
4. On the **Account Store Type** page, ensure that **Active Directory Domain Services** is selected, and then click **Next**.
5. On the **Enable this Account Store** page, ensure that the **Enable this account store** check box is selected, and then click **Next**.
6. On the **Completing the Add Account Store Wizard** page, click **Finish**.

Add and configure a claims-aware application

Use the following procedures on the adfsresource computer to add a claims-aware application to the Federation Service for Trey Research.

- Add a claims-aware application
- Enable Adatum ClaimApp Claim

Add a claims-aware application

Use the following procedure to add a claims-aware application to the Federation Service.

To add a claims-aware application

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **My Organization**, right-click **Applications**, point to **New**, and then click **Application**.
3. On the **Welcome to the Add Application Wizard** page, click **Next**.
4. On the **Application Type** page, click **Claims-aware application**, and then click **Next**.
5. On the **Application Details** page, in **Application display name**, type **Claims-aware Application**.
6. In **Application URL**, type **https://adfsweb.treyresearch.net/claimapp/**, and then click **Next**.
7. On the **Accepted Identity Claims** page, click **User principal name (UPN)**, and then click **Next**.
8. On the **Enable this Application** page, ensure that the **Enable this application** check box is selected, and then click **Next**.
9. On the **Completing the Add Application Wizard** page, click **Finish**.

Enable Adatum ClaimApp Claim

Now that the Federation Service recognizes the application, use the following procedure to enable the Adatum ClaimApp Claim group claim for that application.

To enable Adatum ClaimApp Claim

1. In the **Applications** folder, click **Claims-aware Application**.
2. Right-click **Adatum ClaimApp Claim**, and then click **Enable**.

Create both sides of the federated trust using import and export functionality

Creating federated trusts between partner organizations is easier in Windows Server 2008 R2 than it was in earlier Windows operating systems because of enhanced, policy-based export and import functionality. In this section, you use this import and export functionality to exchange policy files between the A. Datum and Trey Research organizations to successfully create the federated trust.
This section includes the following procedures:

- Export the trust policy from A. Datum
- Import the A. Datum trust policy to Trey Research
- Create a claim mapping in Trey Research

- Export the partner policy from Trey Research
- Import the Trey Research partner policy to A. Datum

Export the trust policy from A. Datum

On the adfsaccount computer at A. Datum, use the following procedure to export the trust policy data that you use in the next procedure to create one side of the federation trust relationship between A. Datum and Trey Research.

Export the trust policy from A. Datum

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, right-click **Trust Policy**, and then click **Export Basic Partner Policy**.
3. In the **Export Basic Partner Policy** dialog box, click **Browse**, in **File name** type **d:\adfsaccount**, click **Save**, and then click **OK**.

NOTE: If this were an actual AD FS production environment, the administrator in A. Datum would now send the exported policy file to the resource partner administrator at Trey Research by e-mail or other means.

Import the A. Datum trust policy to Trey Research

On the adfsresource computer at Trey Research, use the following procedure to import the A. Datum trust policy data that you must have to finish creating the first side of the federation trust and to add A. Datum as an account partner to the Trey Research trust policy.

Import the A. Datum trust policy to Trey Research

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **Partner Organizations**, right-click **Account Partners**, point to **New**, and then click **Account Partner**.
3. On the **Welcome to the Add Account Partner Wizard** page, click **Next**.
4. On the **Import Policy File** page, under **Partner interoperability policy file** type **\\adfsaccount\d$\adfsaccount.xml**, click **Yes**, and then click **Next**.
5. On the **Account Partner Details** page, ensure that:
    - **Display name** displays **A. Datum**.
    - **Federation Service URI** displays **urn:federation:adatum**.
    - **Federation Service endpoint URL** displays **https://adfsaccount.adatum.com/adfs/ls/**, and then click **Next**.
6. On the **Account Partner Verification Certificate** page, ensure that **Use the verification certificate in the import policy file** is selected, and then click **Next**.
7. On the **Federation Scenario** page, ensure that **Federated Web SSO** is selected, and then click **Next**.

