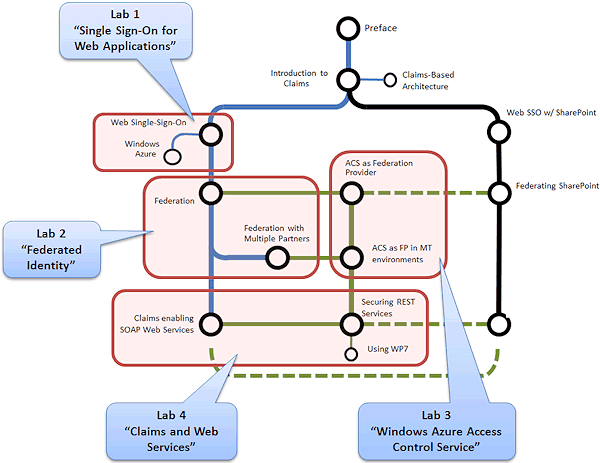
# Introduction

The Hands-On-Labs described in this document complement the guidance and examples in the patterns & practices book "*A Guide to Claims–based Identity and Access Control, 2nd Edition*". This guide is available to purchase, to download as a PDF file, or can be viewed online. For more information, see <http://claimsid.codeplex.com/>.

These Hands-On-Labs consist of a series of four individual labs, each of which focuses on a specific topic area from the guide. The following schematic shows how the labs relate to the chapters of the guide.

* 1. 

In more detail, the labs are:

* + Lab 1: "**Single Sign-On for Web Applications**". This lab demonstrates how you can make an existing application claims-aware and perform authorization based on claims presented by users when they are authenticated. It also demonstrates how you can work with WIF sessions, and publish the application to Windows Azure.
  + Lab 2: "**Federated Identity**". This lab demonstrates how you can set up federated authentication for an application, and how you can transform claims received from an identity provider or token issuer.
  + Lab 3: "**Windows Azure Access Control Service**". This lab demonstrates the use of Windows Azure AppFabric Access Control Service (ACS) as an authentication provider and token issuer. It shows how to configure ACS though the portal website, how to use the ACS management API, and how you can configure ADFS to trust ACS.
  + Lab 4: "**Claims and Web Services**". This lab demonstrates how you can use claims-based authentication in both SOAP and REST web services.

Each of the labs is divided into separate exercises related to the topics of the lab. The example code for the labs contains a separate Begin and End solution for each exercise. This means that you can run individual exercises if you wish, without needing to work through them all in order. However, many build on the knowledge gained in earlier labs and exercises, and you should consider working through them in order if you have time. In most cases the exercises should take only around 30 minutes each to complete.

* 1. **Note**: In some cases the End solutions we provide contain additional code that is not added during the tasks of the exercises, such as pop-up tool tips that provide information for users of the application (and which will help you to understand how they work).
  2. If you are using Internet Explorer 8 to view the example web pages, you may find that it caches pages after you log out of the applications so that it appears you are still logged on. To resolve this you can add directives to the applications to prevent Internet Explorer caching pages. You can add the following ASP.NET directive immediately after the existing **<%@Page ...>** directive in the Default.aspx pages:
  3. **<%@ OutputCache Location="None" %>**
  4. Alternatively, add all of the following META elements to any ASP.NET or HTML page:
  5. **<META Http-Equiv="Cache-Control" Content="no-cache">**
  6. **<META Http-Equiv="Pragma" Content="no-cache">**
  7. **<META Http-Equiv="Expires" Content="0">**

