ASPNETDB as Additional Profile Provider Configuration Guide

Introduction

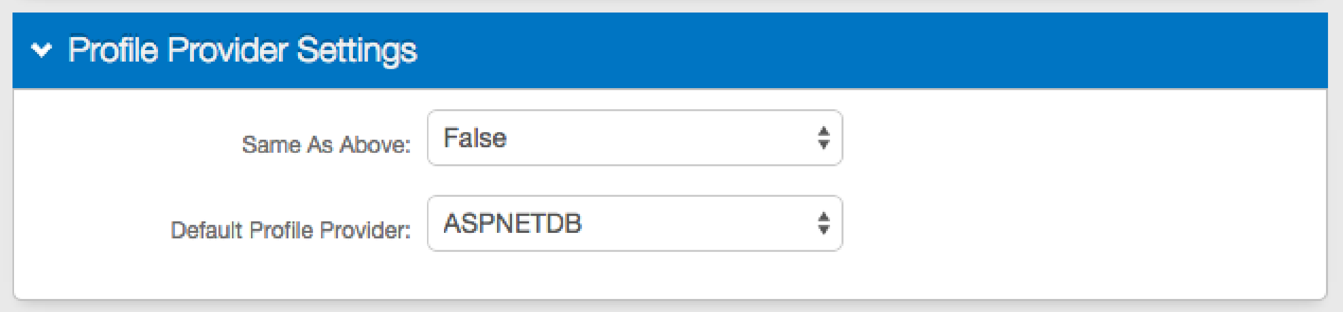
Use this guide along with the **Data Tab Configuration** guide to configure a SecureAuth IdP realm that uses ASPNETDB as an additional Profile Provider.

Prerequisites

1. Have an on-premises **ASPNETDB** data store (refer to **ASPNETDB Configuration Guide** for configuration steps to create the database)

2. Designate a service account with read access (and optional write access) for SecureAuth IdP

ASPNETDB Configuration Steps



Profile Provider Settings

Same As Above: False

Default Profile Provider: ASPNETDB

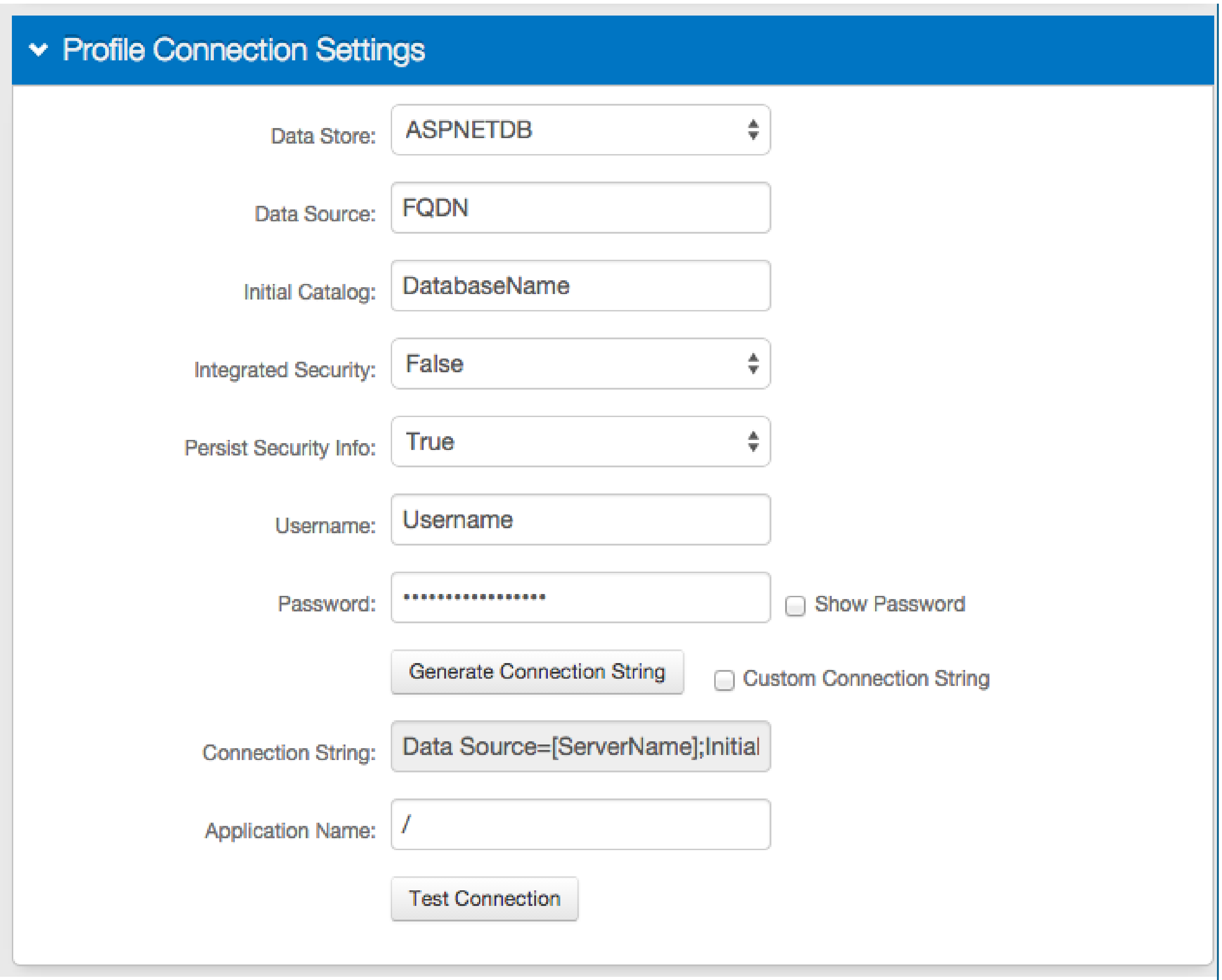
1. In the **Profile Provider Settings** section, select **True** from the **Same as Above** dropdown to copy the data store integration from the **Membership Connection Settings** section for use in profile connection; or select **False** if that directory is only used for the membership connection.