8. On the **Account Partner Identity Claims** page, ensure that the **UPN Claim** and **E-mail Claim** check boxes are selected, and then click **Next**.
9. On the **Accepted UPN Suffixes** page, type **adatum.com**, click **Add**, and then click **Next**.
10. On the **Accepted E-mail Suffixes** page, type **adatum.com**, click **Add**, and then click **Next**.
11. On the **Enable this Account Partner** page, ensure that the **Enable this account partner** check box is selected, and then click **Next**.
12. On the **Completing the Add Account Partner Wizard** page, click **Finish**.

Create a claim mapping in Trey Research

On the adfsresource computer at Trey Research, use the following procedure to create an incoming group claim mapping to use for the sample claims-aware application. In the next procedure, you export this claim mapping to A. Datum along with other policy data that is relevant to the creation of this federated trust relationship.

NOTE: At A. Datum, when you import the policy data from Trey Research, you will be prompted to automatically create an outgoing group claim mapping based on the name of the incoming group claim mapping that you create in this procedure (ClaimAppMapping). This part of the import process helps prevent typographical errors.

Create a claim mapping in Trey Research

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **Partner Organizations**, double-click **Account Partners,** right-click **A. Datum**, point to **New**, and then click **Incoming Group Claim Mapping**.
3. In the **Create a New Incoming Group Claim Mapping** dialog box, in **Incoming group claim name**, type **ClaimAppMapping**.

NOTE: This value is case sensitive. It must match exactly the value that you specified in the outgoing group claim mapping in the account partner organization, A. Datum.

4. In **Organization group claim**, click **Adatum ClaimApp Claim**, and then click **OK**.

Export the partner policy from Trey Research

On the adfsresource computer at Trey Research, use the following procedure to export the Trey Research partner policy data to use in the next procedure to create the second side of the federation trust relationship.

Export the partner policy from Trey Research

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **Partner Organizations**, double-click **Account Partner**, right-click **A. Datum**, and then click **Export Policy**.

3. In the **Export Partner Policy** dialog box, click **Browse**, in **File name** type **d:\adfsresource**, click **Save**, and then click **OK**.

NOTE: If this were an actual AD FS production environment, the administrator in Trey Research would now send the exported partner policy file to the account partner administrator by e-mail or other means.

Import the Trey Research partner policy to A. Datum

On the adfsaccount computer at A. Datum, use the following procedure to import the Trey Research partner policy data that you must have so that you can finish creating the second side of the federation trust and add Trey Research as a resource partner to the A. Datum trust policy.

Import the Trey Research partner policy to A. Datum

1. Click **Start**, point to **Administrative Tools**, and then click **Active Directory Federation Services**.
2. Double-click **Federation Service**, double-click **Trust Policy**, double-click **Partner Organizations**, right-click **Resource Partners**, point to **New**, and then click **Resource Partner**.
3. On the **Welcome to the Add Resource Partner Wizard** page, click **Next**.
4. On the **Import Policy File** page, click **Yes**, under **Partner interoperability policy file** type **\\adfsresource\d$\adfsresource.xml**, and then click **Next**.
5. On the **Resource Partner Details** page, ensure that:
    - **Display name** displays **Trey Research**.
    - **Federation Service URI** displays **urn:federation:treyresearch**.
    - **Federation Service endpoint URL** displays **https://adfsresource.treyresearch.net/adfs/ls/**, and then click **Next**.
6. On the **Account Partner Verification Certificate** page, ensure that **Use the verification certificate in the import policy file** is selected, and then click **Next**.
7. On the **Federation Scenario** page, ensure that **Federated Web SSO** is selected, and then click **Next**.
8. On the **Resource Partner Identity Claims** page, ensure that the **UPN Claim** and **E-mail Claim** check boxes are selected, and then click **Next**.
9. On the **Select UPN Suffix** page, ensure that **Replace all UPN suffixes with the following** displays **adatum.com**, and then click **Next**.
10. On the **Select E-mail Suffix** page, ensure that **Replace all E-mail suffixes with** displays **adatum.com**, and then click **Next**.
11. On the **Map Claim Transformations** page, under **Mapping** select **Trey ClaimApp Claim**, and then click **Next**.
12. On the **Enable this Resource Partner** page, ensure that the **Enable this resource partner** check box is selected, and then click **Next**.
13. On the **Completing the Add Resource Partner Wizard** page, click **Finish**.

