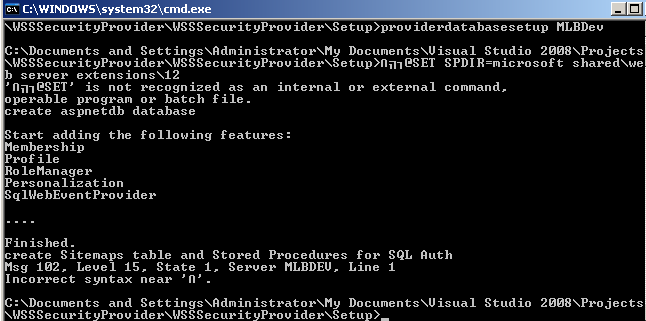
**Purpose:**

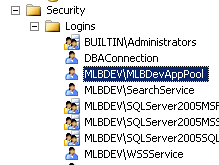
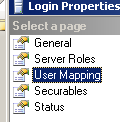
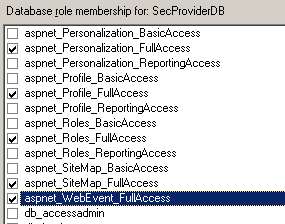
This creates a database called SecProviderDB. The database itself is nearly identical to the ASPNetProviderDB, but has an added table and some stored procedures for mapping the site’s fully qualified domain name (FQDN) to the provider’s application name. This will allow us to keep a list of users and roles that are unique to a FQDN, even though we may have multiple FQDN’s running in a single web application.

**Procedure:**

1. Open a command prompt and navigate to the setup folder.
2. Execute ProviderDatabaseSetup.bat and pass in the name of the database instance you want to create the database in. For example, I want to create it in my default instance, so I just enter: ProviderDatabaseSetup MLBDEV, because MLBDEV is my server’s name. If you have an instance named WSS, then you would enter ProviderDatabaseSetup <servername>\WSS (note in this case, you’ll also need to adjust the connection strings in the web.configs later on)



|  |  |
| --- | --- |
| Before | After |
|  |  |

1. Now you need to add your app pool accounts to this database. If you need to find these accounts, go to central admin operations tab and click Service Accounts (in the security administration group)
   1. Go to the Security> Logins and double click one of your app pool accounts:
   2. Select the User Mapping page: 
   3. Check the box next to our new database (SecProviderDB) and make sure that line is selected: 
   4. In the Role Membership section, check the boxes next to anything called aspnet\_xxx\_FullAccess and leave the public role checked as well: 
   5. Click OK
   6. Repeat this for the central admin app pool and any app pools for web applications where you want to use this Forms Based Authentication feature.

**Credit:**

This script is a modified version of one originally available in the SharePoint for Hosters kit: <http://www.codeplex.com/SharePointHosters>