Data Tier.Net Quick Start

Setting up Data Tier.Net

**Update 3.2.2025 Major Milestone - New Setup Screen –
This project now targets .NET 9.**

**Update 1.4.2024: Major Milestone – Version 2 Templates with 2
Projects**

Update 8.6.2023: **Major Milestone**

DataTier.Net now targets .NET 5 - 9

The biggest difference usage difference with the .NET Core versions
(NET 5, .NET 6, .NET 7,.NET 8 and .NET 9 versions) is you must
create a system environment variable to hold the connection string.

DataGateway

Create a new instance of a Gateway object, pass in the environment
variable name.

.NET Core Versions .NET 5, NET 6, .NET 7, .NET 8 and .NET 9.

```
// create a new gateway object and pass in the connectionName
Gateway gateway = new Gateway(ConnectionConstants.Name);

// .NET Framework, the connection string is read from the app or
web.config
Gateway gateway = new Gateway();
```

Installing Project Templates

The Setup Control (shown on the next page), has options for .NET
Framework and .NET 5 - .NET 9.

There are two steps to setting up DataTier.Net:

Step 1. Create DataTier.Net.Database in SSMS

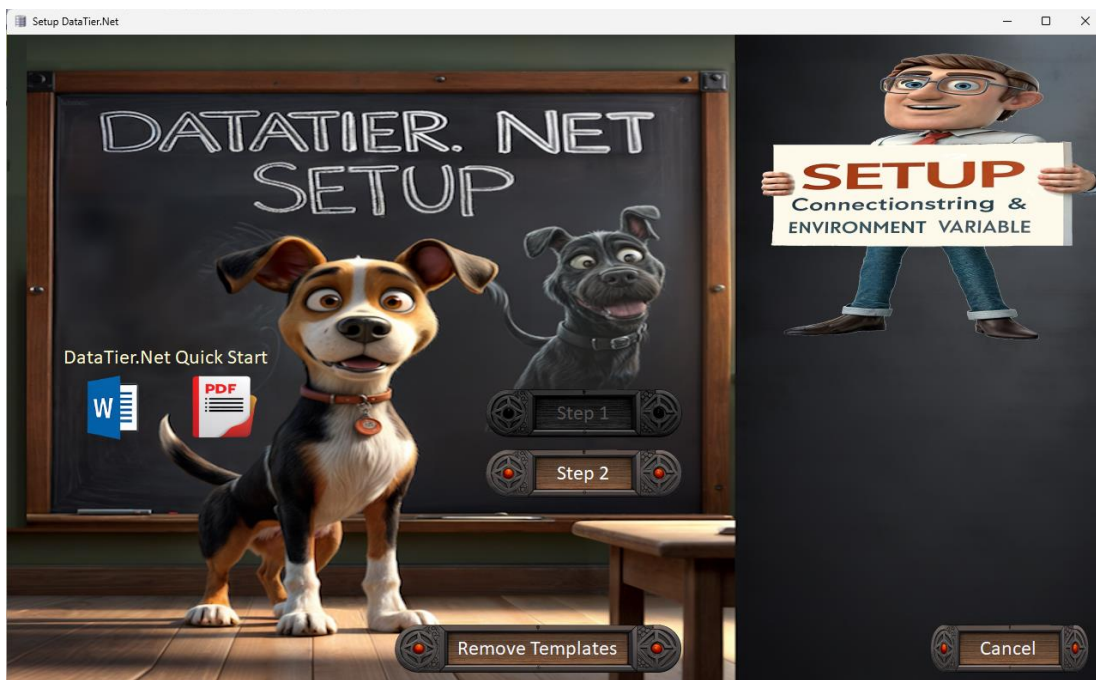Create a new database named 'DataTier.Net.Database' in SQL Server Management Studio.

Check the box 'Database has been created'.

This will enable the link to the DataTier.Net.Database DataTier.Net.Database.Schema.sql as shown on the next page

Execute this SQL script to create the tables and stored procedures that DataTier.NET uses internally. This will enable the Step 2 button.

Step 2: Build Connectionstring & Set Environment Variables

8.6.2023: New Connection String Builder

Note: Microsoft.Data.SqlClient 5.0+ requires you to specify Encrypt=true in your connectionstring, or it will not work.

Using Connection String Builder

Windows Authentication (recommended)

If you are using Integrated Security, ensure your Windows Login has permission to execute stored procedures and modify data in your SQL Server.

SQL Server Authentication

SA login: If you are using the sa (system administer) account on SQL Server then you can skip the next section.

SQL Server Login:

Create a new SQL Server login named 'DataTierNetUser'. Include the user in the following two roles: DataReader, DataWriter.