Step 5: Configuring Client Certificates and Testing the Sample Application

In this step, you prepare, distribute, and use the certificates of Active Directory Federation Services (AD FS) to test access from the client computer to a claims-aware application.
This step includes the following procedures:

- Export adfsweb and adfsaccount certificates to a file
- Export adfsweb and adfsaccount certificates to a file
- Access the claims-aware application from the client computer

Export adfsweb and adfsaccount certificates to a file

Administrative credentials

To perform the procedures in this step, you must log on to the adfsweb and adfsaccount computers using with the domain administrator account.
Use this procedure to export the server authentication certificates for adfsweb and adfsaccount to files. By performing this step now and then importing the certificates to the adfsclient computer in the next step, you will optimize the user experience by preventing certificate prompts that users normally see when they access the federated applications. The adfsresource server authentication certificate was exported to a file in step 2. It is not necessary to export that certificate again. In the next procedure, you import these certificates to the adfsclient computer.

To export adfsweb and adfsaccount certificates to a file

1. On the adfsweb computer, click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the console tree, click **ADFSWEB**.
3. In the center pane, double-click **Server Certificates**.
4. In the center pane, right-click **adfsweb.treyresearch.net**, and then click **Export**.
5. In the **Export Certificate** dialog box, click the **...** button.
6. In **File name**, type **d:\adfsweb**, and then click **Open**.
7. Type a password for the certificate, confirm it, and then click **OK**.
8. Repeat steps 1 through 7 on the adfsaccount computer. In step 6, save the file as C:\adfsaccount.

Import adfsweb, adfsaccount, and adfsresource certificates

Administrative credentials

To perform the procedures in this step, you must log on to the adfsclient computer using the local administrator account.
Use this procedure to import each of the server authentication certificates from adfsweb, adfsaccount, and adfsresource to the Local Computers Trusted Root Certification Authorities certificate store.

To import adfsweb, adfsaccount, and adfsresource certificates

1. Log on to the adfsclient computer with the local administrator account, click **Start**, in **Search programs and files**, type **mmc**, and then click **OK**. Click **File**, and then click **Add/Remove Snap-in**.
2. Click **Certificates**, click **Add**, click **Computer account**, and then click **Next**.
3. Click **Local computer: (the computer this console is running on)**, click **Finish**, and then click **OK**.
4. In the console tree, double-click the **Certificates (Local Computer)** icon, double-click the **Trusted Root Certification Authorities** folder, right-click **Certificates**, point to **All Tasks**, and then click **Import**.
5. On the **Welcome to the Certificate Import Wizard** page, click **Next**.
6. On the **File to Import** page, click **Browse**, in **File name** type **\\adfsresource\d$\adfsresource.pfx**, click **Open**, and then click **Next**.

NOTE: You may have to map the network drive to obtain the adfsresource.pfx file. You can also copy the adfsresource.pfx file directly from adfsresource to adfsclient, and then point the wizard to that location.

7. On the **Password** page, type the password for the adfsresource.pfx file, and then click **Next**.
8. On the **Certificate Store** page, click **Place all certificates in the following store**, and then click **Next**.
9. On the **Completing the Certificate Import Wizard** page, verify that the information you provided is accurate, and then click **Finish**.
10. Repeat these steps on the adfsclient computer until you have imported the adfsaccount and adfsweb certificates, and then proceed to the next section.

Access the claims-aware application from the client computer

Administrative credentials

To perform the procedures in this step, it is not necessary to log on with administrative credentials to the client computer. In other words, if you are logged on to the client as Alan Shen (alansh), you can access the claims-aware application without adding alansh to any of the local administrator groups (for example, Power Users, Administrators) on the adfsclient computer.

Use the following procedure to access the sample claims-aware application from a client that is authorized for that application.

To access the claims-aware application from the client computer

1. Log on to the adfsclient computer as alansh.
2. Open a browser window, and then install the required certificates on the client by doing the following:
    a. Go to https://adfsaccount.adatum.com/.