2. Select **ASPNETDB** from the **Default Profile Provider** dropdown if ASPNETDB is to be used as the default profile provider

ⓘ

- If another **ASPNETDB** data store is configured in the **Membership Connection Settings** section, and **True** is selected from the **Same as Above** dropdown, then those settings appear in the **Profile Connection Settings** (below) and must be modified to reflect the settings of the new ASPNETDB data store
- Only one **ASPNETDB** can be utilized for profile connection
- If another directory is selected from the **Default Profile Provider** dropdown, then **ASPNETDB** must be selected from **Source** dropdown in the **Profile Fields** section for the SecureAuth IdP **Properties** that are mapped to ASPNETDB fields

Profile Connection Settings



Profile Connection Settings

Data Store: ASPNETDB

Data Source: FQDN

Initial Catalog: DatabaseName

Integrated Security: False

Persist Security Info: True

Username: Username

Password: ••••••••••••••••• □ Show Password

□ Custom Connection String

Connection String: Data Source=[ServerName];Initial

Application Name: /

3. Select **ASPNETDB** from the **Data Store** dropdown

4. Provide the **Fully Qualified Domain Name (FQDN)** or the **IP Address** in the **Data Source** field

5. Provide the **Database Name** in the **Initial Catalog** field

6. Select **True** from the **Integrated Security** dropdown if the IIS app pool's service account is to be used in the connection (see **Integrated Auth Requirements** below)

Select **False** to specify an ASPNETDB account instead

Integrated Auth Requirements

1. Join the server to the domain to utilize a domain service account

2. In IIS, set the application pool **Identity** for both the **.NET v4.5** and **SecureAuth0** app pools to use the preferred service account; and set **Load User Profile** to **True**

3. Make the service account a member of the local administrators group of the SecureAuth IdP server(s)

4. Perform an **IIS reset** after making the changes

7. Select **True** from the **Persist Security Info** dropdown if access to the username and password information is allowed

8. Provide the **User ID** of the SecureAuth IdP Service Account (if **False** is selected in step 6)

9. Provide the **Password** associated to the **User ID** (if **False** is selected in step 6)

10. Click **Generate Connection String**, and the **Connection String** auto-populates

11. Provide the **Application Name** in which users that access this realm can be found, e.g. ***/***

12. Click **Test Connection** to ensure that the connection is successful

ⓘ Refer to **Data Tab Configuration** to complete the configuration steps in the **Data** tab of the Web Admin