



# How-to Create a Reusable Framework for your Database

## *Step-by-Step*

Published: 7 November 2016

Copyright © Genesys Source. All rights reserved. Printed in USA.

Microsoft, OneDrive, Windows, Xbox, and Xbox Live are registered trademarks of Microsoft Corporation.

All other product names and logos are trademarks and service marks of their respective companies.

This document is provided "as-is." Information in this document, including URL and other Internet website references, may change without notice. Genesys Source assumes no liability for damages incurred directly or indirectly from errors, omissions, or discrepancies between the product and this document.

An attempt has been made to state all allowable values where applicable throughout this document. Any values or parameters used beyond those stated may have unpredictable results.

# Contents

Why Reusability? .....	4
What do I need? .....	4
What tech is in the Genesys Framework? .....	4
Installing the Genesys Framework .....	4
Download the sample app by:.....	5
From GetFramework.com, click Genesys Framework Quick-Start for Mvc, or WebApi, complete the free check-out, and click the Download link on your confirmation page or email.....	5
Install the Visual Studio template by:.....	5
Create a new MVC or Web API Genesys Framework Project from Visual Studio .....	5
How-to Pull your Data through your new Reusable Framework .....	6
The Genesys Framework Quick-Start Solution .....	6
Genesys Framework for MVC .....	7
Genesys Framework for Web API.....	7
<b>Step 1:</b> Publish the FrameworkData database to your SQL Server .....	7
<b>Step 2:</b> Wire the CustomerInfo view to your existing table .....	8
<b>Step 3:</b> Run the WebApp or WebServices project to Enjoy Your New Resuable Framework! .....	9

## Why Reusability?

Your code and your database have been running well and doing the job for years. But you notice that programmers are becoming kings of the copy/paste anti-pattern. Your "Customer" data access code and screens have been copied 10 times from the original code to each page that displays Customer information.

And now you want to add a *CustomerSince* field, but you do not want to change the customer class in all 10 projects.

You think, *"Why didn't I create a central customer class that all projects can use?"*

Well, you are not the only one. Reusability is great, once you have invested the time. *But who wants to invest the person-hours and risk missed deadlines, to retrofit your Customer screens for reusability?*

Genesys Framework's was created to solve this problem:

*Code reusability quickly and with a low learning curve. Allowing you to incrementally create your reusable framework one page at a time.*