       The browser displays a "Certificate Error: Navigation Blocked" error message that notifies you that the incoming certificate was not issued by a trusted certification authority. This

error is expected behavior when you deploy Active Directory Federation Services (AD FS) servers with self-signed certificates.

b. Click the **Continue to this website (not recommended)** link.

c. In the address bar, click **Certificate Error**, and then click **View certificates**.

d. In the **Certificate** dialog box, click **Install Certificate**.

e. On the **Welcome to the Certificate Import Wizard** page, click **Next**.

f. On the **Certificate Store** page, click **Place all certificates in the following store**, and then click **Browse**.

g. In the **Select Certificate Store** dialog box, highlight **Trusted Root Certification Authorities**, click **OK**, and then click **Next**.

h. On the **Completing the Certificate Import Wizard** page, click **Finish**.

i. On the **Security Warning** dialog box, click **Yes**.

j. Click **OK** twice.

k. Repeat steps a through j using https://adfsresource.treyresearch.net and https://adfsweb.treyresearch.net to install all three certificates into the Trusted Root Certification Authorities certificate store.

3. Go to https://adfsweb.treyresearch.net/claimapp/. When you are prompted for your home realm, click **A. Datum Corporation**, and then click **Submit**.
4. At this point **SSO Sample Application** appears in the browser. You can see which claims were sent to the Web server in the **SingleSignOnIdentity.SecurityPropertyCollection** section of the sample application.

NOTE: If for any reason you have problems accessing the claims-aware application, consider running the *iisreset* ***command or restarting the adfsweb computer. Then, try to access the application again.***

This section is optional:::::::::::::::::::

Appendix: Creating the Sample Claims-Aware Application

You can use the claims-aware application in this appendix to test which claims a Federation Service sends in Active Directory Federation Services (AD FS) security tokens. This appendix includes instructions for

setting up a sample claims-aware application on your Web server. By using this sample claims-aware application and the supporting instructions in Step 3: Configuring the Web Server, you can prepare the application for testing on the client computer and complete the Web server setup process.
The sample claims-aware application is made up of the following three files:

- Default.aspx
- Web.config
- Default.aspx.cs

Membership in the local **Administrators** group of adfsweb is the minimum required to complete the procedures in this step. Review details about using the appropriate accounts and group memberships at Local and Domain Default Groups (http://go.microsoft.com/fwlink/?LinkId=83477).
For this application to function correctly, you must use the following procedures to create each of the required files in order. After you create the files, move them to the D:\inetpub\wwwroot\claimapp directory on the adfsweb computer.

- Create the Default.aspx file
- Create the Web.config file
- Create the Default.aspx.cs file

Create the Default.aspx file

Use the following procedure to create the Default.aspx file.

To create the Default.aspx file

1. Start Notepad.
2. Copy and paste the following code into a new Notepad file:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<%@ OutputCache Location="None" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-
1252">
<title>Claims-aware Sample Application</title>
<style>
<!--
.pagetitle { font-family: Verdana; font-size: 18pt; font-weight: bold;}
.propertyTable td { border: 1px solid; padding: 0px 4px 0px 4px}
```