After you fill in the authentication options, the Build Conn String button will become enabled:
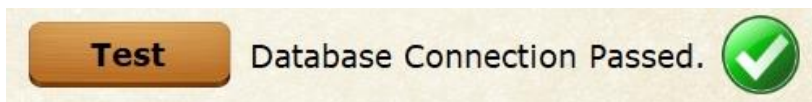

Once you build your connection string, the other buttons will become enabled as shown on the picture at the bottom of the previous page.

Test Database Connection

You should test your database connection before you click the 'Install Connectionstring & Update App.config' button.

Provided your Connectionstring works, you should see this message:

Five seconds after you see the installed message, the Connection String Builder Form closes, and you are shown a message that a restart is required:
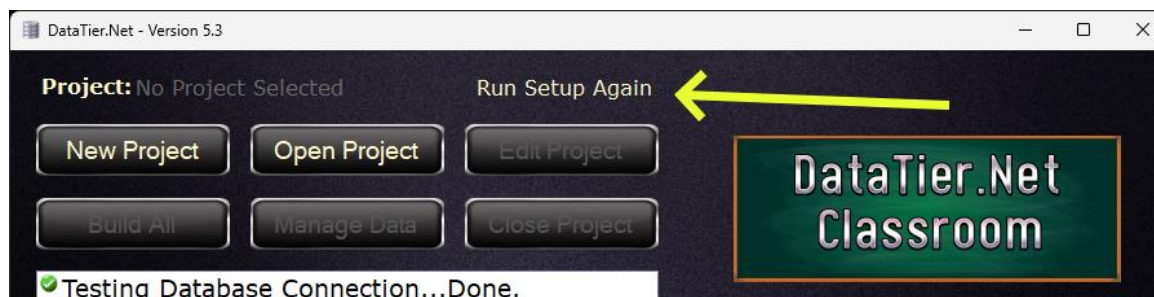
Restart Required

After setting up your connectionstring, DataTier.NET must restart. The .Net Framework configuration files cannot be reloaded after the app starts. Now that .NET Core is the primary focus on may change this feature in the future if I have time.

DataTier.NET now uses Environment Variables so a restart is no longer required.

Run Setup Again

To run the Setup Control again, click the Run Setup Again button:

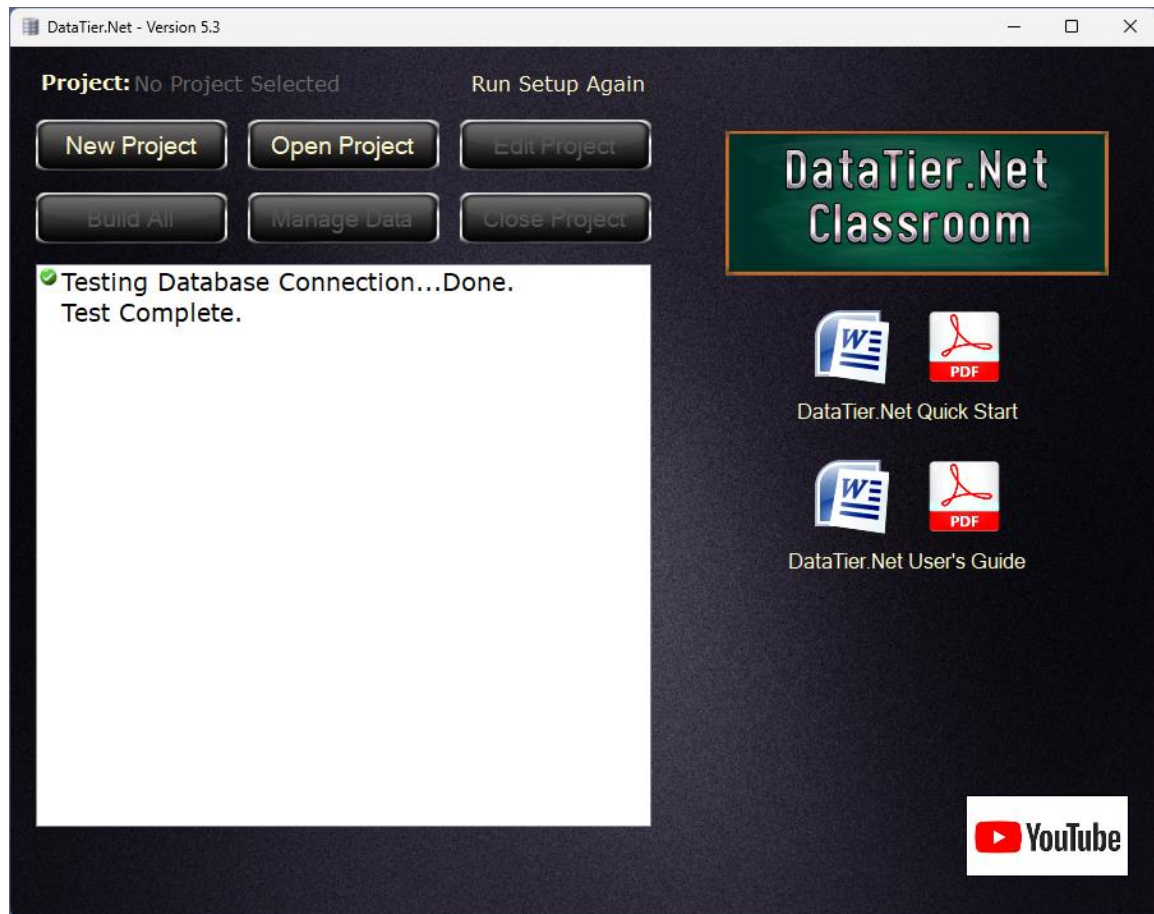## Development Version vs Install Version