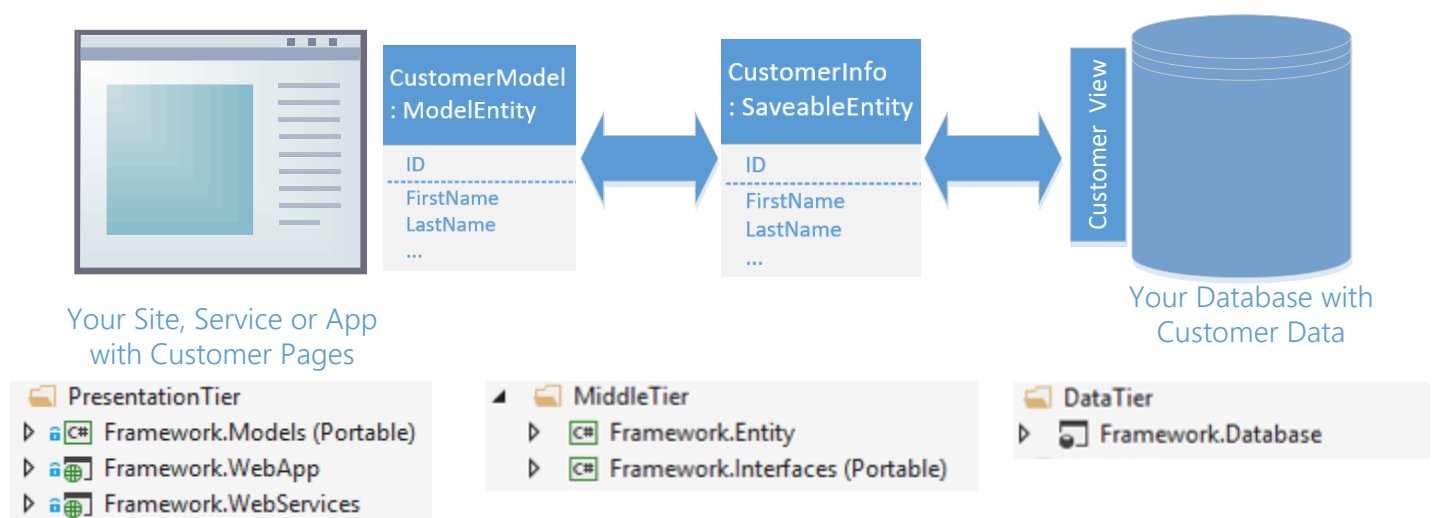


Figure 1: Reusable Customer entity for your app and database

<i>What do I need?</i>	<i>What tech is in the Genesys Framework?</i>
Visual Studio (2015 preferred)	.Net 4.6, MVC 5, WebAPI 2, Entity Framework 6,
SQL Server (2016 preferred)	Universal Windows Platform, C# 6, Html 5, CSS
Windows (10 preferred)	3, Bootstrap 3, JQuery 3

## Installing the Genesys Framework

The Genesys Framework is available as an on-line sample app download, and in Visual Studio as a solution template.

Download the sample app by:

From [GetFramework.com](http://GetFramework.com), click Genesys Framework Quick-Start for Mvc, or WebApi, complete the free check-out, and click the Download link on your confirmation page or email.

Now unzip and open the Genesys Framework solution file and you are ready to *Error! Reference source not found.* as described in the section below.

Install the Visual Studio template by:

From Visual Studio, click on *Tools->Extensions and Updates*, and select *Online->Templates->Visual C#:*

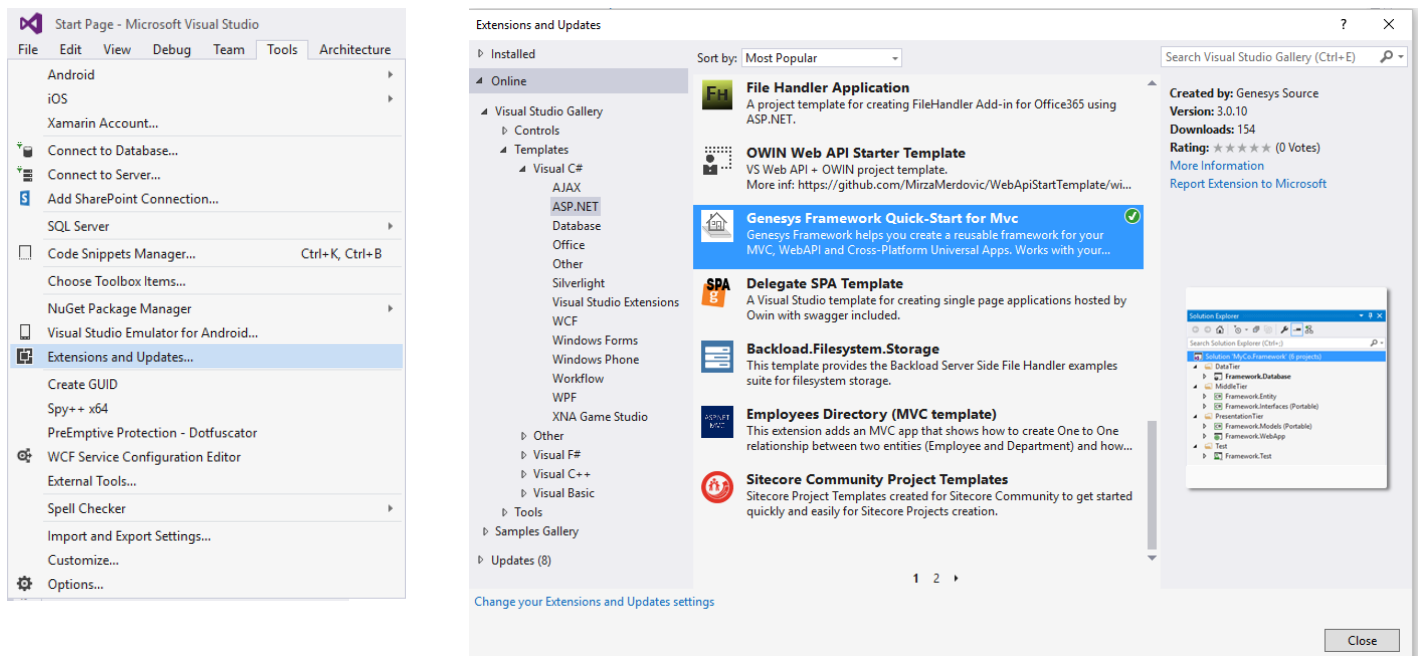


Figure 2: Install Genesys Framework from Visual Studio: Tools -> Extensions and Updates