```
.propertyTable th { border: 1px solid; padding: 0px 4px 0px 4px; font-
weight: bold; background-color: #cccccc ; text-align: left }
.propertyTable { border-collapse: collapse;}
td.l{ width: 200px }
tr.s{ background-color: #eeeeee }
.banner { margin-bottom: 18px }
.propertyHead { margin-top: 18px; font-size: 12pt; font-family: Arial;
font-weight: bold; margin-top: 18}
.abbrev { color: #0066FF; font-style: italic }
-->
</style>
</head>
<body>
<form ID="Form1" runat=server>
<div class=banner>
<div class=pagetitle>SSO Sample</div>
[ <asp:HyperLink ID=SignOutUrl runat=server>Sign Out</asp:HyperLink> |
<a href="<%=Context.Request.Url.GetLeftPart(UriPartial.Path)%>">Refresh
without viewstate data</a>]
</div>
<div class=propertyHead>Page Information</div>
<div style="padding-left: 10px; padding-top: 10px">
<asp:Table runat=server ID=PageTable CssClass=propertyTable>
<asp:TableHeaderRow>
<asp:TableHeaderCell>Name</asp:TableHeaderCell>
<asp:TableHeaderCell>Value</asp:TableHeaderCell>
<asp:TableHeaderCell>Type</asp:TableHeaderCell>
</asp:TableHeaderRow>
</asp:Table>
</div>
<div class=propertyHead>User.Identity</div>
<div style="padding-left: 10px; padding-top: 10px">
<asp:Table CssClass="propertyTable" ID=IdentityTable runat=server>
<asp:TableHeaderRow>
<asp:TableHeaderCell>Name</asp:TableHeaderCell>
<asp:TableHeaderCell>Value</asp:TableHeaderCell>
<asp:TableHeaderCell>Type</asp:TableHeaderCell>
</asp:TableHeaderRow>
</asp:Table>
</div>
<div class=propertyHead>(IIdentity)User.Identity</div>
<div style="padding-left: 10px; padding-top: 10px">
<asp:Table CssClass="propertyTable" ID=BaseIdentityTable runat=server>
<asp:TableHeaderRow>
<asp:TableHeaderCell>Name</asp:TableHeaderCell>
```

```
<asp:TableHeaderCell>Value</asp:TableHeaderCell>
<asp:TableHeaderCell>Type</asp:TableHeaderCell>
</asp:TableHeaderRow>
</asp:Table>
</div>
<div class=propertyHead>(SingleSignOnIdentity)User.Identity</div>
<div style="padding-left: 10px; padding-top: 10px">
<asp:Table CssClass="propertyTable" ID=SSOIdentityTable runat=server>
<asp:TableHeaderRow>
<asp:TableHeaderCell>Name</asp:TableHeaderCell>
<asp:TableHeaderCell>Value</asp:TableHeaderCell>
<asp:TableHeaderCell>Type</asp:TableHeaderCell>
</asp:TableHeaderRow>
</asp:Table>
</div>
<div
class=propertyHead>SingleSignOnIdentity.SecurityPropertyCollection</div
>
<div style="padding-left: 10px; padding-top: 10px">
<asp:Table CssClass="propertyTable" ID=SecurityPropertyTable
runat=server>
<asp:TableHeaderRow>
<asp:TableHeaderCell>Uri</asp:TableHeaderCell>
<asp:TableHeaderCell>Claim Type</asp:TableHeaderCell>
<asp:TableHeaderCell>Claim Value</asp:TableHeaderCell>
</asp:TableHeaderRow>
</asp:Table>
</div>
<div class=propertyHead>(IPrincipal)User.IsInRole(...)</div>
<div style="padding-left: 10px; padding-top: 10px">
<asp:Table CssClass="propertyTable" ID=RolesTable runat=server>
</asp:Table>
<div style="padding-top: 10px">
<table>
<tr><td>Roles to check (semicolon separated):</td></tr>
<tr><td><asp:TextBox ID=Roles Columns=55 runat=server/></td><td
align=right><asp:Button UseSubmitBehavior=true ID=GetRoles runat=server
Text="Check Roles" OnClick="GoGetRoles"/></td></tr>
</table>
</div>
</div>
</form>
</body>
</html>
```

3. Save the Notepad file as Default.aspx in the D:\inetpub\wwwroot\claimapp directory.

Create the Web.config file

Use the following procedure to create the Web.config file.

To create the Web.config file

1. Start Notepad.
2. Copy and paste the following code into a new Notepad file:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<configSections>
<sectionGroup name="system.web">
<section name="websso"
type="System.Web.Security.SingleSignOn.WebSsoConfigurationHandler,
System.Web.Security.SingleSignOn, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35, Custom=null" />
</sectionGroup>
</configSections>
<system.web>
<sessionState mode="Off" />
<compilation defaultLanguage="c#" debug="true">
<assemblies>
<add assembly="System.Web.Security.SingleSignOn, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35, Custom=null"/>
<add assembly="System.Web.Security.SingleSignOn.ClaimTransforms,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35,
Custom=null"/>
</assemblies>
</compilation>
<customErrors mode="Off"/>
<authentication mode="None" />
<httpModules>
<add
name="Identity Federation Services Application Authentication Module"
type="System.Web.Security.SingleSignOn.WebSsoAuthenticationModule,
System.Web.Security.SingleSignOn, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35, Custom=null" />
</httpModules>
<websso>
<authenticationrequired />
<eventloglevel>55</eventloglevel>
<auditsuccess>2</auditsuccess>
<urls>
<returnurl>https://adfsweb.treyresearch.net/claimapp/</returnurl>
</urls>
<cookies writecookies="true">
```

```
<path>/claimapp</path>
<lifetime>240</lifetime>
</cookies>
<fs>https://adfsresource.treyresearch.net/adfs/fs/federationserverservi
ce.asmx</fs>
</websso>
</system.web>
<system.diagnostics>
<switches>
<add name="WebSsoDebugLevel" value="0" /> <!-- Change to 255 to enable
full debug logging -->
</switches>
<trace autoflush="true" indentsize="3">
<listeners>
<add name="LSLogListener"
type="System.Web.Security.SingleSignOn.BoundedSizeLogFileTraceListener,
System.Web.Security.SingleSignOn, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35, Custom=null"
initializeData="d:\logdir\claimapp.log" />
</listeners>
</trace>
</system.diagnostics>
</configuration>
```

