

# SQL Server Configuration Guide

## Introduction

Use this guide along with the [Data Tab Configuration](#) guide to configure a SQL Server-integrated SecureAuth IdP realm.

NOTE: If connecting SecureAuth IdP to SQL Server User Data Store using Windows Authentication, click [here](#) to view the instructional PDF.

## Prerequisites

1. Have an on-premises **SQL Server** data store
2. Designate a service account with read access (and optional write access) for SecureAuth IdP

## SQL Server Configuration Steps

## Membership Connection Settings

Data Store:	SQL Server	▾
Data Source:	FQDN	
Initial Catalog:	DatabaseName	
Integrated Security:	False	▾
Persist Security Info:	True	▾
User ID:	Username	
Password:	.....	<input type="checkbox"/> Show Password
	<input type="button" value="Generate Connection String"/>	<input type="checkbox"/> Custom Connection String
Connection String:	Data Source=FQDN;Initial Catalog=DatabaseName;Persist Secu	
Password Format:	clear	▾
Allowed Groups:	Admins	
Denied Groups:		
Get User SP:	FIELD	
Reset Password SP:	FIELD	
Create User SP:	FIELD	
	<input type="button" value="Test Connection"/>	

1. In the **Membership Connection Settings**, select **SQL Server** from the **Data Store** dropdown
2. Provide the **Fully Qualified Domain Name (FQDN)** or the **IP Address** in the **Data Source** field
3. Provide the **Database Name** in the **Initial Catalog** field
4. 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 a SQL service 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

5. Select **True** from the **Persist Security Info** dropdown if access to the username and password information is allowed
6. Provide the **User ID** of the SecureAuth IdP Service Account (if **False** is selected in step 4)
7. Provide the **Password** associated to the **User ID** (if **False** is selected in step 4)
8. Click **Generate Connection String**, and the **Connection String** auto-populates
9. Select how the Service Account **Password** is to be stored in the directory from the **Password Format** dropdown
10. Create a list of **Allowed Groups** that can access the target resource of this realm, e.g. **Admins**
11. Create a list of **Denied Groups** that cannot access the target resource of this realm
12. Provide the **Stored Procedure Name** for **Get User SP**
13. Provide the **Stored Procedure Name** for **Reset Password SP**
14. Provide the **Stored Procedure Name** for **Create User SP**
15. Click **Test Connection** to ensure that the connection is successful

If using a **Custom Connection String** and experience an error when testing the connection, then refer to the **Custom Connection String Error** section below for a workaround



Refer to [Data Tab Configuration](#) to complete the configuration steps in the **Data** tab of the Web Admin



Refer to [SQL User Data Store Tables and Stored Procedures Configuration Guide](#) for information regarding profile mapping

## Custom Connection String Error

Membership Connection Settings

Data Store:

Data Source:

Initial Catalog:

Integrated Security:

Persist Security Info:

User ID:

Password:   Show Password

Custom Connection String

Connection String:

Password Format:

Allowed Groups:

Denied Groups:

Get User SP:

Reset Password SP:

Create User SP:

If manually entering a custom connection string, an error may occur when testing the connection, which hinders the SQL Server to successfully integrate with SecureAuth IdP

This error may occur only if **Custom Connection String** is checked, the **Connection String** is manually entered into the field rather than generated by the Web Admin, and the fields that comprise the generated **Connection String** are left empty / default

## Workaround

### System Info

#### Links

Web Config Backups: [Click to view Web Config Backups.](#)

Web Config Editor: [Click to edit Web Config file.](#)

1. In the **Links** section, select **Click to edit Web Config File**

### Web Config Editor

#### Web Config Editor

```
<add name="SQLServer" connectionString="Data Source=[ServerName];Initial Catalog=[DatabaseName];User ID=[SQLUserName];Password=[SQLUserPassword]"
providerName="SqlProvider" />
```

2. Search for **SQLServer** and manually enter the connection string into the web.config file
3. Click **Save**

This enables a successful connection; however, clicking **Test Connection** in the **Data** tab may still yield an error