You must build DataTier.NET in Visual Studio 2019. The project references Visual Studio itself and VS 2022 is not compatible with DataTier.NET.

Development Version Only

Most of this document is the same for the release installed version and the source code version. If you are running the source code version, you may need to set the DataTier.Net.Client project as the start-up project in Solution Explorer in Visual Studio.

Test Your Setup

Run DataTier.NET and you should see:

Troubleshooting Connection Issues

If you do not see the above message, the most likely cause is either the Data Tier.Net SQL Server database is not installed or the environment variable for DataTierNetConnection is not setup with a valid connectionstring.

Solving Security Problems

It is outside of the scope of this document to solve SQL permission problems, but one useful stored procedure included in the DataTier.Net.Database could help.
The script is installed in the DataTier.Net.Database.

In SQL Server Management Studio, create a new query window and ensure the correct database is selected before running the following query
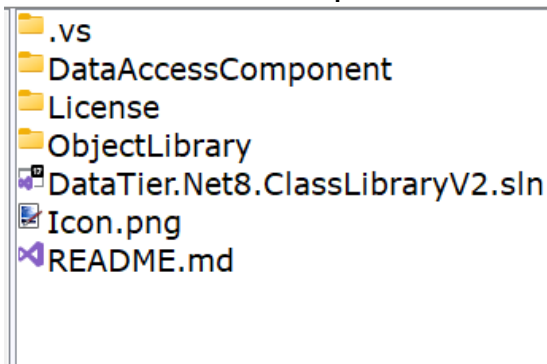
UpdateProcPermissions 'DataTierNetUser'

I regret not saving the author of this procedure to give them credit, but this has saved me on numerous occasions.
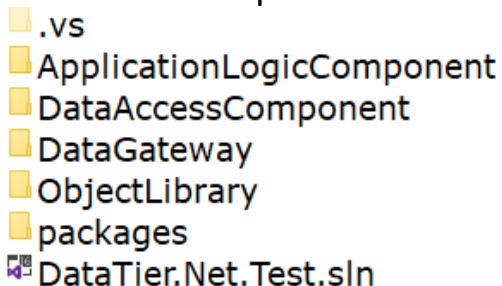
More Info About DataTier.Net Project Templates

The project templates are used to help you create your own DataTier.Net projects by creating a Class Library which includes all the required references and projects required to build a DataTier.Net solution. The image below shows the directory structure of a Data Tier.Net Project Template:
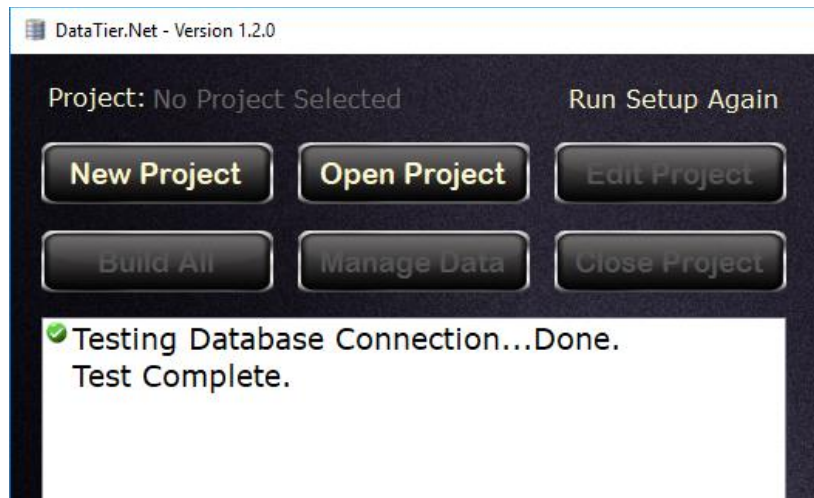
The Version 2 Templates use only two projects:
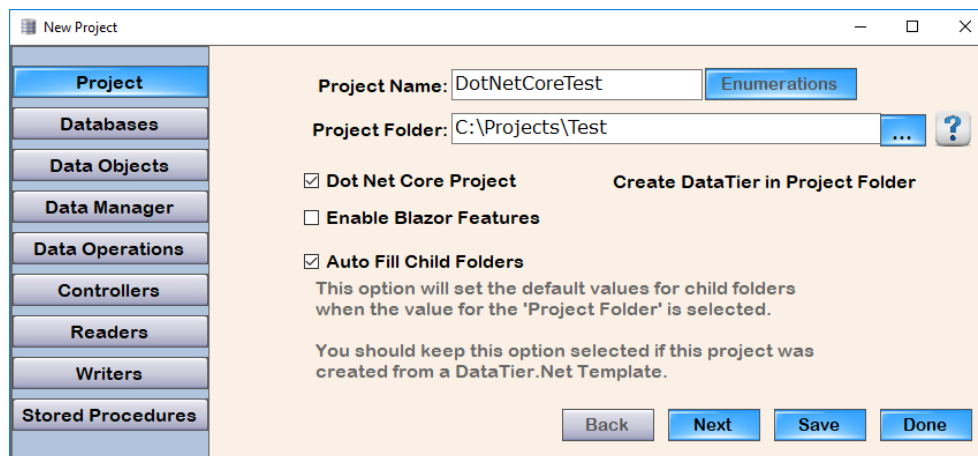

