GENESYSSOURCE

Genesys Source Framework

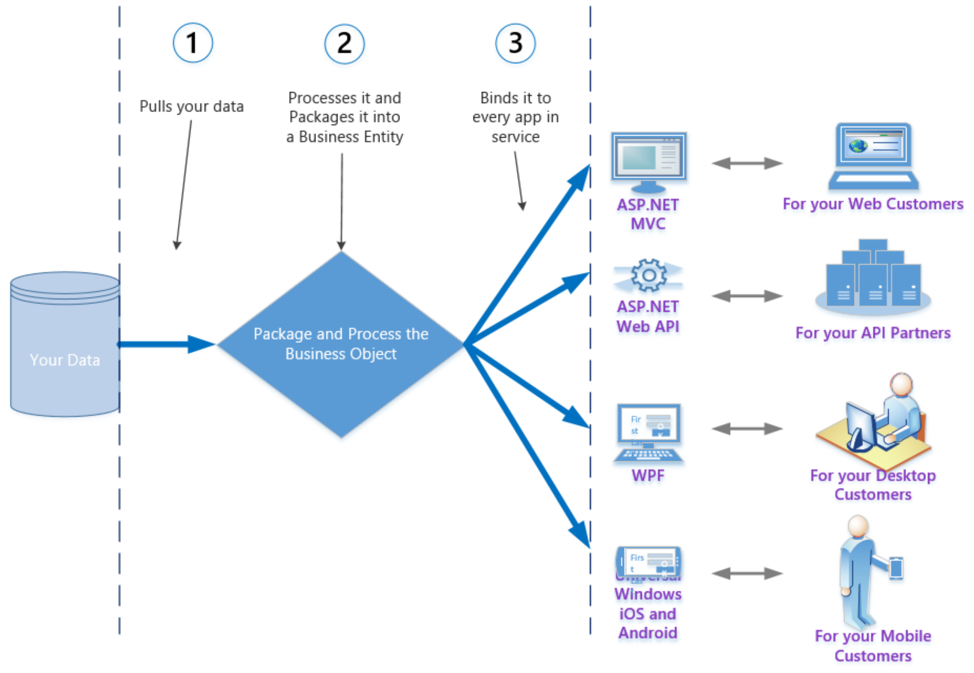
eBook

GENESYSSOURCE
22431 Antonio, Suite B160-843
Rancho Santa Margarita, Ca. 92688
+1.949.544.1900 | GenesysSource.com

Contents

One Framework – Your Data – Any Platform

The *Genesys Framework* is a C# framework that enables your business objects to run cross-platform, full-stack on .NET Core and .NET Framework, in Web or Mobile or Desktop…in minutes.


1
Pulls your data
2
Processes it and Packages it into a Business Entity
3
Binds it to every app in service
Your Data
Package and Process the Business Object
ASP.NET MVC
For your Web Customers
ASP.NET Web API
For your API Partners
WPF
For your Desktop Customers
Windows iOS and Android
For your Mobile Customers

Tell me a little more about the *Genesys Framework*

What is it?	The *Genesys Framework* is a full-stack business object framework, exposing your data as C# objects, from your SQL Database to any type of .NET app. Your *Customer* object, for example, can exist as a: ➢ Data object: EF-enabled to pull data from your SQL Server via common repository-pattern methods like: *GetById*(), *GetAll*(), *Save*() and *Delete*(). ➢ Domain object: Enriched domain object containing domain behavior such as *IsEmployee*, *IsActive*, *HasRegistered*, *CurrentStatus*, etc. ➢ View Model object: Model for your screen Views that are thin, atomic and transportable. Can be extended to have view-specific properties such as a Gender select list, without altering your *CustomerModel* class. ➢ Data-transfer object: Expose your object as a DTO for data-transfer specific operations such as inter-web-services data sharing or returning your object through public API endpoints.
Why do I care?	Full-stack projects in seconds: Your entire C# stack solution is up in seconds, ready for you to add your business objects. From Visual Studio: *File -> New -> Project*, select *Genesys Source Quick-Start*, and your stack framework is runnable and ready to code. Cross-platform by default: Enable your business-objects to run truly everywhere: In web apps - In any mobile app - In desktop native apps - As middle-tier domain service - As Public API web services - even in client-side TypeScript.

	Full-stack by default: Other frameworks focus only on the UI, only in the web, only for data access. *Genesys Framework* includes Database projects (SSDT), Middle Tier projects (.NET Standard) and projects for all major apps (MVC, Web API, UWP, WPF.)
How do I get it?	The *Genesys Source Framework* is available where you need it most: From Microsoft: ➢ At *Microsoft* on http://marketplace.VisualStudio.com ➢ On *Microsoft Azure* at http://bit.ly/2zw5UzN ➢ In Visual Studio on Tools -> Extensions and Updates From GitHub: ➢ On *GitHub* at http://github.com/genesyssource From Genesys Source: ➢ At *Genesys Source* at http://cloud.Genesyssource.com

1: Genesys Framework: One Framework – Your Data – Any Platform

Pre-requisites

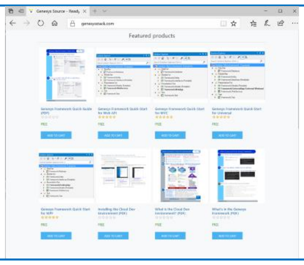
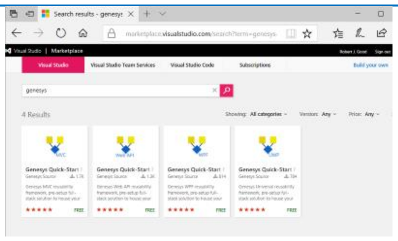
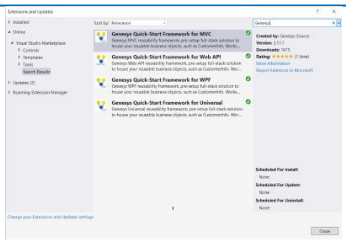
To get the most out of the *Genesys Framework*, the following skills are recommended:

- ✓ Moderate C# and .NET, HTML and XAML
- ✓ Low/Moderate T-SQL and Database design
- ✓ Awareness of N-tier, MVC, MVVM and REST
- ✓ Visual Studio Community (or greater) from https://www.visualstudio.com/downloads/
- ✓ SQL Server Management Studio from https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms

Installing the *Genesys Framework*

The *Genesys Framework* is free open-source on GitHub and available as a Zip download, a Vsix download or through Visual Studio Marketplace…installable with any method that you prefer most.

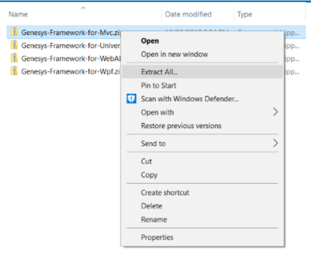
The easiest way to get the *Genesys Framework* is to download our Quick-Start projects for MVC, Web API, WPF and Universal. The Quick-Start projects are a small starter with the basics to get you running fast.