3. Save the Notepad file as Web.config in the D:\inetpub\wwwroot\claimapp directory.

Create the Default.aspx.cs file

Use the following procedure to create the Default.aspx.cs file.

To create the Default.aspx.cs file

1. Start Notepad.
2. Copy and paste the following code into a new Notepad file:

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Configuration;
using System.Reflection;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Security;
using System.Security.Principal;
using System.Web.Security.SingleSignOn;
```

```
using System.Web.Security.SingleSignOn.Authorization;
public partial class _Default : System.Web.UI.Page
{
const string NullValue = "<span class=\"abbrev\" title=\"Null
Reference, or not applicable\"><b>null</b></span>";
static Dictionary<string, string> s_abbreviationMap;
static _Default()
{
s_abbreviationMap = new Dictionary<string, string>();
//
// Add any abbreviations here. Make sure that prefixes of
// replacements occur *after* the longer replacement key.
//
s_abbreviationMap.Add("System.Web.Security.SingleSignOn.Authorization",
"SSO.Auth");
s_abbreviationMap.Add("System.Web.Security.SingleSignOn", "SSO");
s_abbreviationMap.Add("System", "S");
}
protected void Page_Load(object sender, EventArgs e)
{
SingleSignOnIdentity ssoId = User.Identity as SingleSignOnIdentity;
//
// Get some property tables initialized.
//
PagePropertyLoad();
IdentityLoad();
BaseIdentityLoad();
SSOIdentityLoad(ssoId);
SecurityPropertyTableLoad(ssoId);
//
// Filling in the roles table
// requires a peek at the viewstate
// since we have a text box driving this.
//
if (!IsPostBack)
{
UpdateRolesTable(new string[] { });
}
else
{
GoGetRoles(null, null);
}
//
// Get the right links for SSO
//
```

```
if (ssoId == null)
{
SignOutUrl.Text = "Single Sign On isn't installed...";
SignOutUrl.Enabled = false;
}
else
{
if (ssoId.IsAuthenticated == false)
{
SignOutUrl.Text = "Sign In (you aren't authenticated)";
SignOutUrl.NavigateUrl = ssoId.SignInUrl;
}
else
SignOutUrl.NavigateUrl = ssoId.SignOutUrl;
}
}
void SecurityPropertyTableLoad(SingleSignOnIdentity ssoId)
{
Table t = SecurityPropertyTable;
if (ssoId == null)
{
AddNullValueRow(t);
return;
}
//
// Go through each of the security properties provided.
//
bool alternating = false;
foreach (SecurityProperty securityProperty in
ssoId.SecurityPropertyCollection)
{
t.Rows.Add(CreateRow(securityProperty.Uri, securityProperty.Name,
securityProperty.Value, alternating));
alternating = !alternating;
}
}
void UpdateRolesTable(string[] roles)
{
Table t = RolesTable;
t.Rows.Clear();
bool alternating = false;
foreach (string s in roles)
{
string role = s.Trim();
t.Rows.Add(CreatePropertyRow(role, User.IsInRole(role), alternating));
```