Version 1 Templates Use 4 Projects

Create a DataTier.Net Project

Click the New Project button. This will launch the Project Wizard Control.

New Project Wizard

Type in a Name for your project and browse for or enter a Project Folder.

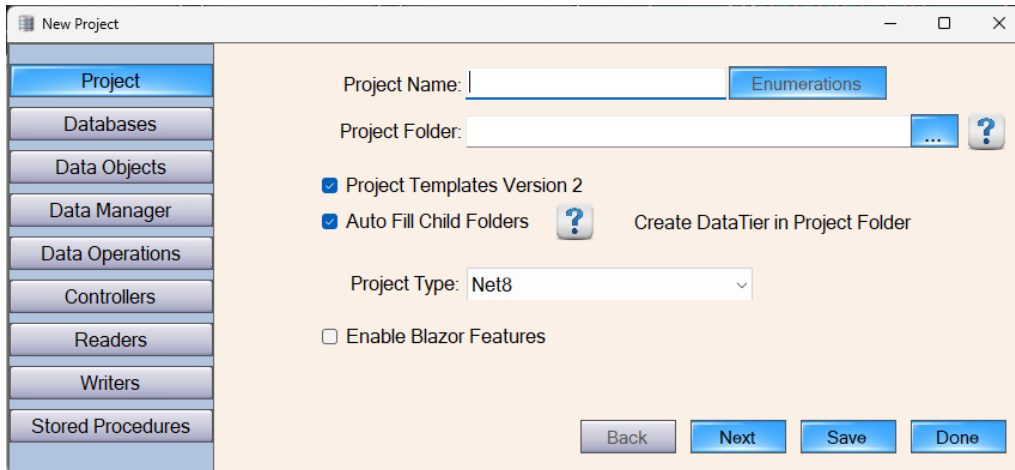The project folder is the folder above the four projects in your data library.

Tip: An easy way to find your project folder is to right click your solution in Visual Studio and select 'Open Folder in File Explorer'.

Creating .NET Core Projects .NET 5 - .NET 9

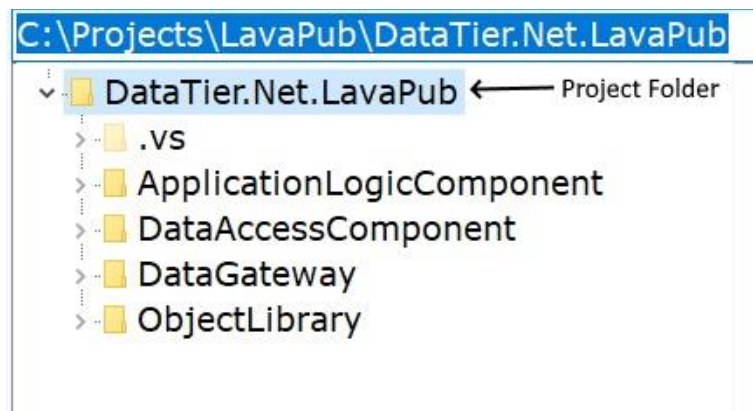.Net Core projects are different as you start by creating a new project.

Click the New Project button and set the name and project folder.