### *Create a new MVC or Web API Genesys Framework Project from Visual Studio*

After you have installed the Genesys Framework Quick-Start MVC or Web API template, you can create a new MVC or Web API project that is ready for reusability.

From Visual Studio...

1. Click File -> New Project
2. Select Templates -> Visual C#
3. Search for (or press G in the list) to find the Genesys Framework templates

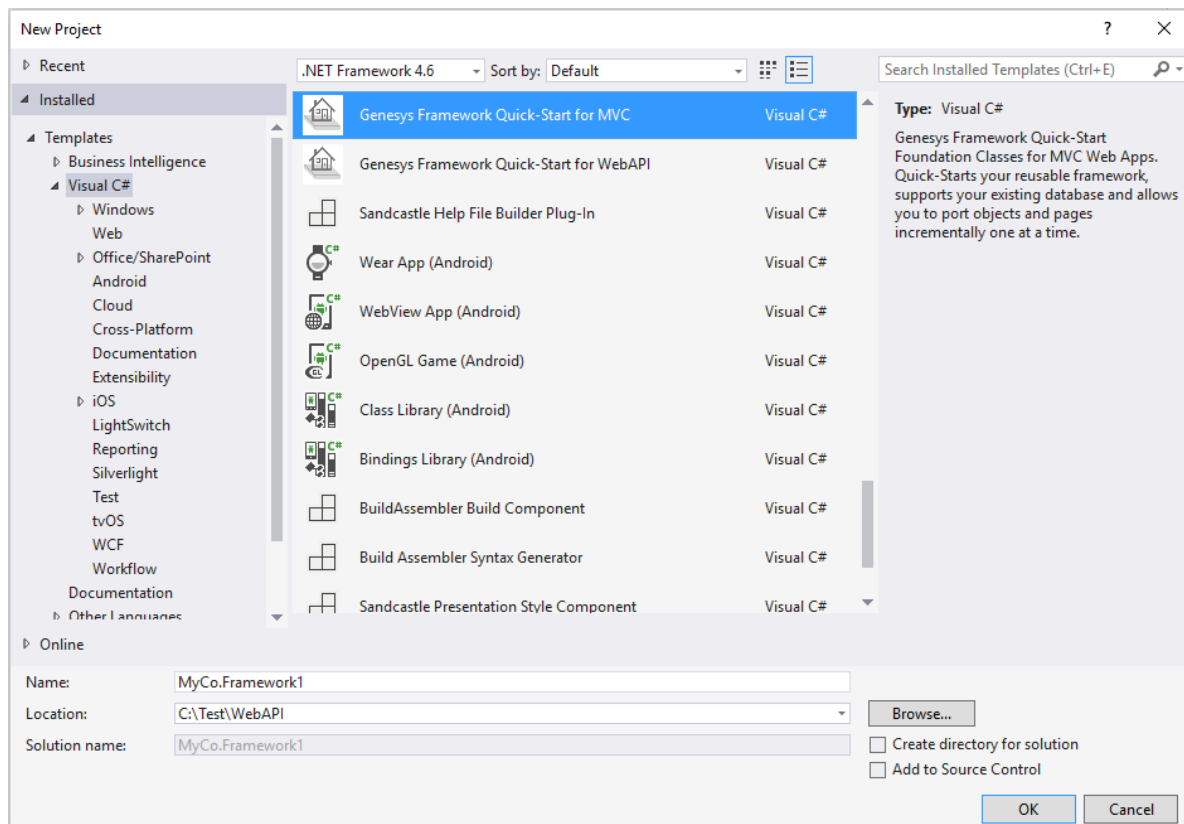


Figure 3: Visual Studio File -> New Project -> Visual C# -> Genesys Framework

## How-to Pull your Data through your new Reusable Framework

Your database and code is the goldmine of your organization. Let's add a reusable framework directly to your existing database without altering any of your existing tables or existing code base.