Get as Zip, Vsix or directly in Visual Studio			
① Get	Download Zip GenesysSource.com	Download Vsix VisualStudioMarketplace.com	Install in Visual Studio Tools -> Extensions and Updates
			
Framework for MVC	Download Framework-for-MVC.sln	Download Framework.MVC.vsix	Tools -> Extensions and Updates -> Search: GENESYS -> Download

Framework for Web API	Download Framework-for-WebAPI.sln	Download Framework.WebAPI.vsix	Tools -> Extensions and Updates -> Search: GENESYS -> Download
Framework for WPF	Download Framework-for-WPF.sln	Download Framework.WPF.vsix	Tools -> Extensions and Updates -> Search: GENESYS -> Download
Framework for Universal	Download Framework-for-Universal.sln	Download Framework.Universal.vsix	Tools -> Extensions and Updates -> Search: GENESYS -> Download
Framework for Core	Download Framework-for-Core.sln	Download Framework.Core.vsix	Tools -> Extensions and Updates -> Search: GENESYS -> Download

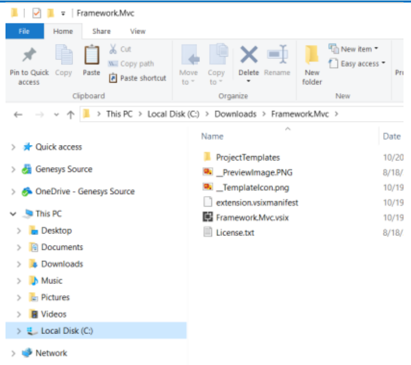
2 Install

Extract Zip


Name | Date modified | Type
Genesys-Framework-for-Mvc.zip
Genesys-Framework-for-Universal
Genesys-Framework-for-WebAPI
Genesys-Framework-for-Wpf.zip
Open
Open in new window
Extract All...
Pin to Start
Scan with Windows Defender...
Open with
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties

If downloaded as Zip: Extract to a local folder

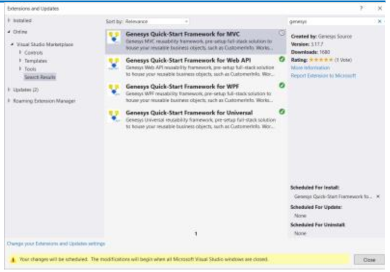
Run Vsix


Framework.Mvc
File | Home | Share | View
Pin to Quick access | Copy | Paste | Cut | Copy path | Paste shortcut | Move to | Copy to | Delete | Rename | New folder | New item | Easy access
This PC > Local Disk (C:) > Downloads > Framework.Mvc
Quick access
Genesys Source
OneDrive - Genesys Source
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Network

Name	Date
ProjectTemplates	10/20
_PreviewImage.PNG	8/18/
_TemplateIcon.png	10/19
extension.vsixmanifest	10/19
Framework.Mvc.vsix	10/19
License.txt	8/18/

If downloaded as a VSIX, double-click the .vsix file and follow instructions

Close Visual Studio to Install


Extensions and Updates
Installed
Online
Visual Studio Marketplace
Controls
Templates
Tools
Search Results
Updates (2)
Roaming Extension Manager
Sort by: Relevance
genesys

Genesys Quick-Start Framework for MVC
Genesys MVC reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works...

Genesys Quick-Start Framework for Web API
Genesys Web API reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Wor...

Genesys Quick-Start Framework for WPF
Genesys WPF reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works...

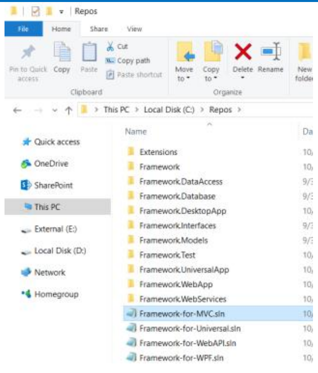
Genesys Quick-Start Framework for Universal
Genesys Universal reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Wor...

Created by: Genesys Source
Version: 3.17.7
Downloads: 1680
Rating: ★★★★★ (1 Vote)
More Information
Report Extension to Microsoft
Scheduled For Install:
Genesys Quick-Start Framework fo...
Scheduled For Update:
None
Scheduled For Uninstall:
None
Change your Extensions and Updates settings
Your changes will be scheduled. The modifications will begin when all Microsoft Visual Studio windows are closed.
Close

Close/Reopen Visual Studio to allow Extensions and Updates to install the Genesys Quick-Start

3 Open

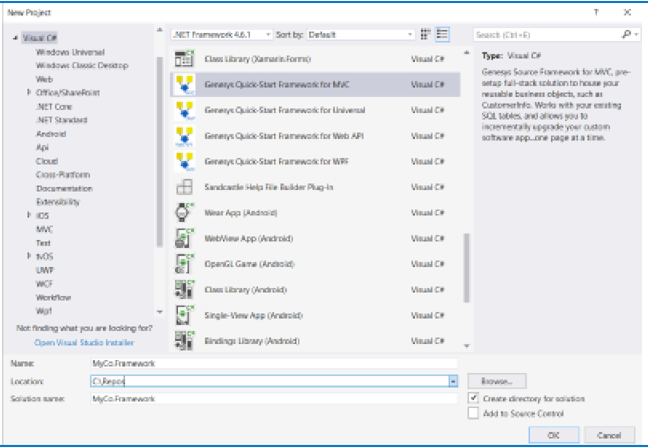
Open Solution File (.sln)


Repos
File | Home | Share | View
Pin to Quick access | Copy | Paste | Cut | Copy path | Paste shortcut | Move to | Copy to | Delete | Rename | New folder
This PC > Local Disk (C:) > Repos
Quick access
OneDrive
SharePoint
This PC
External (E:)
Local Disk (D:)
Network
Homegroup
Name
Extensions
Framework
Framework.DataAccess
Framework.Database
Framework.DesktopApp
Framework.Interfaces
Framework.Models
Framework.Test
Framework.UniversalApp
Framework.WebApp
Framework.WebServices
Framework-for-MVC.sln
Framework-for-Universal.sln
Framework-for-WebAPI.sln
Framework-for-WPF.sln

If downloaded as Zip: Extract and Open .sln file

Create Solution

File -> New Project -> C#


New Project
Visual C#
Windows Universal
Windows Classic Desktop
Web
Office/SharePoint
.NET Core
.NET Standard
Android
Api
Cloud
Cross-Platform
Documentation
Extensibility
iOS
MVC
Test
tvOS
UWP
WCF
Workflow
Wpf
Not finding what you are looking for?
Open Visual Studio Installer
.NET Framework 4.6.1 | Sort by: Default
Search (Ctrl+E)

Class Library (Xamarin.Forms)	Visual C#
Genesys Quick-Start Framework for MVC	Visual C#
Genesys Quick-Start Framework for Universal	Visual C#
Genesys Quick-Start Framework for Web API	Visual C#
Genesys Quick-Start Framework for WPF	Visual C#
Sandcastle Help File Builder Plug-In	Visual C#
Wear App (Android)	Visual C#
WebView App (Android)	Visual C#
OpenGL Game (Android)	Visual C#
Class Library (Android)	Visual C#
Single-View App (Android)	Visual C#
Bindings Library (Android)	Visual C#

Type: Visual C#
Genesys Source Framework for MVC, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works with your existing SQL tables, and allows you to incrementally upgrade your custom software app...one page at a time.
Name: MyCo.Framework
Location: C:\Repos
Solution name: MyCo.Framework
Browse...
[✓] Create directory for solution
[ ] Add to Source Control
OK | Cancel

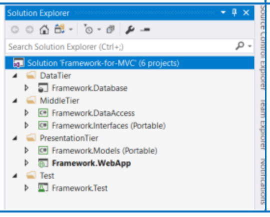
If installed via Extensions and Updates: File -> New -> Project -> C# -> Genesys Quick-Start

④

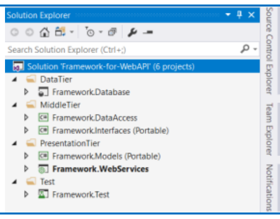
Build

Build All

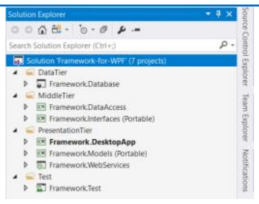
MVC


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Framework-for-MVC' (6 projects)
DataTier
Framework.Database
MiddleTier
Framework.DataAccess
Framework.Interfaces (Portable)
PresentationTier
Framework.Models (Portable)
Framework.WebApp
Test
Framework.Test

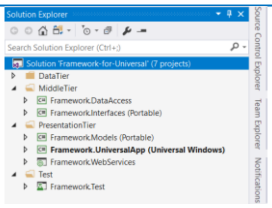
Web API


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Framework-for-WebAPI' (6 projects)
DataTier
Framework.Database
MiddleTier
Framework.DataAccess
Framework.Interfaces (Portable)
PresentationTier
Framework.Models (Portable)
Framework.WebServices
Test
Framework.Test

WPF


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Framework-for-WPF' (7 projects)
DataTier
Framework.Database
MiddleTier
Framework.DataAccess
Framework.Interfaces (Portable)
PresentationTier
Framework.DesktopApp
Framework.Models (Portable)
Framework.WebServices
Test
Framework.Test

Universal


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Framework-for-Universal' (7 projects)
DataTier
MiddleTier
Framework.DataAccess
Framework.Interfaces (Portable)
PresentationTier
Framework.Models (Portable)
Framework.UniversalApp (Universal Windows)
Framework.WebServices
Test
Framework.Test

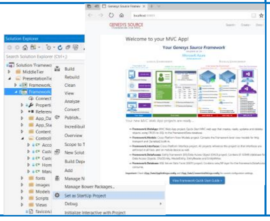
Right-click solution -> Configuration Manager -> Check Build and Deploy
Right-click the solution -> Rebuild solution

⑤

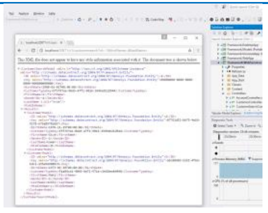
Run

Run!