New projects now default to .NET 9. Change the Project Type for older versions.
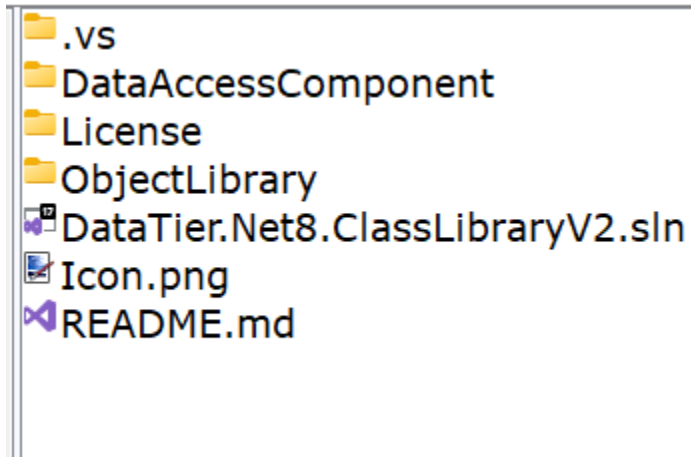

Ignore the .Net Framework instructions below when they tell you to create a new project. Other than that, almost everything is the same with .Net Framework and .Net Core.

Project Structure of a DataTier.Net Project

The project folder is the folder above the four projects in your solution.

Version 2 Template Structure only has two projects: Data Access Component and Object Library

📁 .vs
📁 DataAccessComponent
📁 License
📁 ObjectLibrary
DataTier.Net8.ClassLibraryV2.sln
Icon.png
README.md

Auto Fill Child Folders

☑ **Auto Fill Child Folders**

This option will set the default values for child folders when the value for the 'Project Folder' is selected.

You should keep this option selected if this project was created from a DataTier.Net Template.

By default, this value is selected and will populate the other tabs of the project wizard for you. Leave this option selected, else use at your own risk.

Add a database to your project.

Click the 'Next Button' to select the database tab and click the Add button:

This will launch the Database Editor

Note: this connection is only used by Data Tier.Net to retrieve the database structure. The connection string for your own DataTier.Net projects is configured in the app.config or web.config files in your solutions.
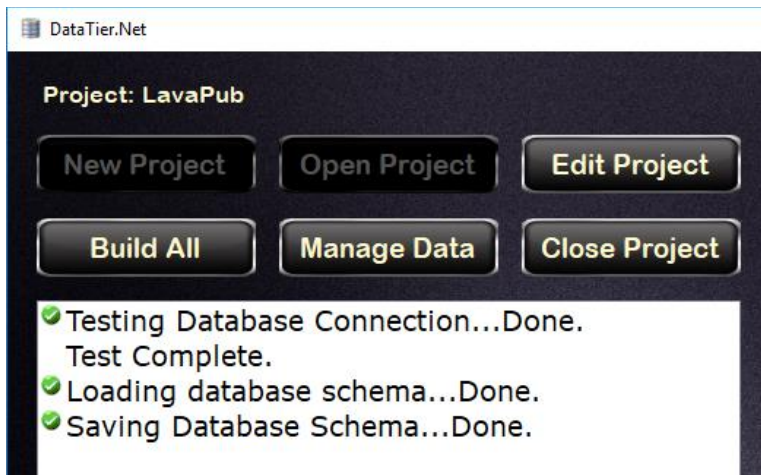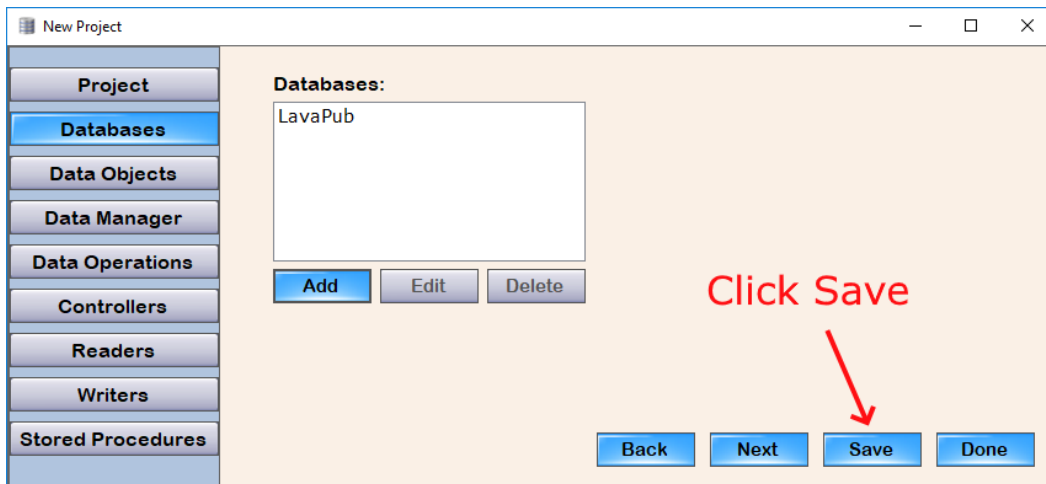
Tip: It is recommended you type in your server's name. If you click the Server button to browse for Servers, you may need to add \SQLExpress to the server's name detected. This is a known issue.

