Setup Forms Based Authentication Under SharePoint 2010

Introduction

This document will cover the steps for installing and configuring Forms Based Authentication (FBA) on a SharePoint 2010 site. The document is presented in multiple steps:

Step#1:	Prerequisites and assumptions
Step#2:	Installing the FBA database
Step#3:	Configure IIS to access the FBA database
Step#4:	Activate FBA on the SharePoint Web Services website
Step#5:	Create a SharePoint application that is FBA enabled (Claims)
Step#6:	Activate FBA on the newly created SharePoint application
Step#7:	Configure SuperUser and SuperReader accounts

Step#1: Prerequisites and assumptions

Prerequisites:

- SharePoint 2010 (Enterprise or Foundation) is installed, up-to-date, and functional
- You have administrative rights to the server(s) hosting the SharePoint applications
- You are familiar with and have access to SharePoint 2010's central administration site
- You are familiar with and have access to the SQL Database which will house the FBA database

Assumptions:

This document presents the implementation of Forms Based Authentication from the perspective of least privileged security. As such, the guide will walk through the steps necessary to implement FBA using integrated security rather than SQL User Authentication.

Step#2: Installing the FBA database

Why?

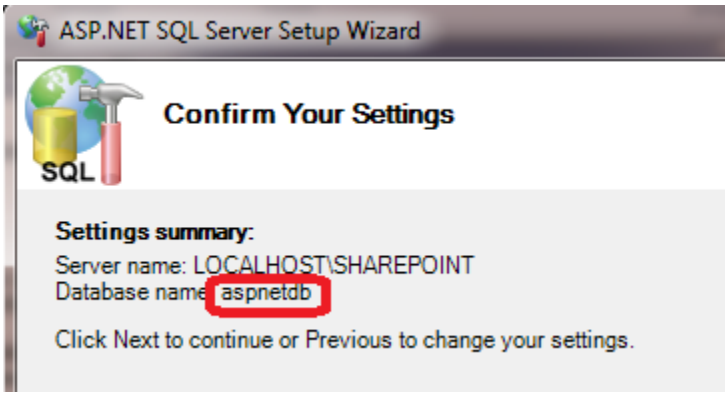
Forms Based Authentication requires a SQL database to store the user logon information.

Details:

1. Launch the SQL Server Setup Wizard via the following command line
   C:\Windows\Microsoft.NET\Framework\V2.0.50727\aspnet_regsql.exe

2. Follow the wizard steps to install and configure the membership database.

3. **IMPORTANT**: Note the database name being created.
   The database name will be listed on the **Confirm Your Settings** wizard screen
   In our example, we used the default of **aspnetdb**



Step#3: Configure IIS to access the FBA database

Why?

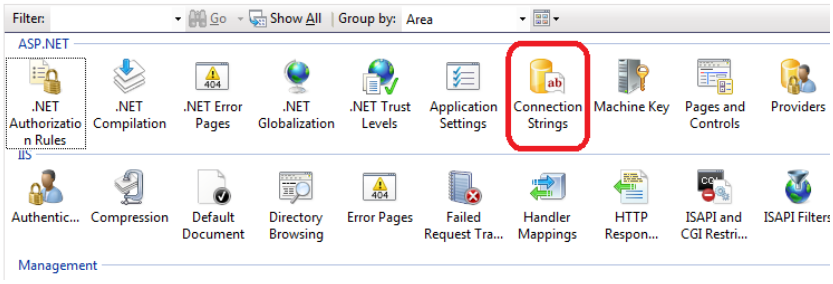
The Forms Authentication data is stored in the SQL Server created in step#2. IIS needs to be configured to know where to look for the database.