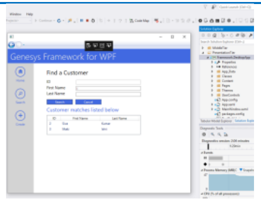
MVC



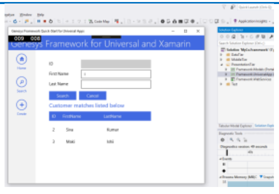
Web API



WPF



Universal



Right-click the project -> Set as Startup Project
Press *F5* or ▶ to run

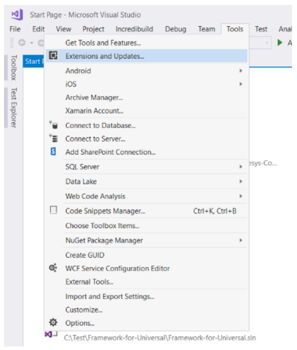
1: Installing Genesys Source Framework .NET Solutions

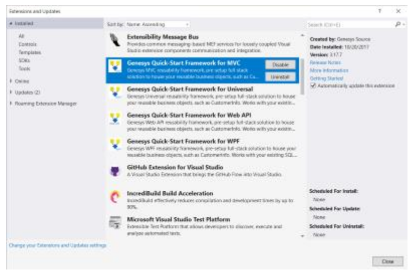
Uninstalling the *Genesys Framework*

If the *Genesys Framework* was downloaded via Zip file, no uninstallation is necessary.

For installs through the Visual Studio Marketplace or in Visual Studio, follow these steps to uninstall the Genesys Framework from your IDE:

①	②	③
Open Visual Studio	Extensions and Updates	Click Uninstall and Close Visual Studio





Extensibility Message Bus
Provides common messaging-based MEF services for loosely coupled Visual Studio extension components communication and integration.

Genesys Quick-Start Framework for MVC
Genesys MVC reusability framework, pre-setup full-stack solution to house your reusable business objects, such as Cu...

Genesys Quick-Start Framework for Universal
Genesys Universal reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works with your existin...

Genesys Quick-Start Framework for Web API
Genesys Web API reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works with your existin...

Genesys Quick-Start Framework for WPF
Genesys WPF reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works with your existing SQL...

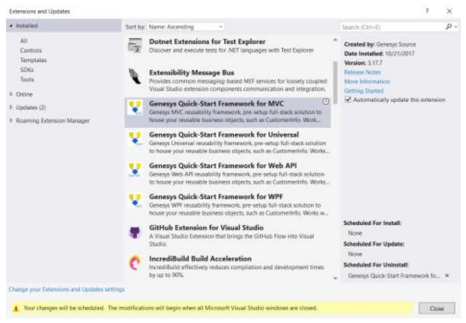
GitHub Extension for Visual Studio
A Visual Studio Extension that brings the GitHub Flow into Visual Studio.

IncrediBuild Build Acceleration
IncrediBuild effectively reduces compilation and development times by up to 90%.

Microsoft Visual Studio Test Platform
Extensible Test Platform that allows developers to discover, execute and analyze automated tests.

Created by: Genesys Source
Date Installed: 10/20/2017
Version: 3.17.7
☑ Automatically update this extension

Scheduled For Install: None
Scheduled For Update: None
Scheduled For Uninstall: None



Dotnet Extensions for Test Explorer
Discover and execute tests for .NET languages with Test Explorer

Extensibility Message Bus
Provides common messaging-based MEF services for loosely coupled Visual Studio extension components communication and integration.

Genesys Quick-Start Framework for MVC
Genesys MVC reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Work...

Genesys Quick-Start Framework for Universal
Genesys Universal reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works...

Genesys Quick-Start Framework for Web API
Genesys Web API reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works...

Genesys Quick-Start Framework for WPF
Genesys WPF reusability framework, pre-setup full-stack solution to house your reusable business objects, such as CustomerInfo. Works w...

GitHub Extension for Visual Studio
A Visual Studio Extension that brings the GitHub Flow into Visual Studio.

IncrediBuild Build Acceleration
IncrediBuild effectively reduces compilation and development times by up to 90%.

Created by: Genesys Source
Date Installed: 10/21/2017
Version: 3.17.7
☑ Automatically update this extension

Scheduled For Install: None
Scheduled For Update: None
Scheduled For Uninstall: Genesys Quick-Start Framework fo...

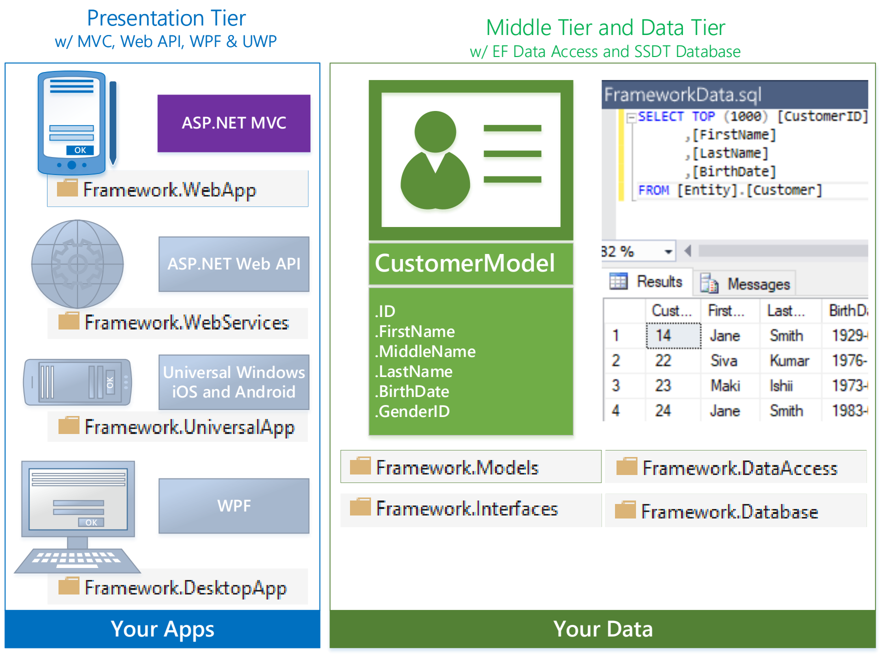
Your changes will be scheduled. The modifications will begin when all Microsoft Visual Studio windows are closed.

2: Uninstalling the Genesys Framework

What is in the *Genesys Framework*?

The *Genesys Framework* includes everything you need to build your business object framework quickly and with a minimal learning curve.