# Pre-requisites for the Labs

* 1. The following programs and components are required to use these labs:
  + [Microsoft Visual Studio 2010](http://msdn.microsoft.com/vstudio/products/) (any version except for Express editions)
  + [Microsoft .NET Framework 4.0](http://go.microsoft.com/fwlink/?linkid=186916) (automatically installed with Visual Studio 2010)
  + [Windows Identity Foundation](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=eb9c345f-e830-40b8-a5fe-ae7a864c4d76) version 4.0
  + [Windows Identity Foundation SDK](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=C148B2DF-C7AF-46BB-9162-2C9422208504)
  + [Windows Azure Tools and SDK](http://msdn.microsoft.com/en-us/evalcenter/ee460823.aspx)
  1. **Note**: You must have a Windows Azure account to perform all of the exercises in Lab 3 and Exercise 2 of Lab 4. Some of the exercises also require the Windows Azure Tools for Visual Studio to be installed. The dependency checker (described later in this document) will check that you have the Windows Azure Tools, and help you to install them if necessary.
  2. Some of the exercises in the labs contain a task where you examine the content of the requests and responses during the authentication process. The labs show how you can do this using the [Fiddler](http://www.fiddlertool.com/fiddler/version.asp) utility with [Internet Explorer 9](http://www.beautyoftheweb.com/) (IE9). If you have not yet installed IE9 you can use Fiddler with another non-Microsoft web browser, or use any other HTTP intercepting proxy or packet inspection tool such as [Microsoft Network Monitor](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=983b941d-06cb-4658-b7f6-3088333d062f&displaylang=en). Alternatively, if you have installed IE9, you can use the developer tools it includes, but these are less versatile and you will not be able to examine all of the requests and responses.
  3. **Note**: If you are using Internet Explorer 8 to work with the examples in these labs, you will find that Fiddler does not detect packets when you specify the URL of the application using **localhost**. The suggested way to resolve this in the Fiddler help files is to use the DNS or network name of your computer. However, this will not work with the examples we provide because the authentication configuration specifies **localhost** URLs. If you decide to use Fiddler, you must also use Internet Explorer 9 or a non-Microsoft browser to view the example applications.
  4. You can download Fiddler free from <http://www.fiddler2.com/fiddler2/version.asp>. If you have not used Fiddler before, you should watch the short but very useful QuickStart video "*Introduction to Fiddler*", available at <http://www.fiddler2.com/Fiddler/help/video/default.asp>.

## Configuring Fiddler for the Labs

* 1. If you decide to use Fiddler to examine packet flow in the labs, follow these steps to configure it so that you can see the contents of the HTTPS requests and responses, and so that extraneous requests and responses are filtered from the results.

To configure Fiddler for the labs

* 1. Open Fiddler from your **Start** menu, or by clicking the icon it adds to your browser toolbar.
  2. Open the **File** menu and ensure **Capture Traffic** is turned on (the menu item has a tick against it).
  3. Open the **Rules** menu and click **Hide Image Requests** so that it has a tick against it.
  4. Open the **Rules** menu again and click **Hide HTTPS CONNECTs** so that it has a tick against it.
  5. Open the **Tools** menu and click **Fiddler Options**.
  6. Click the **HTTPS** tab and ensure **Capture HTTPS CONNECTs** is checked. Select the **Decrypt HTTPS traffic** option so that it is also checked.
  7. You will see a WARNING dialog asking if you want to trust the root certificate that Fiddler uses to intercept HTTPS packets. We suggest you click **No**.
  8. Click **OK** to close the Fiddler Option dialog. This completes the configuration for Fiddler.
     1. If you click **Yes** in the certificate warning dialog, Fiddler adds a certificate named **DO\_NOT\_TRUST\_FiddlerRoot** to your certificate store. You can use the buttons in this dialog to examine the certificate by exporting it to your desktop, or to remove it from your certificate store after use if you do decide to install it. The only difference is that, if you do not trust the certificate, you will see the warning again each time you initially connect when capturing HTTPS traffic.

# Installing the Examples

1. You can download the code samples for these Hands-On-Labs from <http://claimsid.codeplex.com/>. The download file is a Microsoft Installer (MSI) package that installs the sample code on your computer. Before you run the installer, right-click on it in Windows Explorer and click **Properties**. If you see an **Unblock** button, click this. Then double-click the file to install the samples.
   1. After the sample have been installed, open the **Setup\DependencyChecker** subfolder in Windows Explorer and double-click the file **DependencyChecker.exe**. This program scans your computer to ensure you have all the required programs, components, and settings required to run the examples. It provides a link to download any missing components, and can also perform any required configuration changes. You must have the [.NET Framework version 4.0](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en) installed to run the Dependency Checker.

# Running the Examples

The example code for these labs contains a separate folder for each lab. Within this folder is the document that describes the lab and contains the step-by-step instructions for using the lab. This folder also contains a folder named **Source**, within which you will find separate subfolders for each exercise in the lab. The exercise folders contain the **Begin** and **End** solution folders, plus separate folders for any shared code or other assets required for the lab.

You **must** run Visual Studio as an administrator when opening many of the solutions in these labs. If you simply double-click the solution file, Visual Studio may fail to load the projects.