Clicking the ellipses button (…) will refresh the combo box with the list of databases.

Provided the connection to the database can be established, when you click the Browse Database Button, the list of available databases will be populated.

Select the desired database then click the 'Save' button to save your database.

Click the 'Save' button again to exit the Project Wizard:

Update: 7.13.2019: Screenshot updated for DataTier.Net.

When you save your project, the database schema is read and then saved.

Manage Data

If you wish to exclude any tables, views or fields from your project, click the Manage Data button before you build.

If you exclude any tables or fields, the Save button will become enabled.

Note: Views are listed as tables, although their behavior is different.

Click 'Save' if you make any changes. Click Done to exit this form.

You are now ready to build your project with DataTier.Net!

Build Your Project

Clicking the Build All Button will code generate all the required objects and stored procedures to assemble your data tier.

Include the generated files in your project

New Files that are generated must be included into your Visual Studio solution.
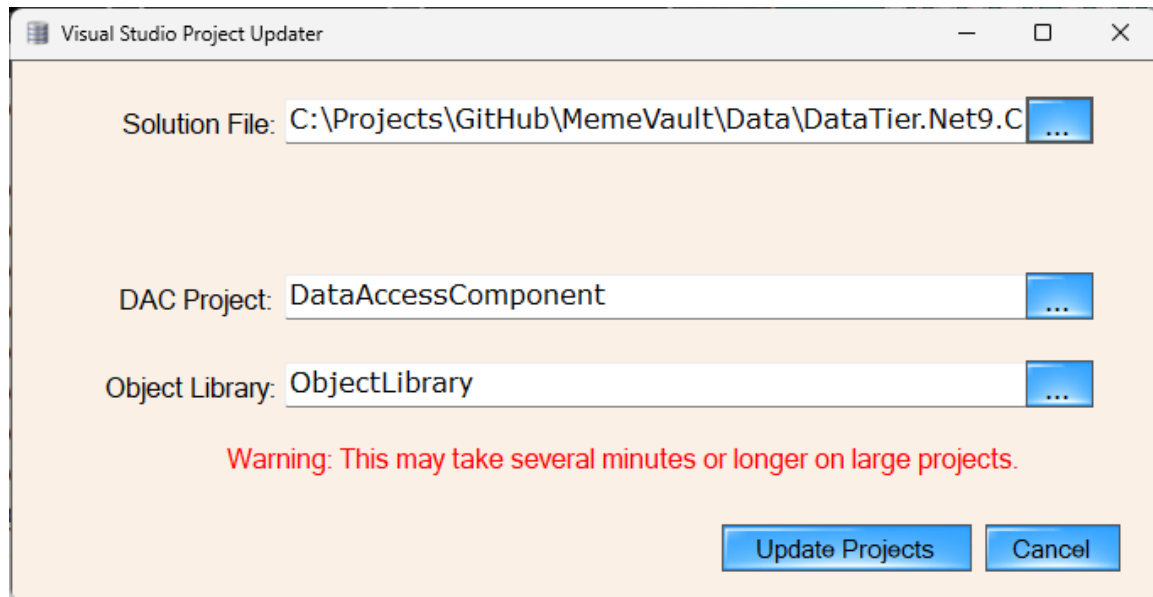
When you rebuild your project with DataTier.Net, you will only have to include new files if you add a new table or view.

If DataTier.Net determines that new files were added, the Visual Studio Project Updater will launch after you build.

Using the Visual Studio Project Updater Control

Step 1: Select the DataTier.Net.ClassLibary VS solution file (.sln).

Data Tier.Net will read the project names from the solution file.

If you created your project from a DataTier.Net.ClassLibrary project template the names will be correct. You can rename your projects after creating from a Data Tier.Net project template.

If the project names are not detected, click the browse button for each project and a browse dialog for each project file will launch.

After your projects are selected, click the Include Project Files button. This will include any files in your project that were generated during the last build.

5.1.2019: The Include Project Files Button Is now clicked on your behalf.

Often, I would forget to click the Include Project Files button, and I would have to manually include all the required project files, and it is quite a tedious job.

You should not have to manually include project files now, but for anyone that ever has this need I will leave the instructions in.