A *Genesys Framework* app includes full-stack projects for your application. From the database (SSDT), to data objects (EF), to models (.NET Core or Framework), exposed in any .NET application type such as MVC or UWP.



```
SELECT TOP (1000) [CustomerID]
      ,[FirstName]
      ,[LastName]
      ,[BirthDate]
  FROM [Entity].[Customer]
```

	Cust...	First...	Last...	BirthD
1	14	Jane	Smith	1929-
2	22	Siva	Kumar	1976-
3	23	Maki	Ishii	1973-
4	24	Jane	Smith	1983-

3: Genesys Framework Code and Runtime

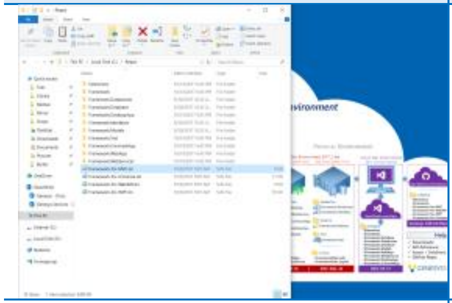
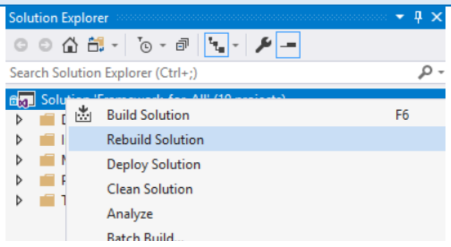
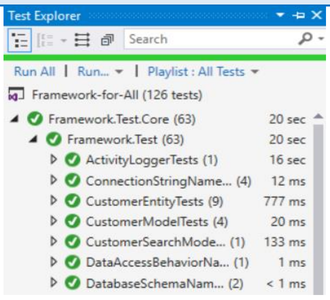
Visual Studio Project	
Framework.WebApp	MVC Web App with all CRUD and Search operations for a Customer entity. Points of interest are: ➢ \App_Data\ConnectionStrings.json – Database connection information ➢ \Views\Home\Index.cshtml – Home Page ➢ \Controllers\Customer\CustomerSearchController.cs – Processes all customer search requests
Framework.WebServices	Web API web services with all CRUD and Search operations for a Customer entity. Points of interest are: ➢ \App_Data\ConnectionStrings.json – Database connection information ➢ \Views\Home\Index.cshtml – Home Page ➢ \Controllers\Customer\CustomerSearchController.cs – Processes all customer search requests

Framework.UniversalApp	UWP Cross-Platform App with all CRUD and Search operations for a Customer entity. Points of interest are: ➢ \App_Data\ConnectionStrings.json – Database connection information ➢ \MainPage.xaml – Home Page ➢ \Pages\Customer\CustomerSearch.xaml – Processes all customer search requests
Framework.DesktopApp	WPF Desktop App with all CRUD and Search operations for a Customer entity. Points of interest are: ➢ \App_Data\ConnectionStrings.json – Database connection information ➢ \MainPage.xaml – Home Page ➢ \Pages\Customer\CustomerSearch.xaml – Processes all customer search requests
Framework.Models	Cross-platform PCL containing bindable screen models for MVC, WPF, UWP, WebForms, WinForms, Xamarin. Points of interest are: ➢ \Customer\CustomerModel.cs – View Model for Customer business object
Framework.Interop	Cross-platform PCL containing interfaces, to ensure all tiers share the same signature. Points of interest are: ➢ \Customer\ICustomer.cs – Interface ensuring compatibility between all Customer objects
Framework.DataAccess	Entity Framework data access objects, providing CRUD operations for Customer. Points of interest are: ➢ \Customer\CustomerInfo.cs – Data Access Object for Customer business object
Framework.Database	SSDT database containing all T-SQL for tables, views, stored procs, schemas, users. Points of interest are: ➢ \Tables\Customer\Customer.sql – Customer table ➢ \Views\CustomerCode\CustomerInfo.sql – View that connects table and code ➢ \Stored Procedures\CustomerCode\CustomerInsert.sql – Stored procedure that inserts to customer table

2: Genesys Framework .NET Projects

Running the *Framework.Test* Integration Tests

All products contain *Framework.Test*, an integration test project that tests your objects and support classes. To run all tests in the solution:

① Open your Solution *i.e. Framework-for-MVC.sln*	② Rebuild All on the Solution	③ Click Run All in the Test Explorer Window
	 Solution Explorer Search Solution Explorer (Ctrl+;) Build Solution F6 Rebuild Solution Deploy Solution Clean Solution Analyze Batch Build...	 Test Explorer Search Run All | Run... | Playlist : All Tests Framework-for-All (126 tests) Framework.Test.Core (63) 20 sec Framework.Test (63) 20 sec ActivityLoggerTests (1) 16 sec ConnectionStringName... (4) 12 ms CustomerEntityTests (9) 777 ms CustomerModelTests (4) 20 ms CustomerSearchMode... (1) 133 ms DataAccessBehaviorNa... (1) 1 ms DatabaseSchemaNam... (2) < 1 ms
1. Open your solution, i.e. *Framework-for-MVC.sln* Visual Studio solution file	2. Right-click the solution and click *Rebuild Solution*	3. Open the *Test Explorer* window - *Test -> Windows -> Test Explorer* 4. Click *Run All* to run all tests - All tests should execute successfully

		Hint for LocalDB: Check SSMS Server *(LocalDb)\MSSQLLocalDB* if *FrameworkData_Primary.mdf* is locked.

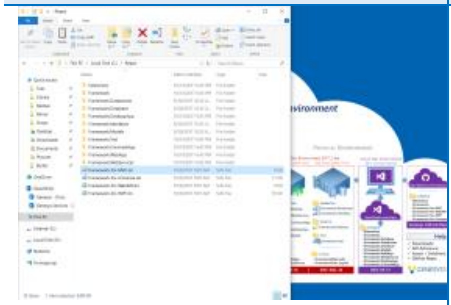
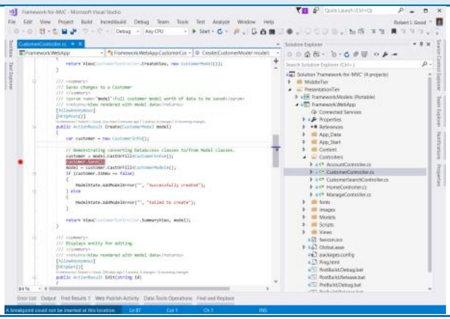
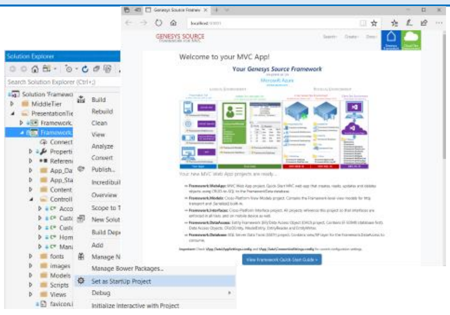
4: Running Framework.Test

Debugging the *Genesys Framework*

The *Genesys Framework* contains App and Services projects that host your application. These Apps can be debugged using standard .NET debugging techniques in Visual Studio Community (or greater.)

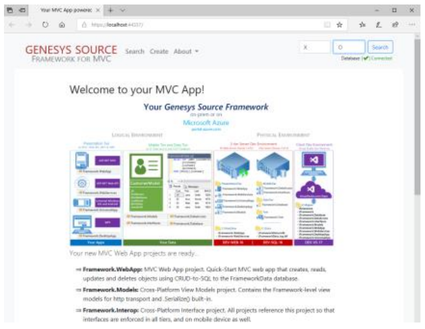
Debugging Framework.WebApp (MVC) and Framework.WebServices (Web API)

To debug your Framework for MVC and Framework for Web API app, follow the procedures below:

1 Open the Solution *I.e. Framework-for-MVC.sln*	2 Set Breakpoint in *CustomerSearchController.cs*	3 Set as StartUp Project and Press *F5 to Run*
		

5. Open the *Framework-for-MVC.sln* Visual Studio solution file
- Default: C:\Source\Framework-for-MVC.sln
6. Navigate to and open Framework.WebApp\Controllers\CustomerController
	7. Set a breakpoint in the `Search()` method
	`public ActionResult Search(CustomerModel data)`
	`{`
	`  var model = new CustomerSearchModel();`
	`  var searchResults =`
	`    CustomerInfo.GetByAny(data).Take(25);`
		8. Right-click *Framework.WebApp* or *Framework.WebServices* project -> click *Set as StartUp Project*
		9. Press *F5* or ▶ to run
		- WebApp Url: http://localhost:30001/
		- WebServices Url: http://localhost:30002/
		10. Home/Index.cshtml should display

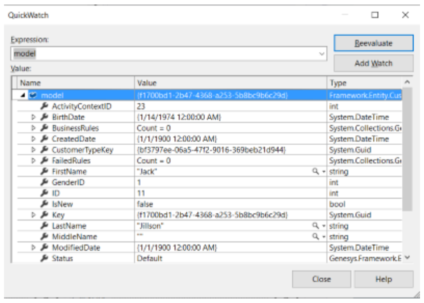
4 Enter First (x) and Last (o) -> Click Search	5 Step-in to CustomerInfo.cs	6 See Customer search data before returning to View


Welcome to your MVC App!