```
alternating = !alternating;
}
}
void IdentityLoad()
{
Table propertyTable = IdentityTable;
if (User.Identity == null)
{
AddNullValueRow(propertyTable);
}
else
{
propertyTable.Rows.Add(CreatePropertyRow("Type name",
User.Identity.GetType().FullName));
}
}
void SSOIdentityLoad(SingleSignOnIdentity ssoId)
{
Table propertyTable = SSOIdentityTable;
if (ssoId != null)
{
PropertyInfo[] props =
ssoId.GetType().GetProperties(BindingFlags.Instance |
BindingFlags.Public | BindingFlags.DeclaredOnly);
AddPropertyRows(propertyTable, ssoId, props);
}
else
{
AddNullValueRow(propertyTable);
}
}
void PagePropertyLoad()
{
Table propertyTable = PageTable;
string leftSidePath = Request.Url.GetLeftPart(UriPartial.Path);
propertyTable.Rows.Add(CreatePropertyRow("Simplified Path",
leftSidePath));
}
void BaseIdentityLoad()
{
Table propertyTable = BaseIdentityTable;
IIdentity identity = User.Identity;
if (identity != null)
{
```

```
PropertyInfo[] props =
typeof(IIdentity).GetProperties(BindingFlags.Instance |
BindingFlags.Public | BindingFlags.DeclaredOnly);
AddPropertyRows(propertyTable, identity, props);
}
else
{
AddNullValueRow(propertyTable);
}
}
void AddNullValueRow(Table table)
{
TableCell cell = new TableCell();
cell.Text = NullValue;
TableRow row = new TableRow();
row.CssClass = "s";
row.Cells.Add(cell);
table.Rows.Clear();
table.Rows.Add(row);
}
void AddPropertyRows(Table propertyTable, object obj, PropertyInfo[]
props)
{
bool alternating = false;
foreach (PropertyInfo p in props)
{
string name = p.Name;
object val = p.GetValue(obj, null);
propertyTable.Rows.Add(CreatePropertyRow(name, val, alternating));
alternating = !alternating;
}
}
TableRow CreatePropertyRow(string propertyName, object propertyValue)
{
return CreatePropertyRow(propertyName, propertyValue, false);
}
TableRow CreatePropertyRow(string propertyName, object value, bool
alternating)
{
if (value == null)
return CreateRow(propertyName, null, null, alternating);
else
return CreateRow(propertyName, value.ToString(),
value.GetType().FullName , alternating);
}
```

```
TableRow CreateRow(string s1, string s2, string s3, bool alternating)
{
TableCell first = new TableCell();
first.CssClass = "l";
first.Text = Abbreviate(s1);
TableCell second = new TableCell();
second.Text = Abbreviate(s2);
TableCell third = new TableCell();
third.Text = Abbreviate(s3);
TableRow row = new TableRow();
if (alternating)
row.CssClass = "s";
row.Cells.Add(first);
row.Cells.Add(second);
row.Cells.Add(third);
return row;
}
private string Abbreviate(string s)
{
if (s == null)
return NullValue;
string retVal = s;
foreach (KeyValuePair<string, string> pair in s_abbreviationMap)
{
//
// We only get one replacement per abbreviation call.
// First one wins.
//
if (retVal.IndexOf(pair.Key) != -1)
{
string replacedValue = string.Format("<span class=\"abbrev\"
title=\"{0}\">{1}</span>", pair.Key, pair.Value);
retVal = retVal.Replace(pair.Key, replacedValue);
break;
}
}
return retVal;
}
//
// ASP.NET server side callback
//
protected void GoGetRoles(object sender, EventArgs ea)
{
string[] roles = Roles.Text.Split(';');
UpdateRolesTable(roles);
```

```
        }
        }
```

3. Save the file as Default.aspx.cs in the D:\inetpub\wwwroot\claimapp directory.

==========Thank You all==you can email me to get this file in pdf format or a link to download==========enayatmeer02@yahoo.com ;;;;;;;;Enayat Meer;;;;