To run Visual Studio as an administrator

* 1. Copy the Visual Studio shortcut from your **Start** menu onto the desktop.
  2. Right-click the shortcut icon and click **Properties**.
  3. In the **Shortcut** tab of the Properties window click the **Advanced** button.
  4. In the Advanced Options dialog, set the **Run as administrator** option.
  5. Click **OK** twice to close the dialogs.
  6. Use this shortcut to start Visual Studio and then load each solution using the options on the **File** menu.
     1. **Note:** You *cannot* drag and drop solution files onto the Visual Studio IDE.

## The HTTPS Root Agency Certificate

* 1. The Dependency Checker installs a certificate named "Root Agency" in the Internet Explorer certificate store. This is used to enable HTTPS for the examples. It is not installed in the certificate store of other browsers you may have installed on your computer, and you will see a warning that the certificate is not trusted when you open the sample in a browser other than Internet Explorer. If you wish to use a different browser you can install the certificate manually, or ignore the security warning when you navigate to the example.

# About the Token Issuers

* 1. All of the labs will work without requiring access to an Active Directory Federation Server (ADFS). They use simple implementations of a mock Security Token Service called Self STS. This is intentional to allow the labs to be used when three is no ADFS instance available.
  2. However, to demonstrate how you can integrate your applications with ADFS each lab ends with an exercise that shows how you can modify the final solution in the lab to use ADFS instead of the Self STS. To carry out these exercises you must have an ADFS 2.0 STS available over your network.

# Integrating with Active Directory Federation Services

* 1. Most of the labs contain a final optional exercise that uses Active Directory Federation Services (ADFS) as an identity provider and token issuer. You do not require ADFS to use the other exercises in these labs. However, if you do wish to perform the optional exercises that show how to modify the examples to use ADFS you must have an installation of ADFS 2.0 available that you can configure.
  2. You can download ADFS 2.0 from <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=118c3588-9070-426a-b655-6cec0a92c10b&displaylang=en> and install it on Windows Server 2008 or Windows Server 2008 R2. Configuration for relying parties and rules is described in the relevant sections of the exercises in each lab.
  3. **Note**: Windows Server 2008 R2 contains a built-in role named Federation Server, but this is ADFS version 1.1 and not version 2.0. For these labs you should not enable the built-in Federation Server role; instead download and install the version of ADFS 2.0 for Windows Server 2008 R2.

A useful guide "*AD FS 2.0 Federation with a WIF Application Step-by-Step Guide*" that will help you to set up ADFS 2.0 is available from <http://www.microsoft.com/downloads/en/details.aspx?familyid=BB9AB270-473B-4852-B26E-031A88EDD113&displaylang=en>. This guide walks through the installation process and setting up a simple test application.

## ADFS Configuration

* 1. The default behavior for ADFS when a user is authenticating against Active Directory is to use Windows Integrated Authentication — this means that in the lab, you won't see the sign-in process happening or have the opportunity to select the Windows user account that you want to use. For these hands-on-labs, you should change the default behavior of ADFS so that it uses forms-based authentication when a user authenticates with Active Directory: this way you'll be able to see the Active Directory sign-in process and select the username you want to use.
  2. In a typical *intranet* environment, you'd want to use Windows Integrated Authentication with Active Directory. You'd use forms-based authentication against Active Directory from the *internet*.

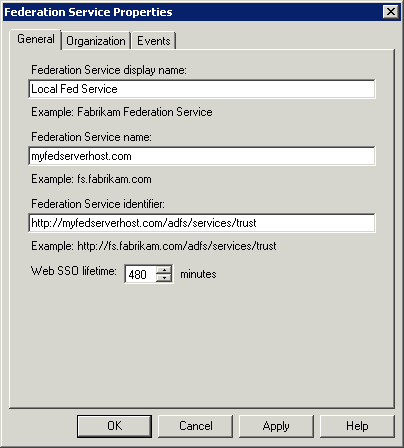
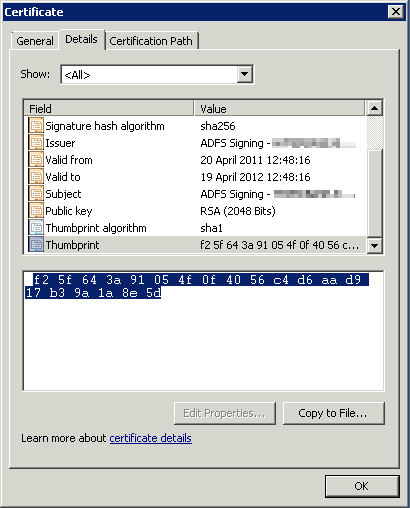
To configure ADFS to use forms-based authentication with Active Directory