Your new MVC Web App projects are ready...

- **Framework.WebApp:** MVC Web App project. Quick-Start MVC web app that creates, reads, updates and deletes objects using CRUD-to-SQL to the FrameworkData database.
- **Framework.Models:** Cross-Platform View Models project. Contains the Framework-level view models for http transport and .Serialize() built-in.
- **Framework.Interop:** Cross-Platform Interface project. All projects reference this project so that interfaces are enforced in all tiers, and on mobile device as well.

11. Search from the home header
12. Enter a single letter (x) into First Name and a single letter (o) into Last Name
13. Click *Search* to hit breakpoint

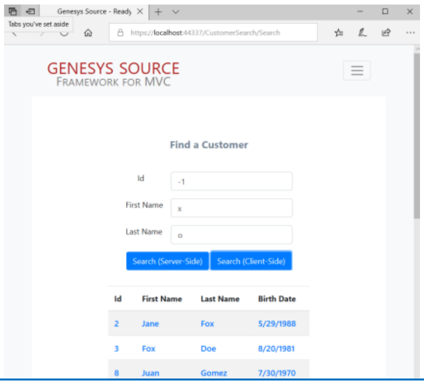

QuickWatch

Expression: model

Value:

Name	Value	Type
model	{f1700bd1-2b47-4368-a253-5b8bc9b6c29d}	Framework.Entity.Cus
ActivityContextID	23	int
BirthDate	{1/14/1974 12:00:00 AM}	System.DateTime
BusinessRules	Count = 0	System.Collections.G
CreatedDate	{1/1/1900 12:00:00 AM}	System.DateTime
CustomerTypeKey	{bf3797ee-06a5-47f2-9016-369beb21d944}	System.Guid
FailedRules	Count = 0	System.Collections.G
FirstName	"Jack"	string
GenderID	1	int
ID	11	int
IsNew	false	bool
Key	{f1700bd1-2b47-4368-a253-5b8bc9b6c29d}	System.Guid
LastName	"Jillson"	string
MiddleName	""	string
ModifiedDate	{1/1/1900 12:00:00 AM}	System.DateTime
Status	Default	Genesys.Framework.E

14. Press F10 to step-over to `if (searchResults.Any())`
15. Select **`searchResults`**
16. Press SHIFT+F9 to see the Quick Watch window


Find a Customer

Id: -1

First Name: x

Last Name: o

Id	First Name	Last Name	Birth Date
2	Jane	Fox	5/29/1988
3	Fox	Doe	8/20/1981
8	Juan	Gomez	7/30/1970

17. Continue execution by pressing F5 to see the search complete

5: Debugging Framework for MVC and Web API

Debugging Framework.UniversalApp (UWP) and Framework.DesktopApp (WPF)

To debug your Framework for Universal and Framework for WPF app, follow the procedures below:

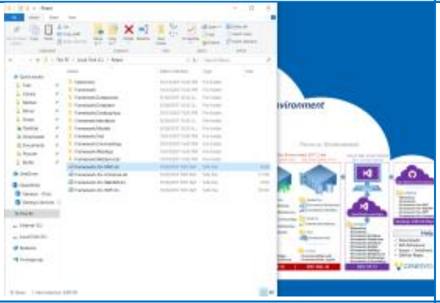
① Open the Solution

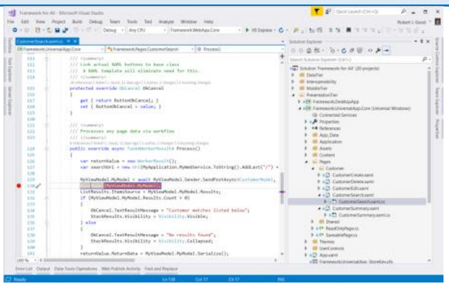
I.e. Framework-for-Uwp.sln

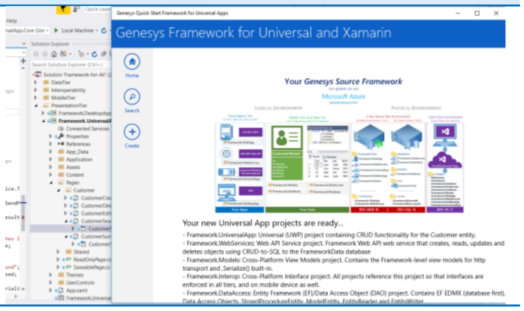
② Set Breakpoint in

CustomerSearch.xaml.cs

③ Set as StartUp Project and Press *F5 to Run*







1. Open the *Framework-for-Uwp.sln* Visual Studio solution file
2. Navigate to and open Customer Search page:

\Framework.UniversalApp.Core\Pages\Customer\CustomerSearch.xaml.cs

3. Set a breakpoint in the **`Process()`** method

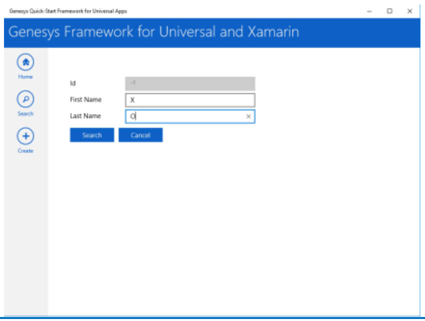
```
public override async Task<WorkerResult>
Process()
{
    ...
    BindModel(MyViewModel.MyModel);
```

4. Right-click *Framework.UniversalApp* or *Framework.DesktopApp* project -> click *Set as StartUp Project*
5. *Right-click solution -> Configuration Manager* -> Check *Build* and *Deploy*
6. Press *F5* or ▶ to run
7. \MainPage.xaml should display

④ Enter First (x) and Last (o) -> Click Search

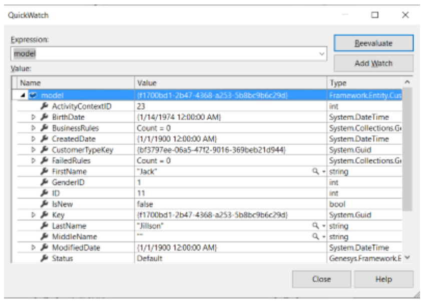
⑤ Step-in to CustomerInfo.cs

⑥ See Customer search data before returning to View


Genesys Quick-Start Framework for Universal Apps
Genesys Framework for Universal and Xamarin
Home
Search
Create
Search
Cancel

Id	-1
First Name	X
Last Name	o

8. Click *Search* icon on left
9. Enter a single letter (x) into First Name and a single letter (o) into Last Name
10. Click *Search* to hit breakpoint

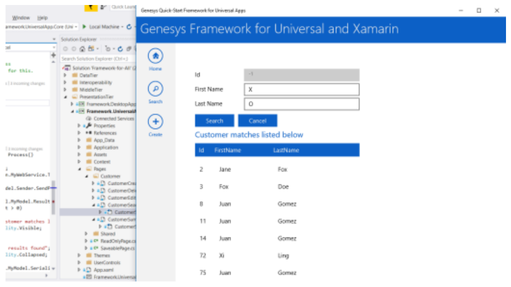


Expression: model

Value:

Name	Value	Type
model	{f1700bd1-2b47-4368-a253-5b8bc9b6c29d}	Framework.Entity.Cus
ActivityContextID	23	int
BirthDate	{1/14/1974 12:00:00 AM}	System.DateTime
BusinessRules	Count = 0	System.Collections.G
CreatedDate	{1/1/1900 12:00:00 AM}	System.DateTime
CustomerTypeKey	{bf3797ee-06a5-47f2-9016-369beb21d944}	System.Guid
FailedRules	Count = 0	System.Collections.G
FirstName	"Jack"	string
GenderID	1	int
ID	11	int
IsNew	false	bool
Key	{f1700bd1-2b47-4368-a253-5b8bc9b6c29d}	System.Guid
LastName	"Jillson"	string
MiddleName	""	string
ModifiedDate	{1/1/1900 12:00:00 AM}	System.DateTime
Status	Default	Genesys.Framework.E