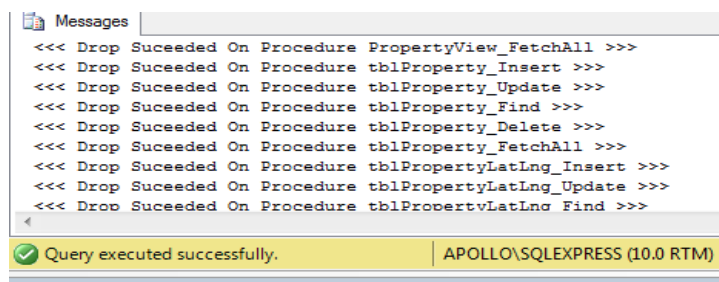**Executing the StoredProcedures.sql**

Update 4.2.2019: Click the link 'Stored Procedures.sql' in DataTier.Net

The file is kind of buried, so I added this link to make it exponentially easier to find. Provided you have SQL Server Management Studio installed, Data Tier.Net will launch SSMS and open the stored procedures.sql file that was just created.

Location of stored procedures.sql in the Data Access Component project:

StoredProcedureManager\StoredProceduresSQL\storedprocedures.sql

After executing the stored procedures, you should see a message like:

Update 12.19.2012: Gateway Generator

The gateway is now code generated when you build.

Load Method

For each table or view in your database, the gateway will create a Load method.

Update 4.5.2019: The above documentation does not state that now each 'Active' table will create a Load method. Tables can now be excluded, but for the remainder of this section, just keep this point in mind.

Delete, Find and Save Methods

For tables that have an Identity Insert (auto-number) primary key, the Delete, Find and Save methods will be code generated.

Update 4.5.2019: The main engine of DataTier.Net, DataJuggler.Net.SQLDatabaseConnector, is now more robust than it was when this 'must be identity insert' rule was implemented. I just haven't had time to implement this.

Once I publish, volunteers are invited to help make improvements.

Save Method

The Save method analyzes the IsNew property of the object being saved, and calls Insert or Update accordingly.

Read the DataTier.Net User's Guide for more information.

Tutorial Videos

Please check out my YouTube channel, as I will update videos and sample projects as often as I am able. My pesky day job gets in the way of creativity but allows me to eat to continue being creative; thus, a paradox.
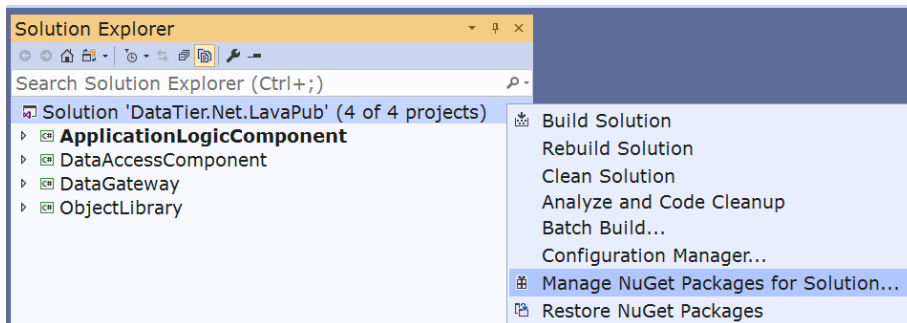
https://www.youtube.com/DataJuggler

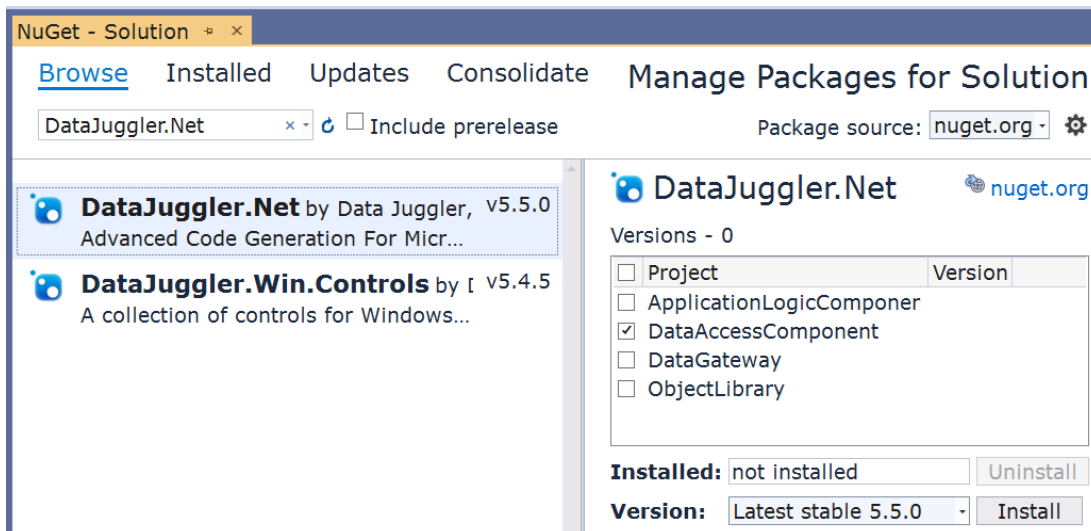**Legacy info for .NET Framework. Not Needed for .NET Core.**

Install DataJuggler.Net NuGet Package

.Net Framework Only. Dot Net Core packages are already installed.