And by the way...let's get your reusable CustomerInfo class up and running in less than 20 minutes. If you know SQL Server Data Tools (SSDT), you can take a coffee break during this exercise and still get done early.

**Important:** The following procedure is to be ran on test systems, against test data, in a test environment. Never run a sample application against any system used for production or important tasks of any kind.

### *The Genesys Framework Quick-Start Solution*

By now you have downloaded the sample app or installed the Visual Studio template, and opening your solution will look like this:

## Genesys Framework for MVC

## Genesys Framework for Web API

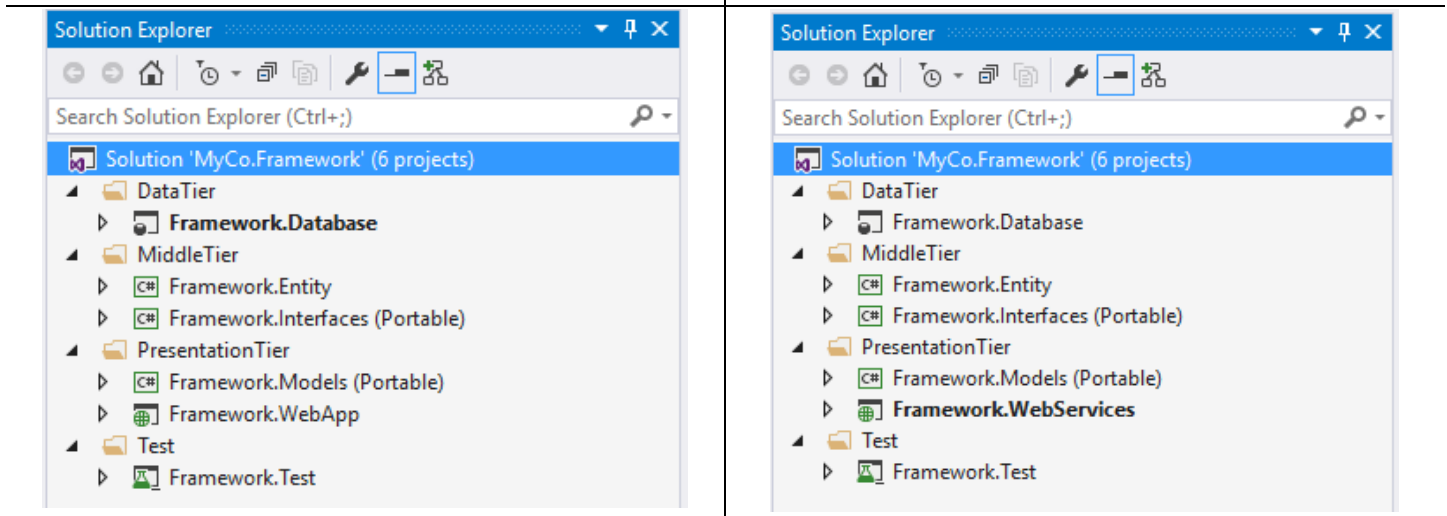


Figure 4: Genesys Framework Quick-Start for MVC and Web API

The solution is organized in the traditional 3-tiers of an app, and a test project:

Data Tier	Middle Tier	Presentation Tier	Test
<b>Framework.Database:</b> SQL Server Data Tools (SSDT) project that contains all SQL objects for your framework.	<b>Framework.Entity:</b> Entity Framework reusable data access objects for all reading/writing to your DB. <b>Framework.Interfaces:</b> Cross-platform Interfaces that enforce consistency in your Entities and Models.	<b>Framework.WebApp:</b> MVC web app demonstrating create, edit, delete, summary and search of a Customer object. <b>Framework.WebService:</b> Web API service demonstrating create, edit, delete, summary and search of a Customer object. <b>Framework.Models:</b> Lightweight cross-platform Models for your MVC views and Web API JSON objects.	<b>Framework.Test:</b> Unit-test project exercising create, edit, delete, summary and search of a Customer object.

Table 1: Genesys Framework Projects by Tier

### Step 1: Publish the FrameworkData database to your SQL Server

The first step is to get your database and the Genesys Framework talking. This is done in the *Framework.Database* project, which is built upon SQL Server Data Tools (SSDT.)