11. Press F10 to step-over to `BindModel(MyViewModel.MyModel);`
12. Select `MyViewModel.MyModel`
13. Press SHIFT+F9 to see the Quick Watch window


Genesys Quick-Start Framework for Universal Apps
Genesys Framework for Universal and Xamarin
Home
Search
Create
Search
Cancel
Customer matches listed below

Id	-1
First Name	X
Last Name	O

Id	FirstName	LastName
2	Jane	Fox
3	Fox	Doe
8	Juan	Gomez
11	Juan	Gomez
14	Juan	Gomez
72	Xi	Ling
75	Juan	Gomez

14. Continue execution by pressing F5 to see the search complete

6: Debugging Framework for UWP and WPF

The *Framework.Database project and FrameworkData* database

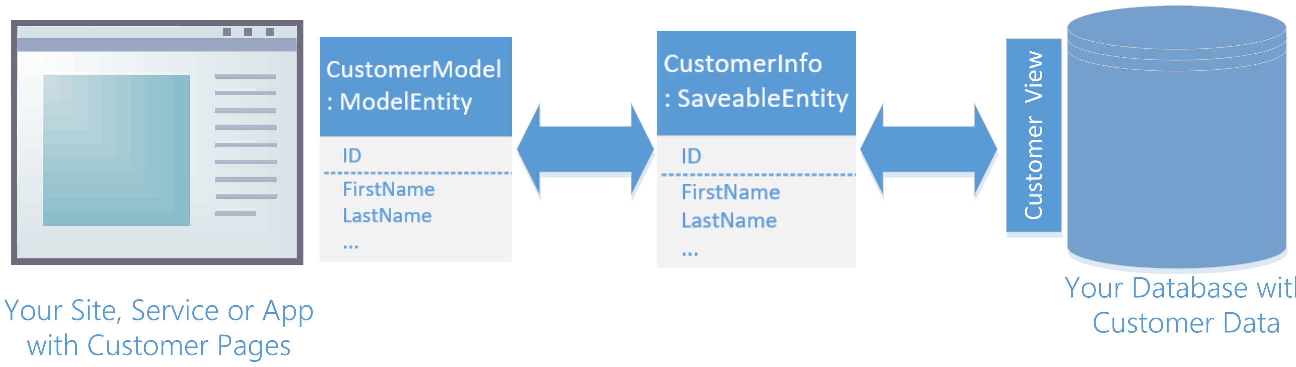
Genesys Framework includes the *Framework.Database* project, which contains a micro-database as a go-between your database and your new code stack projects. Don't have an existing database? No problem, as the *Genesys Framework* default operates 100% on the *FrameworkData* database that *Framework.Database* creates.

Framework.Database is Loose-coupled to your SQL Tables using Views

The *Genesys Framework* pulls data using Entity Framework, which can be tight-coupled directly to SQL Tables, or loose-coupled to SQL Views and Stored Procedures.

Out of the box, *Genesys Framework*:

- Connects to the *FrameworkData* database
- Selects data from SQL Views, i.e. *FrameworkData.CustomerCode.CustomerInfo*
- Insert, update and delete through SQL Stored Procedures, i.e. *FrameworkData.CustomerCode.CustomerInsert*
- *Framework.DataAccess* project contains Repository and Data objects for the data, i.e. *Customer\CustomerInfo.cs*
- In the App project, data is exposed as View Model objects, i.e. *Customer\CustomerModel.cs*



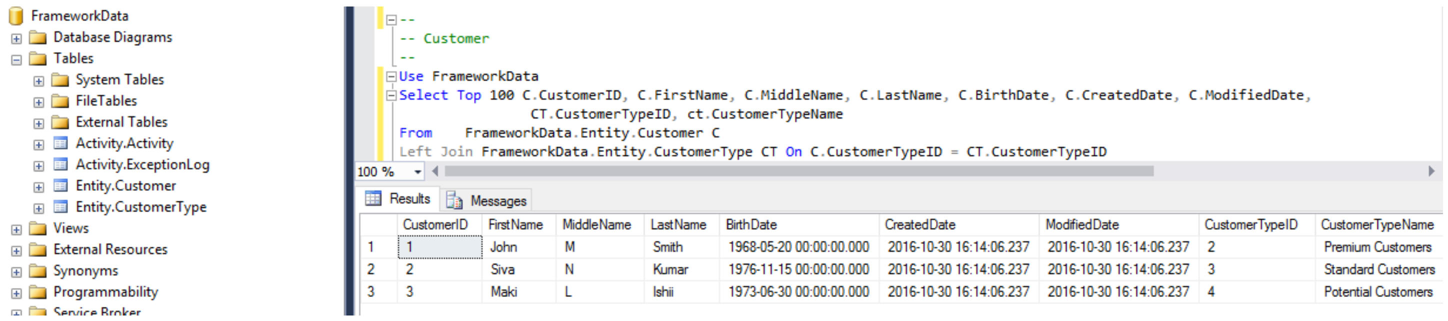
7: Data passing through Framework objects

About Framework.Database (SSDT)

The *Framework.Database* is a SQL Server project built on SQL Server Data Tools (SSDT). This project is responsible for:

1. Holds T-SQL for tables, schemas, indexes, constraints, users and roles
2. Holds and runs the PreDeployment and PostDeployment scripts
3. DB Compare the *Framework.Database* project to the *FrameworkData* database
4. Publishes the *Framework.Database* project to the *FrameworkData* database

Once deployed, you test the *FrameworkData* database as any other SQL Server database. Select from the Customer tables and *CustomerCode* views. Insert, update and delete from the *CustomerCode* stored procedures.



```
--
-- Customer
--
Use FrameworkData
Select Top 100 C.CustomerID, C.FirstName, C.MiddleName, C.LastName, C.BirthDate, C.CreatedDate, C.ModifiedDate,
               CT.CustomerTypeID, ct.CustomerTypeName
From    FrameworkData.Entity.Customer C
Left Join FrameworkData.Entity.CustomerType CT On C.CustomerTypeID = CT.CustomerTypeID
```

	CustomerID	FirstName	MiddleName	LastName	BirthDate	CreatedDate	ModifiedDate	CustomerTypeID	CustomerTypeName
1	1	John	M	Smith	1968-05-20 00:00:00.000	2016-10-30 16:14:06.237	2016-10-30 16:14:06.237	2	Premium Customers
2	2	Siva	N	Kumar	1976-11-15 00:00:00.000	2016-10-30 16:14:06.237	2016-10-30 16:14:06.237	3	Standard Customers
3	3	Maki	L	Ishii	1973-06-30 00:00:00.000	2016-10-30 16:14:06.237	2016-10-30 16:14:06.237	4	Potential Customers

8: Selecting from the Customer table

Re-wire *Framework.Database* SQL Views to connect to your SQL Tables

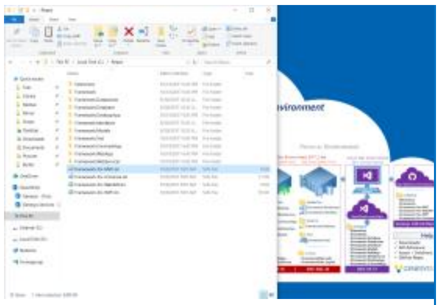
This procedure guides you through the process of re-wiring *Framework.Database.CustomerCode.CustomerInfo* view to pull data from your "Person" table. This is an example of a one-to-one swap:

1. Edit the *CustomerInfo* view.
2. De-reference the *FrameworkData Customer* table.
3. Reference your "*Person*" table that contains *First Name* or *Last Name*.
4. Alias all mismatched or missing fields in the view

