**Windows Phone 7 Sample App Readme**

This sample illustrates early thinking about how a Windows Phone 7 Silverlight application can make use of ACS to call a REST service.    
  
In the sample a directory service is locally hosted that returns a list of contact information. The phone app is then used to access this service. Users are able to authenticate to the service by signing into one of the identity providers Contoso has configured at their ACS Service Namespace.

The sample code is available for [download](http://acs.codeplex.com/releases/view/55185).

**Prerequisites**

To run this sample you will need:

1. Visual Studio 2010 (any version)
2. [Windows Phone Developer Tools RTW](http://go.microsoft.com/fwlink/?LinkId=185968)

**Setting up the ACS Service Namespace and Contact Service**

This section contains information on how to configure a Service Namespace and service that can be used with the sample application. It is recommended to first walkthrough [Getting Started](http://acs.codeplex.com/wikipage?title=Getting%20Started&referringTitle=ASP.NET%20Simple%20Forms) before reading this section.

**Additional Prerequisites**

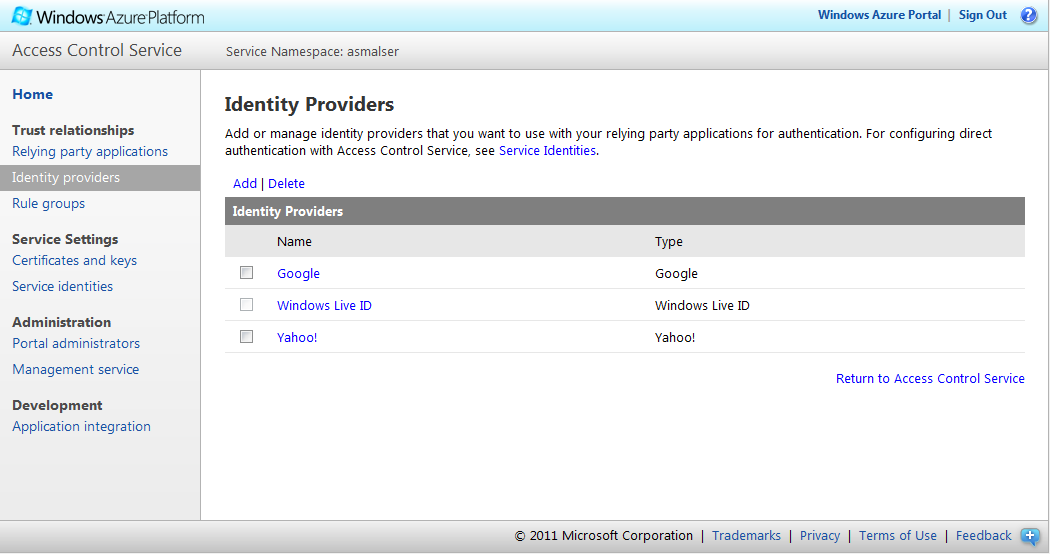
To run this sample you will need:

1. DPE.OAuth project from the [FabrikamShipping Demo](http://www.fabrikamshipping.com/), this is packages as part of the [Source Code](http://code.msdn.microsoft.com/fshipsaassource).