***Critical Warning!*** Do not follow this procedure in production! This procedure is for demonstration purposes only. Never point this, or any other sample, to a live database, live server or any live system of any kind!

1. Open your newly created framework solution.

2. Open `DataTier\Framework.Database\Publish\PublishToDev.publish.xml` file
  - a. Click Edit and change the connection to your existing test database.
  - b. Change the *Database Name* field to your test database name
  - c. Change the *DatabaseData* field to your test database name.
3. Click the *Generate Script* button to see a simulation of what will be published.
  - a. Review carefully to ensure no unwanted changes will occur.
4. Are you ready to push the changes?
  - a. If **No**, stop and do not proceed.
  - b. If **Yes**, click the Publish button and all changes will be pushed to your test database.

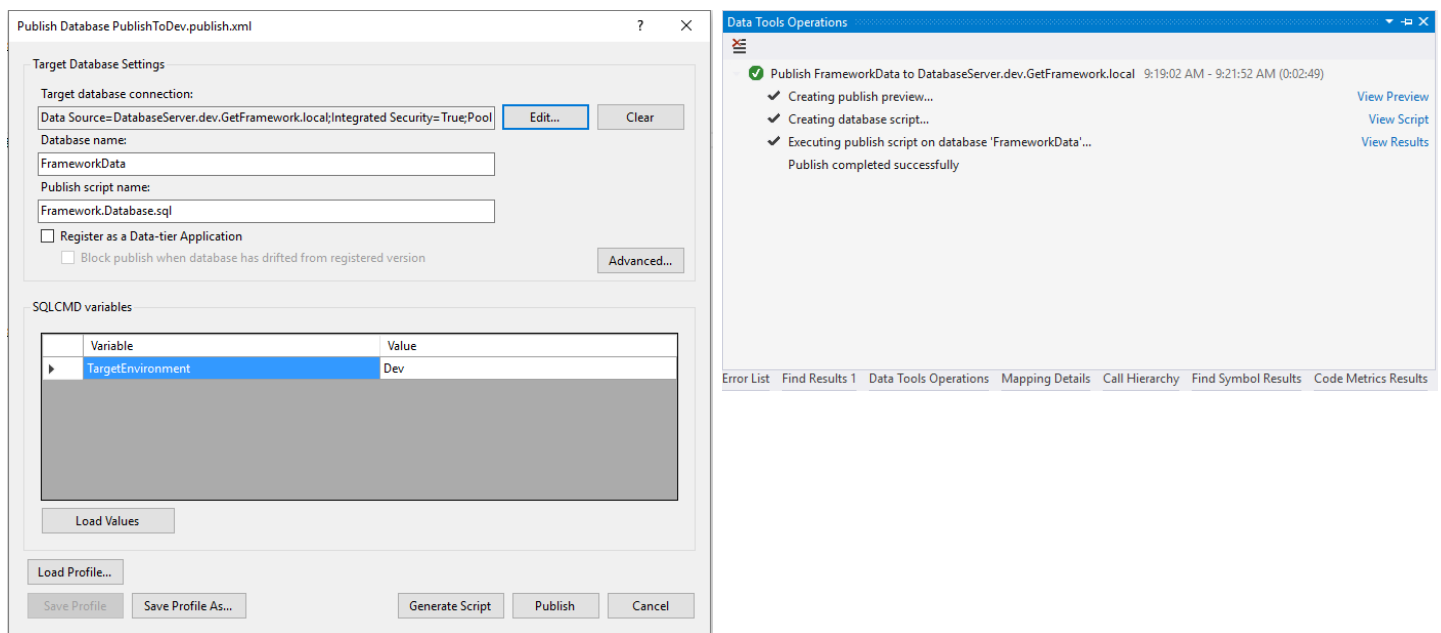


Figure 5: Publishing the Framework.Database SSDT Project to your SQL Server

The contents of the Framework.Database project have now been pushed to your test database. You can now change the connection string and run the Framework.WebApp project as-is.

---

*Remember to change the connection string to your actual test database, found in:*  
**Framework.WebApp\Local\ConnectionStrings.config.**

---

## Step 2: Wire the CustomerInfo view to your existing table

In Step 1, you pushed the Genesys Framework out-of-the-box "Customer" sample tables, views and stored procedures to your test database. If you haven't already, you can run the Genesys Framework as-is.