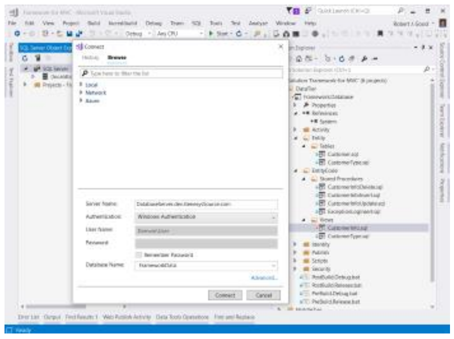
The *Genesys Framework* will now pull data via the *CustomerInfo* view, from your "Person" table

Important Tip: *For this example, keep the field names the same and column type the same (use AS keyword.) No code changes will be necessary. The existing Framework projects will work against your "Person" table as if pulling from the FrameworkData's Customer table.*

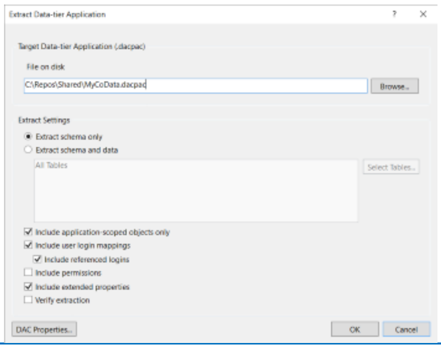
①	②	③
Open your Solution *i.e. Framework-for-MVC.sln*	Connect to your database in SQL Object Explorer	Extract your database schema to a .dacpac file



1. Open the *Framework-for-MVC.sln* Visual Studio solution file
   - Default: C:\Source\Framework-for-MVC.sln



2. Click *View -> SQL Server Object Explorer*
3. Enter connection info to your database
4. Click *Connect* to add the connection



Extract Data-tier Application

Target Data-tier Application (.dacpac)

File on disk: C:\Repos\Shared\MyCoData.dacpac

Extract Settings

(●) Extract schema only
( ) Extract schema and data

All Tables

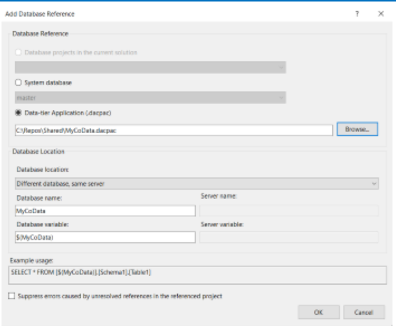
☑ Include application-scoped objects only
☑ Include user login mappings
☑ Include referenced logins
☐ Include permissions
☑ Include extended properties
☐ Verify extraction

5. In SQL Server Object Explorer, right-click your database -> click *Extract Data-tier Application*
6. Select: *Extract Schema Only*
7. Enter file-on-disk as:
   C:\Source\Shared\MyCoData.dacpac
8. Click *OK* to extract your schema to .dacpac

④ Add a Database Reference to your .dacpac

⑤ Open View Views\CustomerCode\CustomerInfo.sql

⑥ Replace the SELECT with T-SQL that pulls data from your table



Add Database Reference

Database Reference

☐ Database projects in the current solution

☐ System database

master

(●) Data-tier Application (.dacpac)

C:\Repos\Shared\MyCoData.dacpac

Database Location

Database location: Different database, same server

Database name: MyCoData

Server name:

Database variable: $(MyCoData)

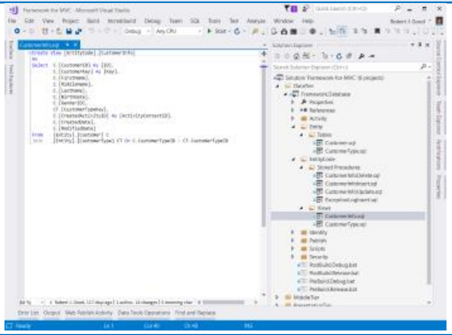
Server variable:

Example usage:

SELECT * FROM [$(MyCoData)].[Schema1].[Table1]

☐ Suppress errors caused by unresolved references in the referenced project

9. In Solution Explorer, right-click your Framework.Database\Views \CustomerCode\CustomerInfo.sql



10. Navigate to and open Customer view: Framework.Database\Views \CustomerCode\CustomerInfo.sql

For example…

If your table is: *[MyCoData].[dbo].[Cust]*

With fields: *Cust_ID, F_Name, L_Name, B_Date*

Change the SELECT to your [Cust] table…

```
Create View [CustomerCode].[CustomerInfo] As
Select     C.[Cust_ID] As [ID],
 C.[F_Name] As [FirstName],
 C.[L_Name] As [LastName],
 C.[B_Date] As [BirthDate],
... (Alias Missing Fields Here) ...
From       [MyCoData].[dbo].[Cust] C
```

11. In CustomerInfo.cs, change the SELECT statement to pull data from your database

Note: Databases must be in same SQL instance

⑦ Alias any missing fields with Default Values

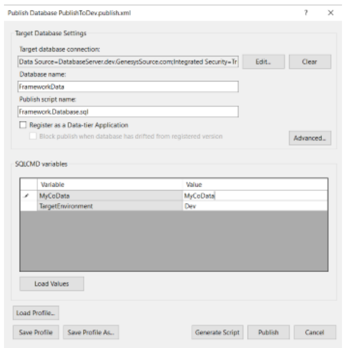
⑧ Publish *FrameworkData* to SQL Server

⑨ Run *Framework.WebApp* to pull your customer data

Alias missing fields for [Cust] example…

```
'' As [MiddleName],
-1 As [GenderID],
-1 As [ActivityContextID],
'00000000-0000-0000-0000-000000000000' As
    [Key],
'00000000-0000-0000-0000-000000000000' As
    [CustomerTypeKey],
'01/01/1900' As [CreatedDate],
'01/01/1900' As [ModifiedDate]
```

12. Alias all fields that do not have an equivalent in your customer data:
- Integer: `-1`
- String: `''`
- Date: `'01/01/1900'`
- Guid: `'00000000-0000-0000-0000-000000000000'`



Publish Database PublishToDev.publish.xml

Target Database Settings

Target database connection: Data Source=DatabaseServer.dev.GenesysSource.com;Integrated Security=Tr

Database name: FrameworkData

Publish script name: Framework.Database.sql

☐ Register as a Data-tier Application

☐ Block publish when database has drifted from registered version

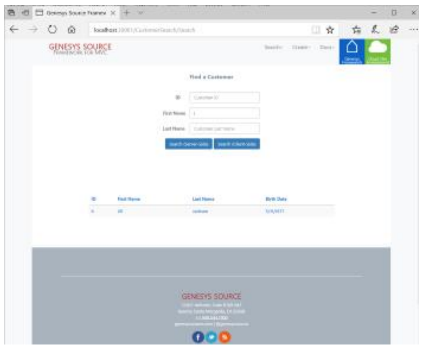
SQLCMD variables

Variable	Value
MyCoData	MyCoData
TargetEnvironment	Dev

13. Open SSDT publish screen:

Framework.Database\Publish\PublishToDev.publish.xml
- Ensure *Target database connection* is correct
- Ensure MyCoData is set to the name of your database

14. Click *Generate Script* and review
15. Click *Publish* to push changes to SQL



16. Ensure connection string is correct:

Framework.WebApp\App_Data\ConnectionStrings.json

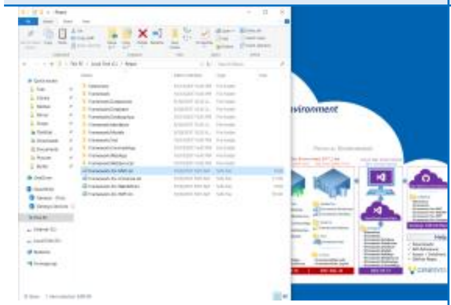
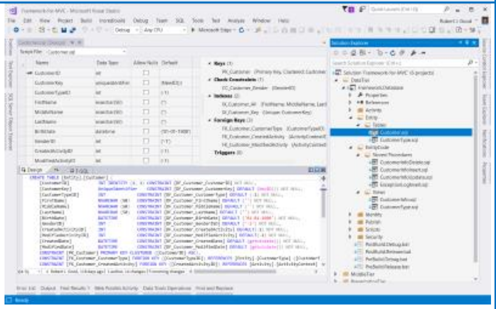
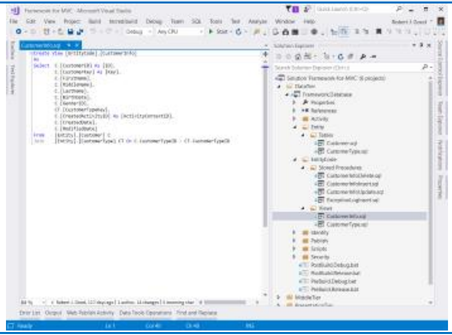
17. Right-click *Framework.WebApp* -> click *Set as Startup Project*
18. Press *F5* or ▶ to run
    - Should run this Url: http://localhost:30001/