* 1. On the machine where you have installed ADFS, locate the ADFS **Web.config** file. Ina typical installation this is located at **C:\intepub\adfs\ls**.
  2. Open this **Web.config** file in a text editor and locate the **microsoft.identityServer.web** section. Modify this section to place **Forms** at the top of the list of **localAuthenticationTypes** as shown in the following snippet.
     1. XML
     2. <microsoft.identityServer.web>
     3. <localAuthenticationTypes>
     4. **<add name="Forms" page="FormsSignIn.aspx" />**
     5. <add name="Integrated" page="auth/integrated/" />
     6. <add name="TlsClient" page="auth/sslclient/" />
     7. <add name="Basic" page="auth/basic/" />
     8. </localAuthenticationTypes>
     9. …
  3. Save the **Web.config** file.

## Information Required for the Lab Exercises

* 1. You will need to record some information about your ADFS for the exercises that use ADFS. You need the URL of the ADFS service, the federation service identifier, and the thumbprint of the certificate the service uses to sign the tokens. Use the following procedure to obtain this information.

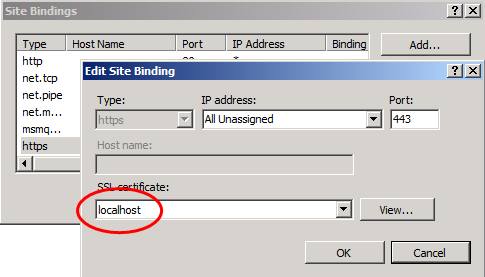
To obtain the required ADFS information for the optional exercises

* 1. Open the **AD FS 2.0 Management** console from your server's **Start** menu.
  2. Right-click the top-level **AD FS 2.0** folder and click **Edit Federation Service Properties**.
  3. Make a note of the value for the **Federation Service name** (the second textbox on the **General** tab page). This is the fully-qualified domain name of the ADFS service. It is the URL you will use in relying party applications that access this ADFS issuer.
     1. 
  4. Make a note of the value for the **Federation Service identifier** (the third textbox on the **General** tab page). This is the name identifier of the ADFS service. It is the value you will use for the **name** attribute of a thumbprint in your Web.config file.
  5. Expand the **AD FS 2.0** folder, expand the **Service** folder, and select the **Certificates** folder. In the list of certificates, double-click the certificate in the **Token-signing** section to display the certificate details.
  6. Click the **Details** tab in the Certificate dialog and select **Thumbprint** in the list of fields. Make a note of the value shown below the list of fields for the thumbprint.
     1. 
     2. If you want to copy the value into your Web.config file, select it but do not include the first "space" character. Paste the selected text into an ASCII text editor (not Windows Notepad or Visual Studio) and then copy it again into your Web.config file. The first "space" character is actually a non-standard character, not a space, and is not visible in editors that use Unicode instead of ASCII. It will cause an error such as "*The issuer of the security token was not recognized...*" if this non-visible character is included in the thumbprint string in your Web.config file.
  7. Some of the exercises use the Visual Studio STS Wizard to set the federated identity properties defined the Web.config file of the example applications. This wizard must be able to access the federation metadata exposed by ADFS. By default, access to this is enabled. If you find that the STS Wizard cannot locate the metadata information, use the following procedure to check that access is enabled.

To check access to metadata is enabled in ADFS 2.0

* 1. Open the **AD FS 2.0 Management** console from your server's **Start** menu.
  2. Expand the **AD FS 2.0** folder, expand the **Service** folder, and select the **Endpoints** folder.
  3. Scroll to the **Metadata** section ensure the **Enabled** column for all three items contains **Yes**.
     1. The STS Wizard In particular requires access to the item **/FederationMetadata/2007-06/ FederationMetadata.xml**.
  4. To enable an item where it is not enabled, right-click it and select **Enable**.

## Installing ADFS on the Same Machine as the Labs

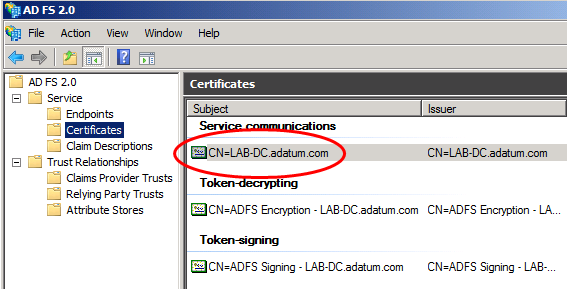
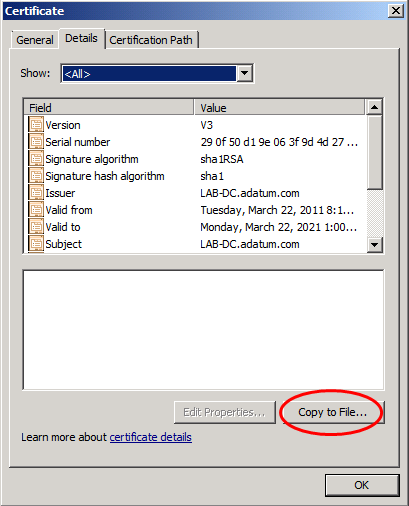
* 1. You can choose to complete these Labs on the machine where you have installed ADFS, or choose to install ADFS on a separate server. If you choose to install these Labs on the machine where you have installed ADFS, you should be aware of which certificate is used to configure the HTTPS binding in IIS.
  2. You should run the Dependency checker tool for these Labs after you have installed ADFS. This will set the HTTPS binding to use the test **localhost** certificate included with these Labs — this is the certificate you want to use. You can verify this setting by examining the **Default Web Site** bindings in IIS, they should show the **localhost** certificate being used:
     1. 

With this setting, you will be able to browse to all the web sites in the Labs without receiving a certificate warning. The exceptions to this are the exercises that use ADFS (Lab 1 Exercise 5, Lab 2 Exercise 3, and Lab 3 Exercise 4): in these exercises, you will receive warning messages about the certificate when you navigate to the ADFS site.

## Installing ADFS on a Separate Machine

* 1. You can choose to complete these Labs on the machine where you have installed ADFS, or choose to install ADFS on a separate server. If you choose to install ADFS on a separate server (or in a separate Virtual Machine), there are some additional configuration steps to complete before you begin the lab exercises that use ADFS. You must install the ADFS service certificate on the machine where you are completing the labs.

To install the ADFS service certificate on your development machine

* 1. On the machine where you have installed ADFS, open the **AD FS 2.0 Management** console, expand the **Service** node, and click **Certificates**. You will see the certificate that ADFS uses for service communications. The **CN** name may be different in your environment.
     1. 
  2. Right click on the service communication certificate and click **View Certificate**, the click the **Details** tab, and then click the **Copy to File** button.
     1. 
  3. On the **Welcome** page of the **Certificate Export Wizard**, click **Next**, make sure that **No, do not export the private key** is selected, and then click **Next**. On the **Export File Format** page, make sure that **DER encoded binary X.509 (.CER)** is selected, and then click **Next**. On the **File to Export** page, click **Browse**, select a suitable location to save the file, name the file **ADFSServiceCert.cer**, then click **Save**, then click **Next**, and the click **Finish**. Click **OK** to dismiss the message **The export was successful**, and then click **OK** to close the **Certificate** dialog.
  4. Copy the **ADFSServiceCert.cer** file to the development machine where you will be completing the labs.
  5. Start **MMC** and install the **Certificates snap-in** to manage certificates for the **Computer Account**. Make sure you are managing certificates on the local computer.
  6. Expand the **Certificates** node, expand the **Trusted People** node, and then click **Certificates**. If you have already run the **Dependency Checker** for the labs, you will see certificates for **Adatum** and **Litware**. Right-click on the **Certificates** node inside the **Trusted People** node, point to **All Tasks** and then click **Import**.
  7. In the **Certificate Import Wizard**, on the **Welcome** page, click **Next**. On the **File to Import** page, click **Browse**, locate the **ADFSServiceCert.cer** file that you copied to this machine in step 4, and click **Open**. In the **Certificate Import Wizard**, on the **File to Import** page, click **Next**, make sure that the **Trusted People** certificate store is selected, click **Next**, and then click **Finish**. Click **OK** to dismiss the message **The import was successful.** The certificate is now installed. The name of the certificate may be different in your environmnent.
     1. 