Details:

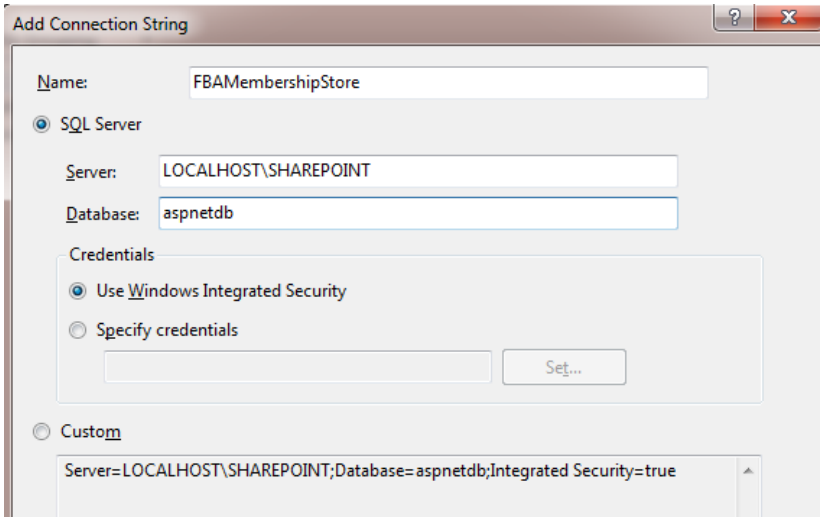
1. Launch **Internet Information Services (IIS) Manager**

2. Select the top level (machine) entry (Usually named after the server)
   Why here? Creating the connection string at the top level allows the connection to be "inherited" by all websites.

3. On the home page (located in the middle of the IIS Manager), double click the **Connection Strings** icon



4. Add a new connection to point to the SQL Server and database the membership store is stored in.
   a. **IMPORTANT**: Note the name of the connection. We use **FBAMembershipStore**
   b. The database name must match the membership store database name from step#1
   c. Be sure to check **Use Windows Integrated Security**



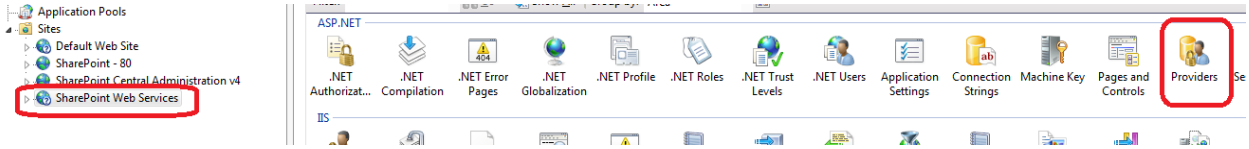
Step#4: Activate FBA on the SharePoint Web Services website

Why?

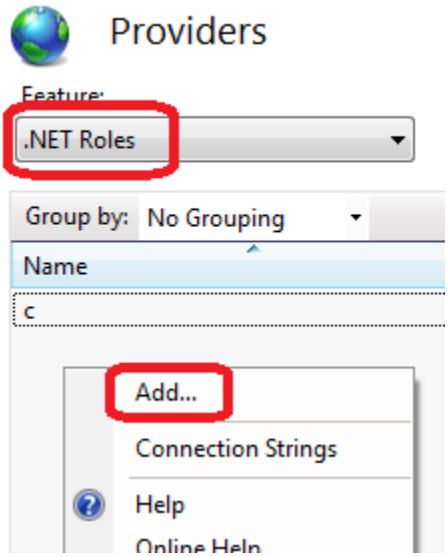
The web service also need to authenticate users. If you do not give the web service site access to the FBA membership store, your FBA will not work

Details

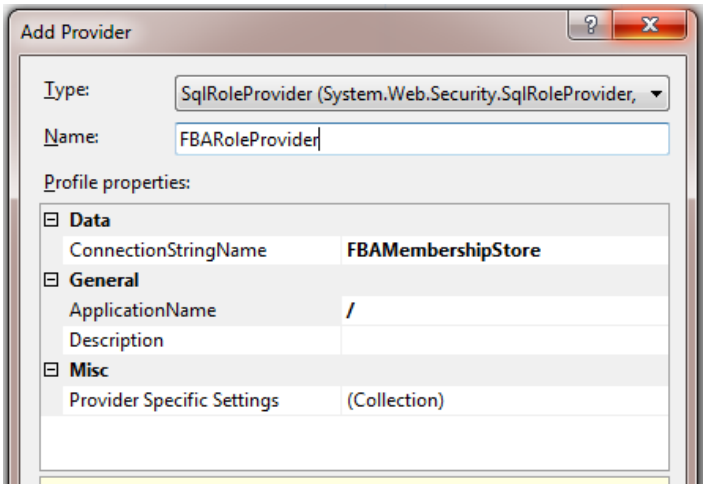
1. Select **Providers** for the SharePoint Web Services site



2. Select **.NET Roles** from the **feature** selector and right click in the screen. Click **Add** on the right click menu.