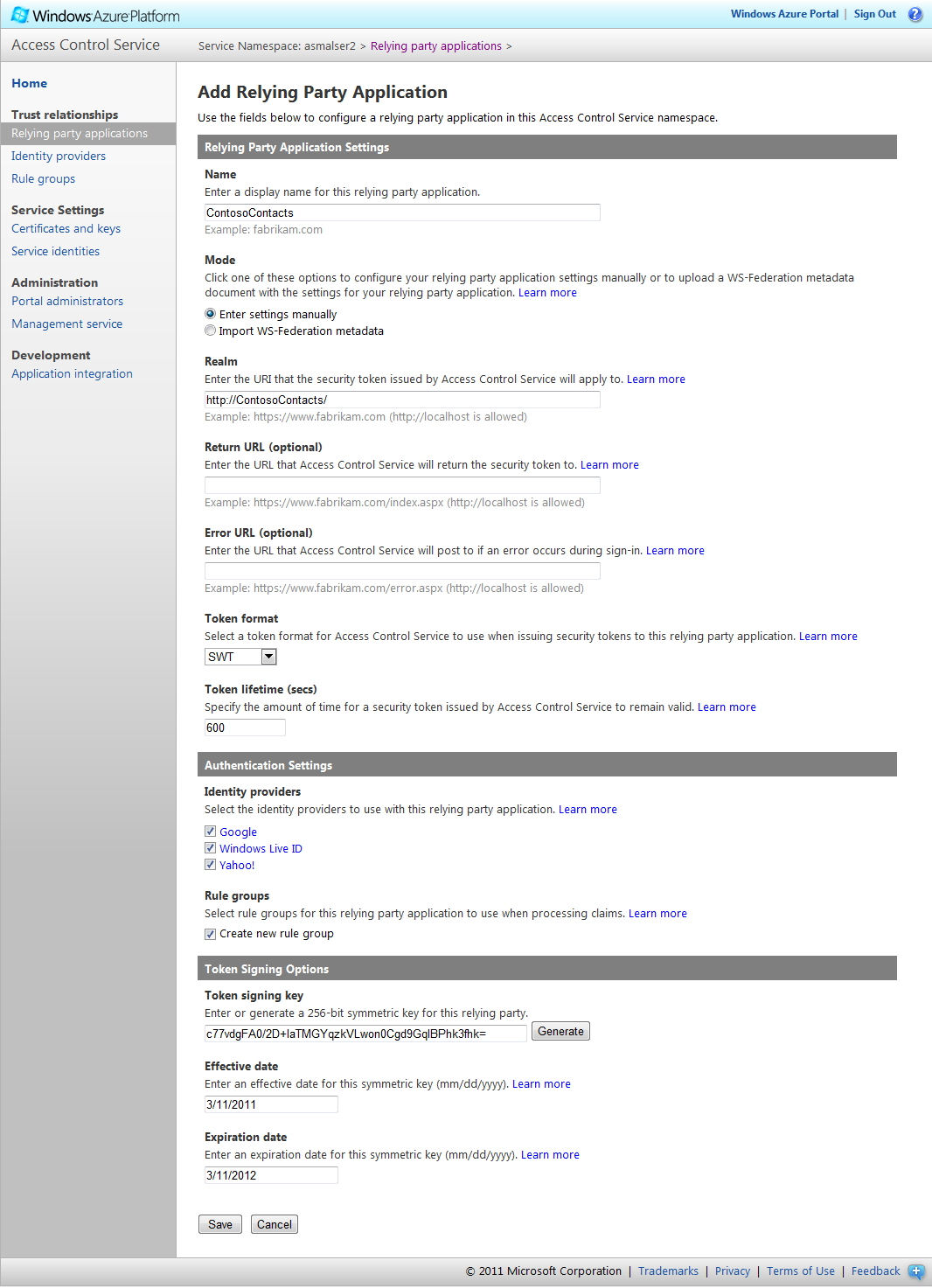
**Setup Steps**

1. Browse to <http://portal.appfabriclabs.com> and sign in.

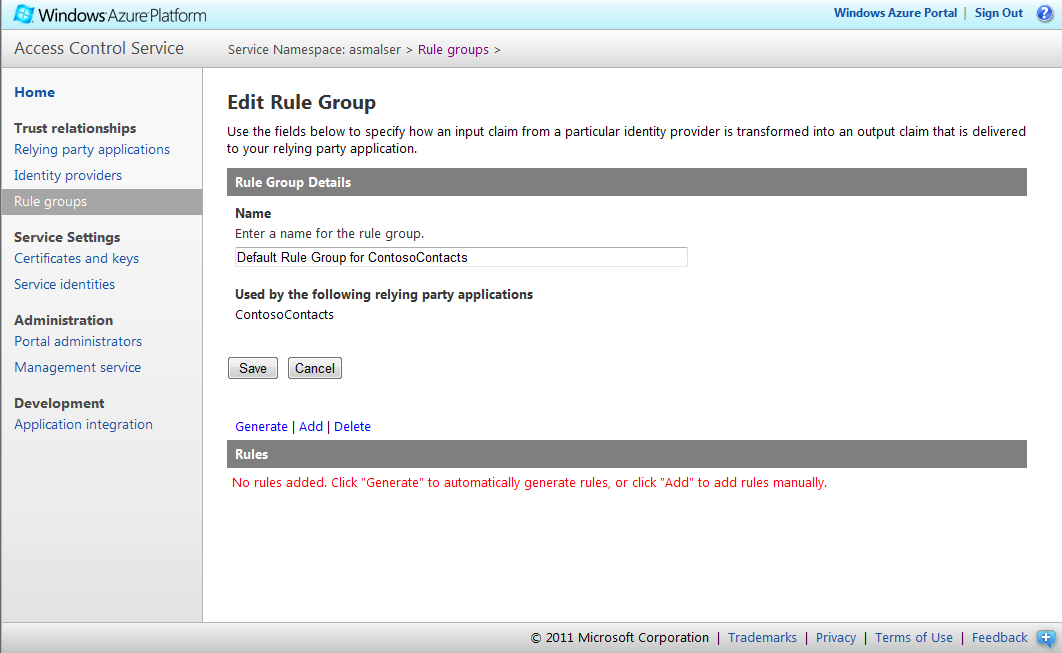
2. Go to the Identity Providers section and add Yahoo! and Google.



3. Go to the Relying Party Applications section and click "Add**"**.   
  
4. Enter the following information. Select "SWT" as the token type, and click "Generate" to create a key below. When complete, click the “Save” button.



5. Go to the Rule Groups section. Click on Default Rule Group for ContosoContacts, and then click Generate to create a default set of rules.



6. Add the DPE.OAuth project from the FabrikamShipping Demo [Source Code](http://code.msdn.microsoft.com/fshipsaassource) to this solution.   
  