Right click the solution and select 'Manage NuGet Packages for Solution' as shown above.

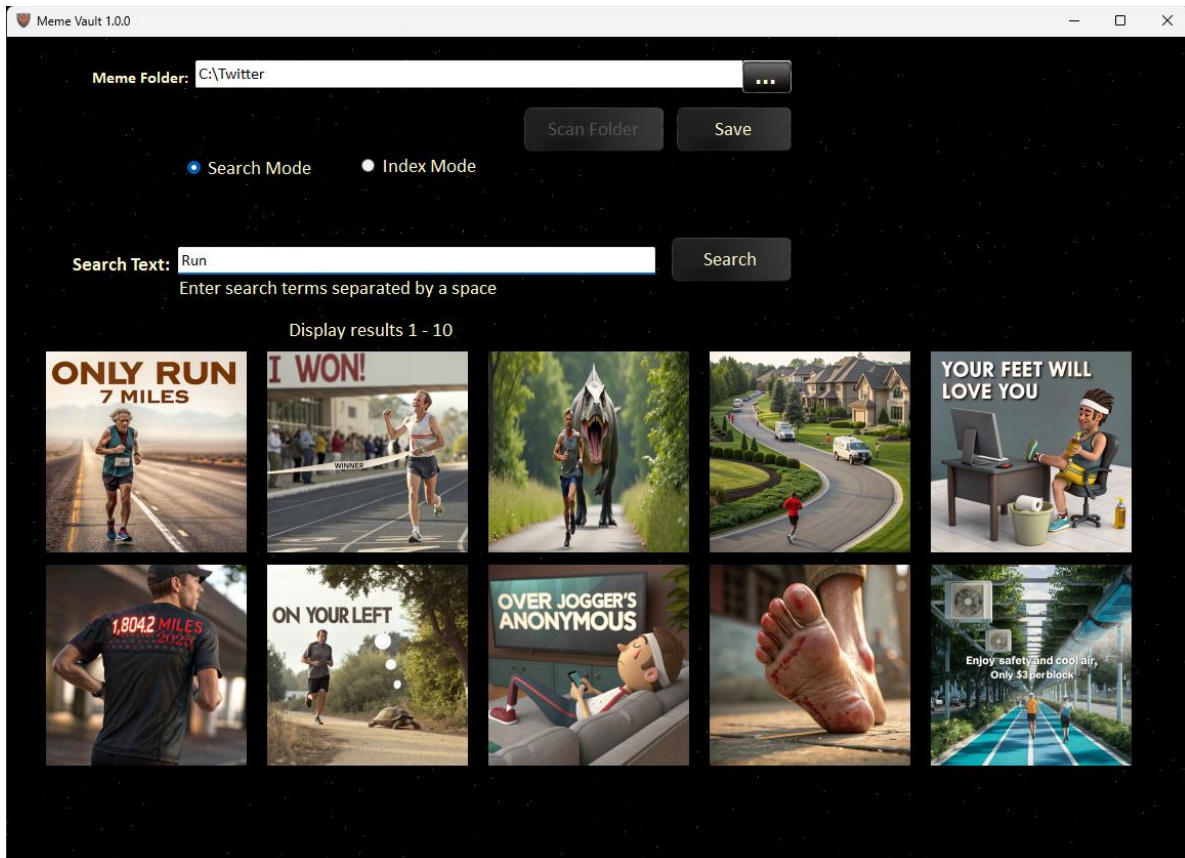In the search box, type in DataJuggler.Net.

Select DataJuggler.Net and check the DataAccessComponent project.

Click Install and accept any prompts with 'OK'.

Build your solution in Visual Studio.

Sample Projects built using DataTier.Net.

Meme Vault https://github.com/DataJuggler/MemeVault

Meme Vault is used to create aliases for images. New video is coming soon.

DB Compare https://github.com/DataJuggler/DBCompare

DB Compare compares the schema from two SQL Server databases and reports any schema differences. The core component of DB Compare is the file DataJuggler.Net.SQLDatabaseConnector; the same object used by Data Tier.Net to read database schema.

XML Mirror https://github.com/DataJuggler/XmlMirror

I created XML Mirror during one of my previous employments, which involved parsing large amounts of XML data.

XML Mirror uses the class CSharpClassWriter to create the following files:

Parser.base.cs   Parser.custom.cs

DataJuggler.Net.CSharpClassWriter is the same file used in Datatier.Net to code generate files.

After Xml Mirror was created, I updated DB Compare to perform remote comparisons of a SQL Server database located on a virtual machine, against a database you can connect to via a connection string, or vice versa.

I hope you enjoy DataTier.Net