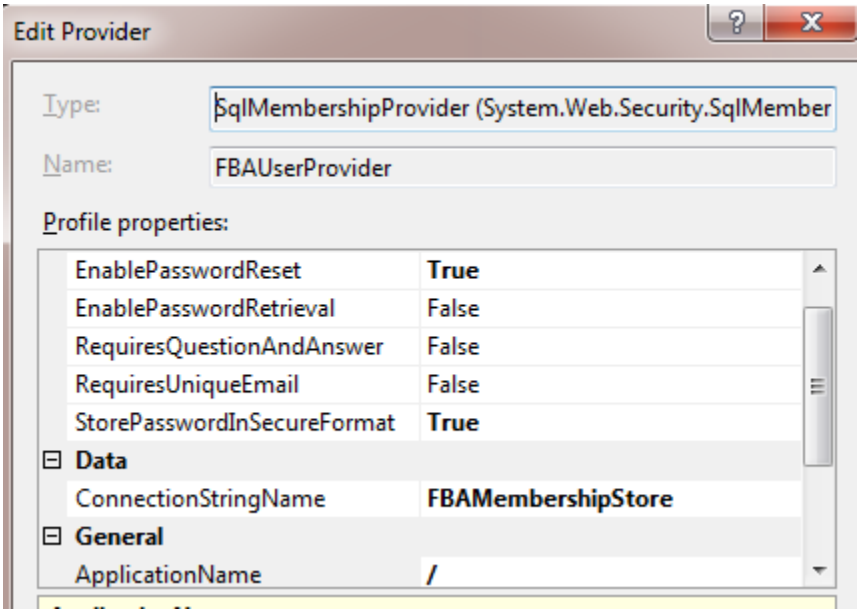
3. Create a new role provider



   a. Set type to **SqlRoleProvider**
   b. Name the provider. We use **FBARoleProvider**
   c. Select the connection string you created in Step#3
   d. Set the ApplicationName to **/**
      Why? See http://weblogs.asp.net/scottgu/archive/2006/04/22/Always-set-the-_2200_applicationName_2200_-property-when-configuring-ASP.NET-2.0-Membership-and-other-Providers.aspx

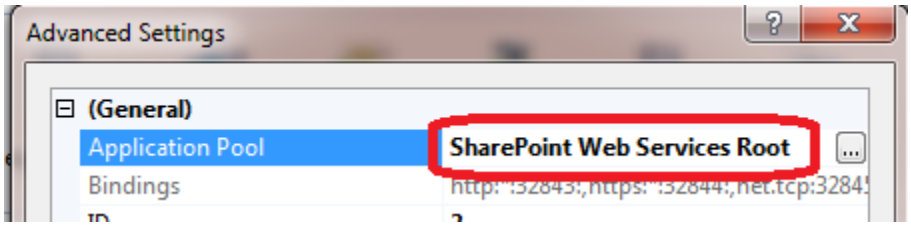
4. Select **.NET Users** from the **feature** selector and right click in the screen. Click **Add** on the right click menu.