7. Add a reference to the DPE.OAuth project to the CustomerInformationService project.   
  
8. In the CustomerInformationService project, open Web.config and make the following modifications:

1. Set the value for IssuerIdentifier by replacing “[Service Namespace](http://acs.codeplex.com/acs/wikipage?title=Service%20Namespace&referringTitle=ACS%20Windows%20Phone%20Sample)” with the service namespace you have configured with ACS. ( for example: <https://contoso.accesscontrol.appfabriclabs.com/> )
2. Set the ServiceKey to be the key which was generated in step 4 above.

9. In the ContosoContactsApp project, open SignIn.xaml and make the following modifications:

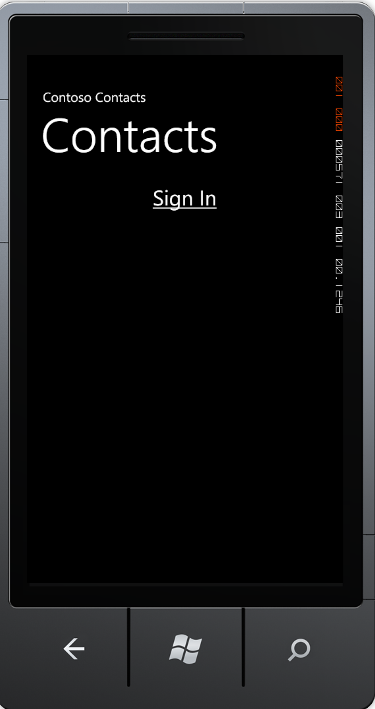
      1.  Set the service namespace under AccessControlServiceSignIn  by replacing “[Service Namespace]”  with the service namespace you have configured with ACS. ( for example: <https://contoso.accesscontrol.appfabriclabs.com/> )

      2.  Set the Realm to the value specified in Step 4 above.

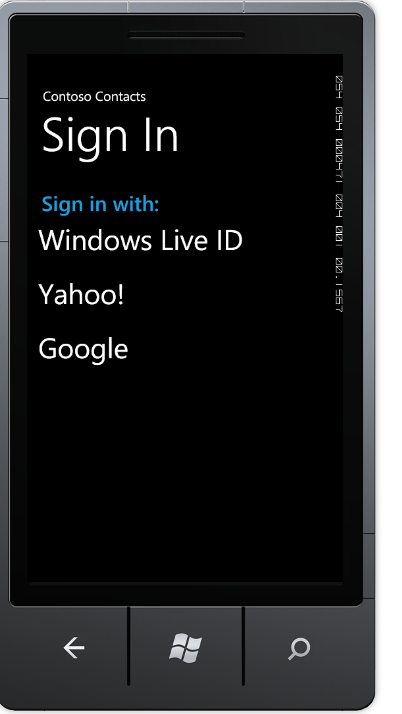
10. Run the phone application as shown in the following section of this document.

**Running the Phone App Sample**

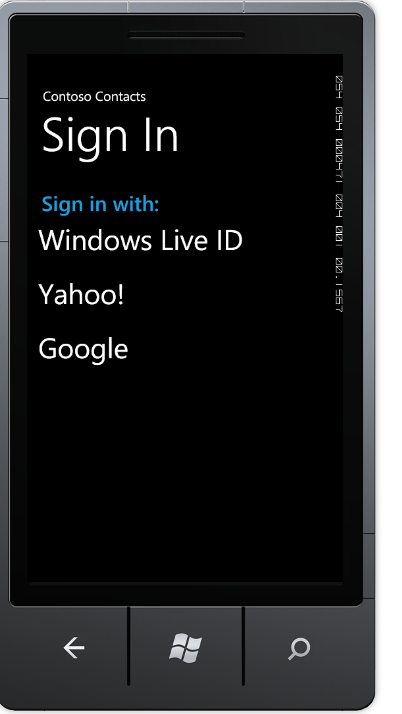
After the ACS Service Namespace and service endpoint have been set up successfully as described by the previous section,   
  
1) Open the sample in Visual Studio (.\ContosoContactsApp\ContosoContactsApp.sln ) in administrator mode  
  
2) Press “F5“ to start the application. When the Windows Phone emulator has started the “Contoso Contacts” app will load, you should see the following:



3) Click on the “Sign In” hyperlink. This will navigate to the Sign In page, hosted in the Silverlight application. Shown below:



4) This page in the app uses the AccessControlServiceSignIn control, from the sample; it is part of the SL.Phone.Federation project.

5) Next, click an identity provider from the list. Once the link has been clicked the identity providers login page will be shown.   
  


6) Sign into the identity provider.   
After signing into the identity provider a token will be issued by ACS, which is used to authenticate to the contact service. The Image below shows the list of contacts, from the service.



7) As an additional feature, the sample will store the token in persistent storage. This allows the service to be called multiple times across invocations of the app. The token can be cleared by clicking the “Sign Out” link on the application bar at the bottom of the app, which is shown below:

