

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.



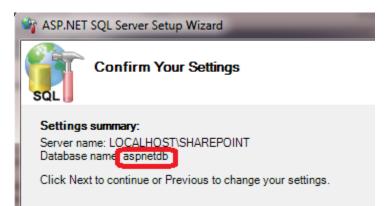
Step#2: Installing the FBA database

Why?

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

Details:

- 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.
- 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





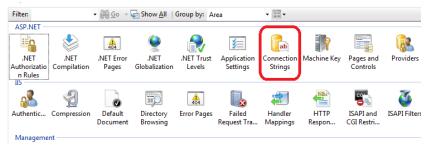
Step#3: Configure IIS to access the FBA database

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
- 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

Add Connection S	tring	?	X
<u>N</u> ame:	FBAMembershipStore		
SQL Server			
Server:	LOCALHOST\SHAREPOINT		
<u>D</u> atabase:	aspnetdb		
- Credentials			
Ose Wire	ndows Integrated Security		
Specify	credentials		
	Se <u>t</u>		
Custom			
Server=LOC	ALHOST\SHAREPOINT;Database=aspnetdb;Integrated Security=true	*	



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

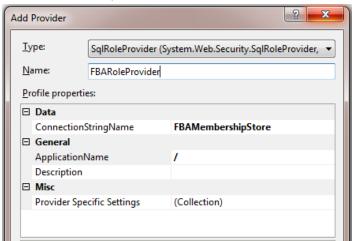


2. Select **.NET Roles** from the **feature** selector and right click in the screen. Click **Add** on the right click menu.

Providers			
Feature			
.NET Role	25 ▼		
Group by	No Grouping 👻		
Name	*		
c			
	Add		
	Connection Strings		
0	Help		
	Online Heln		



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 <u>http://weblogs.asp.net/scottqu/archive/2006/04/22/Always-set-the-</u> <u>2200 applicationName 2200 -property-when-configuring-ASP.NET-2.0-Membership-</u> <u>and-other-Providers.aspx</u>*



- 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

dit F	Provider	? <mark>-</mark> 2	×
Ţу	pe: \$qlMembershipPro	ovider (System.Web.Security.SqlMemb	per
Na	Name: FBAUserProvider		
<u>P</u> r	ofile properties:		
	EnablePasswordReset	True	*
	EnablePasswordRetrieval	False	
	RequiresQuestionAndAnswer	False	
	RequiresUniqueEmail	False	Ξ
	StorePasswordInSecureFormat	True	
	Data	l	_
	ConnectionStringName	FBAMembershipStore	
	General		
	ApplicationName	1	Ŧ

- 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

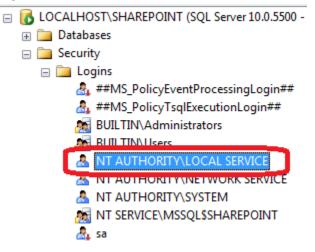
Advanced Settings	? ×
🗆 (General)	
Application Pool	SharePoint Web Services Root 🛛 📖
Bindings	http://sz843;,nttps://sz844;,net.tcp:3284
ID	2

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

Advanced Settings			
	General)		<u> </u>
	NET Framework Version	v2.0	
E	nable 32-Bit Applications	False	
	Managed Pipeline Mode	Integrated	
1	Name	SharePoint Web Services Root	Ξ
0	Queue Length	1000	
5	Start Automatically	False	
	CPU		
L	.imit	0	
L	imit Action	NoAction	
L	imit Interval (minutes)	5	
F	Processor Affinity Enabled	False	
F	Processor Affinity Mask	4294967295	
Process Model			
I	dentity	LocalService	
I	dle Time-out (minutes)	U	



- 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

Login Properties - NT AUTH	ORITY\LOCA	LSERVICE	
Select a page Providence of the second secon	Script	🕶 📑 Help	
Server Roles			
User Mapping Securables		ppe <u>d</u> to this login:	
Status	Map	Database	Us
		aspnetdb	N
		Bdc_Service_DB_2b8	
		master	
		model	
		msdb	
	1	SharePoint_AdminCon	N
	V	SharePoint_Config_b2	N
		tempdb	
		WSS_Content_c5ff8b	
		WSS Logaina 3156e	
Connection	Database	account enabled for: aspnet	db
Server: LOCALHOST\SHAREPOINT	✓ aspne	t_Membership_BasicAccess t_Membership_FullAccess t_Membership_ReportingAc	
Connection: ARMGASYS\dave.amga	aspne	t_Personalization_BasicAcc t_Personalization_FullAcces	s
View connection properties	aspne	t_Personalization_Reporting t_Profile_BasicAccess t_Profile_FullAccess	Acce
Progress	aspne	t_Profile_ReportingAccess	
Ready	🔽 aspne	t_Roles_BasicAccess t_Roles_FullAccess t_Roles_ReportingAccess	



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 \textcircled

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

Claims Authentication Types	Enable Windows Authentication
Choose the type of authentication you want to use for this zone.	Integrated Windows authentication
Negotiate (Kerberos) is the recommended security configuration to use with Windows authentication. If this option is selected and Kerberos is not configured, NTLM will be used. For Kerberos, the	NTLM Basic authentication (credentials are sent in clea
application pool account needs to be Network Service or an	Enable Forms Based Authentication (FBA)
account that has been	ASP.NET Membership provider name
configured by the domain administrator. NTLM	FBAUserProvider
authentication will work with	ASP.NET Role manager name
any application pool account and with the default domain	FBARoleProvider
configuration.	

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
- 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.



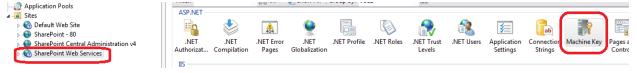
Encryption, FBA, and IIS Oh My!

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

Machine Key
Use this feature to specify hashing and encryption setti
Encryption method:
SHA1 -
Decryption method:
Auto
Validation key
Automatically generate at runtime
Generate a unique key for each application
AutoGenerate,IsolateApps
Decryption key
Automatically generate at runtime
Generate a unique key for each application

- 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