5. Create a new user provider

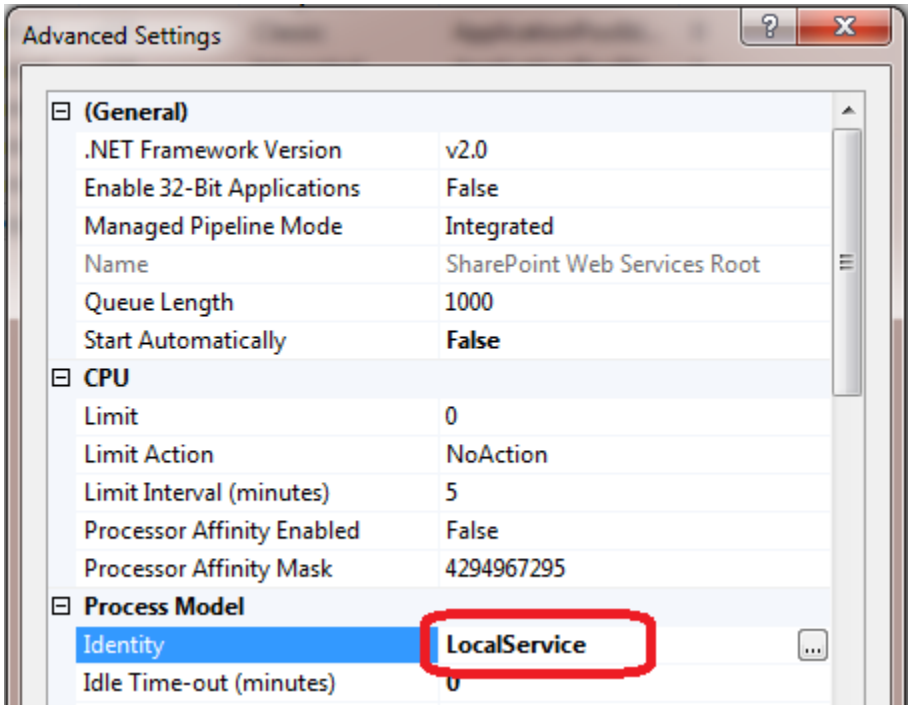


   a. Set type to **SqlMembershipProvider**
   b. Name the provider. We use **FBAUserProvider**
   c. Select the connection string you created in Step#3
   d. Set the ApplicationName to **/**
   e. Set the StorePasswordInSecureFormat
      IMPORTANT: **If you select True** (and you should)**, See "Encryption, FBA, and IIS Oh My!" at the end of this document for additional required steps.**

6. Determine the **Application Pool** credentials the SharePoint application is running under
    a. Right click on the SharePoint Web Services
    b. Click **Manage Web Site** -> **Advanced Settings** from the right click menu
    c. Note the **Application Pool** name

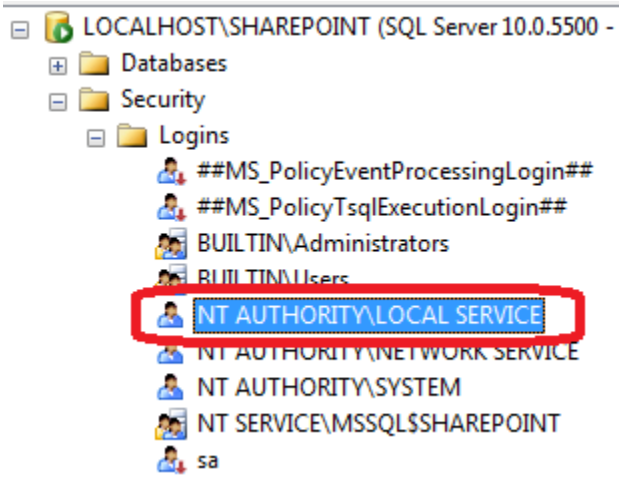


    d. Open the Application Pool **Advanced Settings** and note the **Identity** it is running under

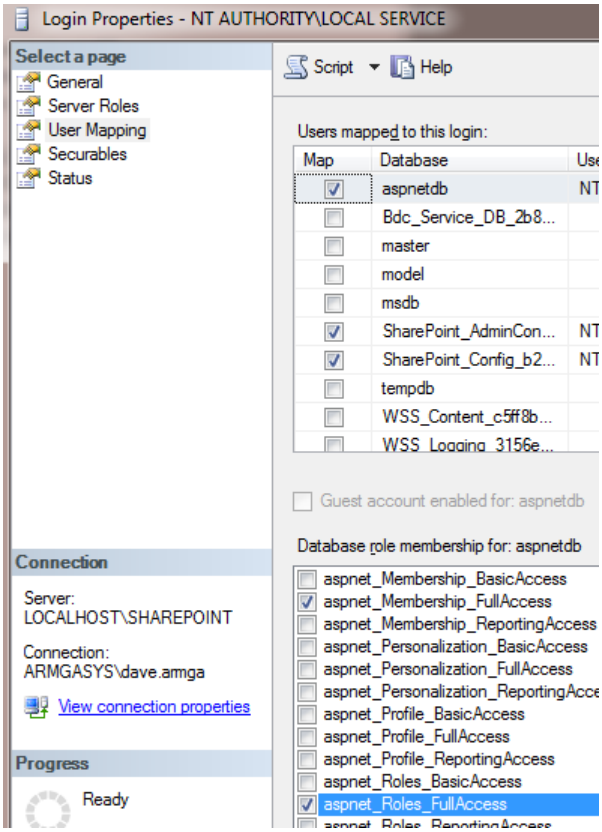


7. Launch **SQL Server Management Studio**

8. Under **Security** -> **Logon** verify the application pool identity (user) exists as a valid SQL Server logon. If not, create the user.