---

*Don't forget to change the connection string to your actual test database, found in:*  
**Framework.WebApp\Local\ConnectionStrings.config -Or -**  
**Framework.WebServices\Local\ConnectionStrings.config**

---

But in the real world, you need your data from your database populating your reusable framework objects. In this case, let's take the CustomerInfo sample object and pull your existing data into this pre-built object.

To pull your data into the Genesys Framework CustomerInfo object:

1. Find one of your tables that one or more of the following fields: *First Name, Middle Name, Last Name, Date of Birth or Gender*.
2. Open **DataTier\Framework.Database\EntityCode\Views\CustomerInfo.sql**
3. Change the view to pull your real data. Let's change the from clause to point to your table, and "stub out" any columns not in your table:
  - a. Change the **From [Entity].[Customer] c** clause from [Customer] to your table name.
  - b. For columns that are in both the Customer table and your table:
    - i. Change the column names to match your table. For example, if your table's primary key is PersonID:
      1. Old: **Select            C.[CustomerID] As [ID]**
      2. New: **Select        C.[PersonID] As [ID]**
  - c. For columns that are in the [Customer] table but not in your table:
    - i. Alias all fields exactly as originally defined. For example, if your table does not have a Middle Name field, alias as follows
      1. Old: **c.[MiddleName],**
      2. New: **' ' As [MiddleName],**
4. Save the changes.
5. Publish the changes to your test database
  - a. Open **DataTier\Framework.Database\Publish\PublishToDev.publish.xml** file and click the **Publish** button.

The CustomerInfo view is now pulling data from your table.

### **Step 3: Run the WebApp or WebServices project to Enjoy Your New Resuable Framework!**

This is the best step of all, being done. The Genesys Framework is now selecting, inserting, updating and deleting data in your database in a cross-platform, full-stack and open-source way.

1. Ensure PresentationTier\Framework.WebApp is the startup project by:
  - a. Right-click PresentationTier\Framework.WebApp

- b. Click *Set as StartUp Project* from the context menu.
2. Press F5 or click the Play button to start the debugger.
3. Once the WebApp/WebService comes up, click Search menu-item to see your data.

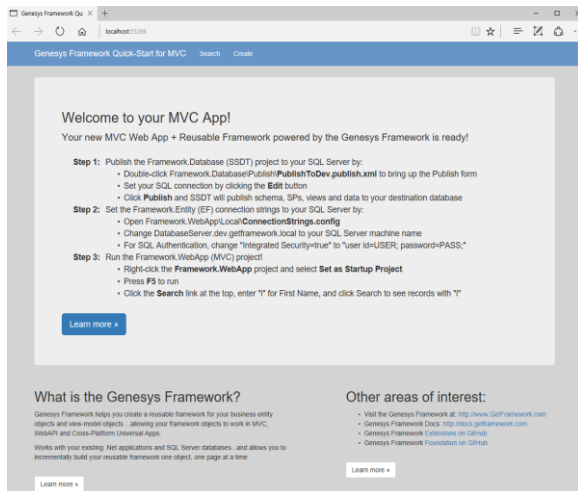


Figure 6: Frameeework.WebApp Home/Index page

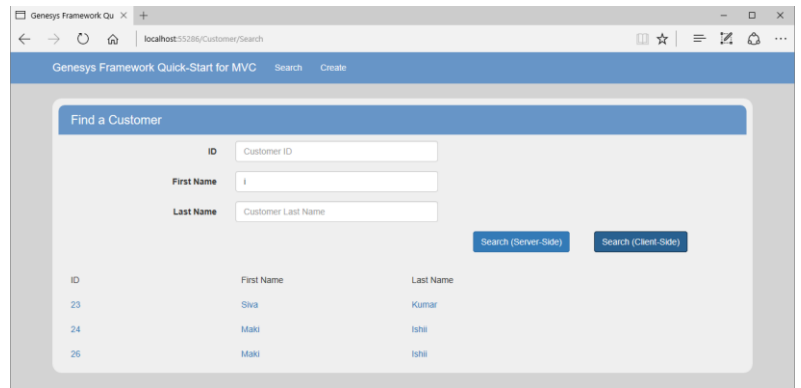


Figure 7: Framework.WebApp Customer/Search page.