9. Grant the user the following roles on the **aspnetdb** database:
   a. aspnet_Membership_FullAccess
   b. aspnet_Roles_FullAccess



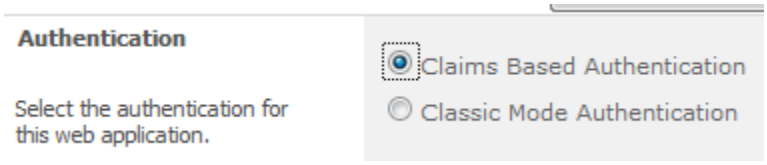
Step#5: Create a SharePoint application that is FBA enabled (Claims)

Why?

Well, because this article would be worthless if we didn't actually create a new SharePoint application with FBA ☺

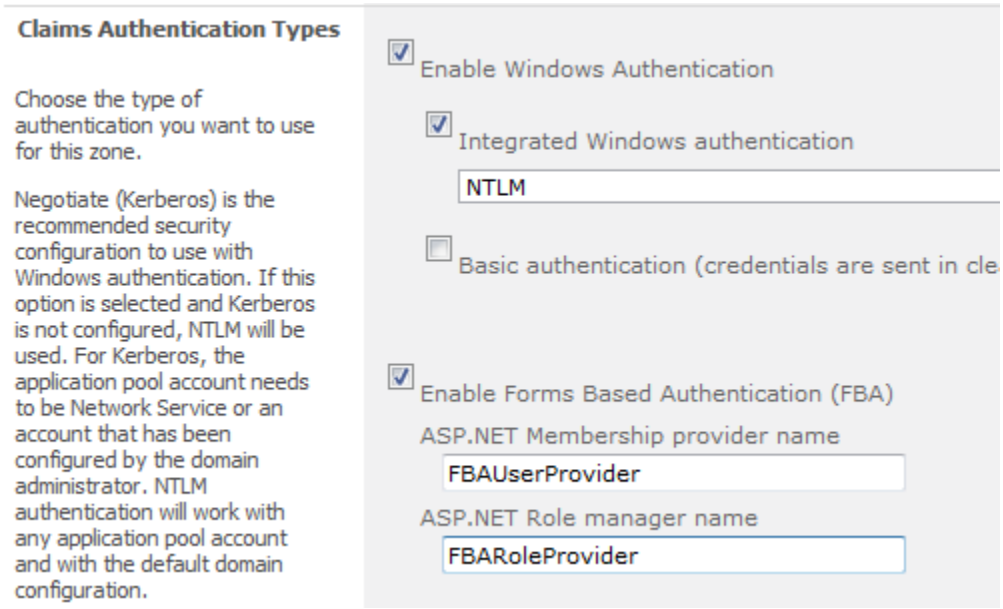
Details

1. Launch **SharePoint 2010 Central Administration**
2. Navigate **Application Management** -> **Manage Web Applications**
3. Click **New** on the ribbon
4. Select **Claims Based Authentication** radio button



5. Configure the **Claims Authentication Types** section as follows
   a. Check the **Enable Forms Based Authentication (FBA)** checkbox
   b. In the **ASP.NET Membership provider** name field, enter **FBAUserProvider**
   c. In the **ASP.NET Role Manager** name, enter **FBARoleProvider**

   IMPORTANT: If you used different names, please substitute those names as appropriate



6. Configure all other settings per your individual needs

Step#6: Activate FBA on the newly created SharePoint application

Why?

Well, because this article would be worthless if we didn't actually create a new SharePoint application with FBA ☺

Details

Repeat all actions under **Step#4: Activate FBA on the SharePoint Web Services website** replacing out the SharePoint Web Services website with the SharePoint application you created in Step#5.

Step#7: Configure SuperUser and SuperReader accounts

Why?

Per Microsoft:

The default Portal Super Reader account is NT Authority\Local Service, which is not correctly resolved in a claims authentication application. As a result, if the Portal Super Reader account is not explicitly configured for a claims authentication application, browsing to site collections under this application will result in an "Access Denied" error, even for the site administrator

This does not make for a good user experience!

Details

1. Create two new active directory accounts to represent the Super User & Reader accounts
   I.E. **YourDomain\SuperUser** and **YourDomain\SuperReader**

2. Launch **SharePoint 2010 Central Administration**

3. Navigate **Application Management** -> **Manage Web Applications**

4. Select the site created under **Step#5: Create a SharePoint application is FBA enabled**

5. Click **User Policy** on the ribbon

6. Click **Add Users**

7. Select **(All zones)** and click **Next**

8. Enter the appropriate user.
   Set the permissions to **Full Control** for the super user account
   Set the permissions to **Full Read** for the super reader account

9. Click **Finish**

10. Repeat steps 6 – 8 for both the super user and super reader accounts

<Continued on next page>

11. Launch **SharePoint 2010 Management Shell**

12. Enter the following script commands:

```
$wa = Get-SPWebApplication -Identity "YourWebApplicationName"
$wa.Properties["portalsuperuseraccount"] = "i:0E#.w|YourDomain\SuperUser"
$wa.Properties["portalsuperreaderaccount"] = "i:0#.w|YourDomain\SuperReader"
$wa.Update()
```

Please note:
-> replace YourWebApplicationName with the appropriate name.
-> Replace the SuperUser and SuperReader accounts with the accounts you created
-> Do not forget to preface the accounts with **i:0#.w|** (I.E. these should *exactly* match the accounts displayed by SharePoint User Policy list)

See: http://technet.microsoft.com/en-us/library/ff758656.aspx for more details

Additional Considerations

If you use SharePoint Central Administration for configuring user security, alerts, and other user centric functions (and you will), it is highly recommended you repeat all actions under **Step#4: Activate FBA on the SharePoint Web Services website** replacing out the SharePoint Central Administration application.

This step will configure Central Administration to have access to the FBA Membership Store.

Encryption, FBA, and IIS Oh My!

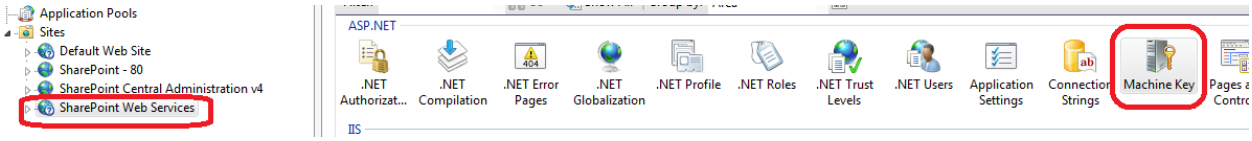
Why?

So you decided encryption of passwords was a good thing. Good for you! Encryption, however, requires keys to encrypt and decrypt data. In the case of IIS, those keys are (by default) randomly generated. We need to configure all SharePoint Applications which use the FBA Membership Store to use the same encryption and decryption keys.

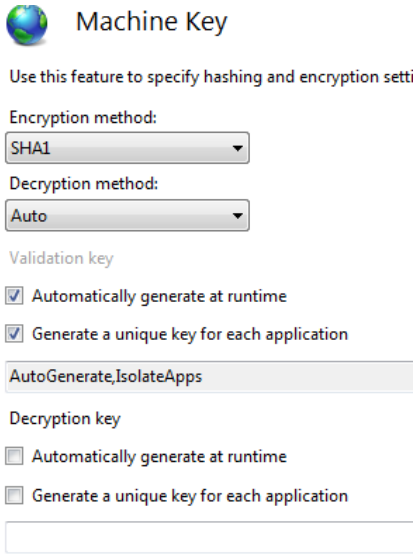
Details

1. Launch **Internet Information Services (IIS) Manager**

2. Select the **SharePoint Web Service** application and open the **Machine Key**



3. Uncheck **Automatically generate at runtime** and **Generate a unique key for each application** under the **Decryption Key** section



4. Click **Generate Keys** in the **Actions** pane
   (Be sure to apply the changes)

5. Copy the generated **Decryption Key** into NotePad or your favorite editor

6. For each SharePoint application using FBA, manually set the Decryption Key to the one you just generated using the process above