Search screen & customer object now pulls your data

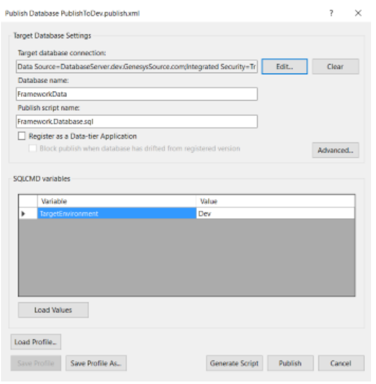
9: Pulling your data through the CustomerInfo object

Add a new Field/Property to the Customer object

This procedure walks you through the process of adding, changing or deleting an entity field. Including the column in the database, the data access object, the model and a MVC View.

① Open *Framework-for-MVC.sln*	② Open Table & Add Column *Tables\Customer\Customer.sql*	③ Open View & Add Column *Views\CustomerCode\CustomerInfo.sql*
		
1. Open the *Framework-for-MVC.sln* Visual Studio solution file - Default: C:\Source\Framework-for-MVC.sln	2. Navigate to and open Customer table: Framework.Database\Tables\CustomerCode\CustomerInfo.sql 3. Add a new field: NickName `[NickName] NVARCHAR (50) CONSTRAINT [DF_Customer_NickName] DEFAULT ('') NOT NULL,`	4. Navigate to and open Customer view: Framework.Database\Views\CustomerCode\CustomerInfo.sql 5. Add the NickName field: `C.[NickName],`
④	⑤ Open *CustomerInfo.cs in*	⑥

Publish *FrameworkData* to SQL Server



Publish Database PublishToDev.publish.xml

Target Database Settings

Target database connection: Data Source=DatabaseServer.dev.GenesysSource.com;Integrated Security=Tr

Database name: FrameworkData

Publish script name: Framework.Database.sql

☐ Register as a Data-tier Application

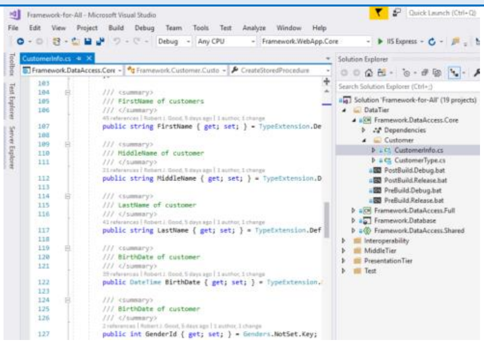
☐ Block publish when database has drifted from registered version

SQLCMD variables

Variable	Value
TargetEnvironment	Dev

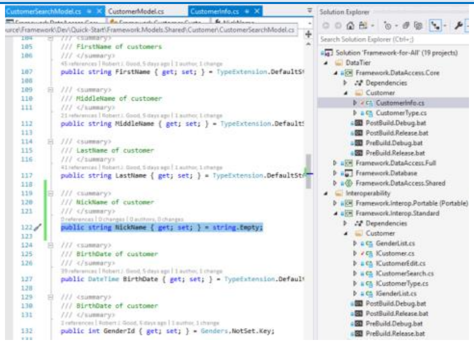
6. Open SSDT publish screen: *Framework.Database\Publish \PublishToDev.publish.xml* - Ensure *Target database connection* is correct
7. Click *Generate Script* and review
8. Click *Publish* to push changes to SQL

Framework.DataAccess



9. Open *CustomerInfo.cs* - *Framework.DataAccess\Customer*

Add NickName to *CustomerInfo*

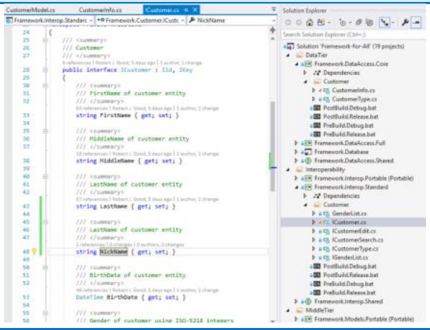


```
/// FirstName of customers
/// </summary>
public string FirstName { get; set; } = TypeExtension.DefaultSt

/// <summary>
/// MiddleName of customer
/// </summary>
public string MiddleName { get; set; } = TypeExtension.Default

/// <summary>
/// LastName of customer
/// </summary>
public string LastName { get; set; } = TypeExtension.DefaultSt

/// <summary>
/// NickName of customer
/// </summary>
public string NickName { get; set; } = string.Empty;

/// <summary>
/// BirthDate of customer
/// </summary>
public DateTime BirthDate { get; set; } = TypeExtension.Defaul

/// <summary>
/// BirthDate of customer
/// </summary>
public int GenderId { get; set; } = Genders.NotSet.Key;
```

10. Add NickName by Copy/Paste the following property:

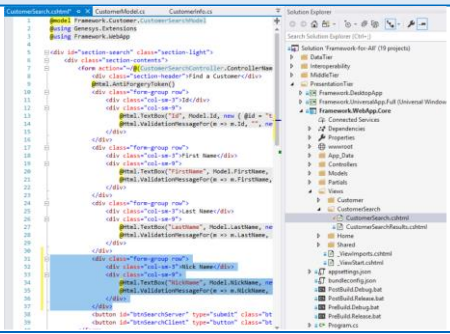
```
public string NickName { get; set; } = string.Empty;
```

7 Add NickName to ICustomer & Models



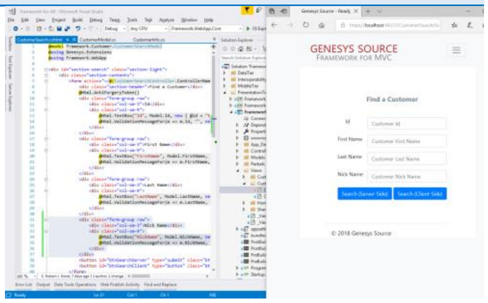
11. Open Framework.Interfaces\Customer \ICustomer.cs
12. Add NickName property as a string - Notice all classes that implement ICustomer throw an error requiring ICustomer.NickName
13. Add NickName to all dependent models (CustomerModel and CustomerSearchModel)

8 Add NickName to Framework.WebApp Search



14. Open Framework.WebApp \Views\CustomerSearch \CustomerSearchResults.cshtml
15. Add Nick Name to table header and body

9 Run!



16. Double-check the connection string, to make sure it is pointing to the proper database
17. Right-click *Framework.WebApp* project -> click *Set as StartUp Project*
18. Press *F5* or ▶ to run

Publishing the *Genesys Framework* to IIS and SQL Server

For the *Genesys Framework* to function in your dev or production environments, you minimally need:

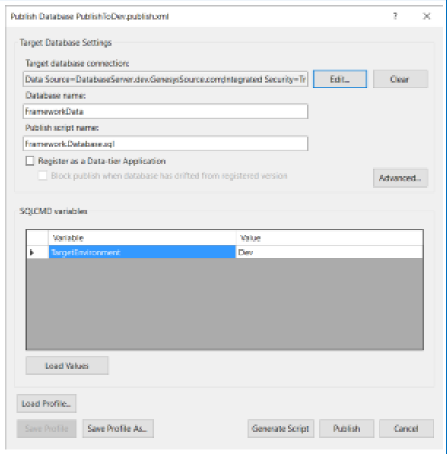
1. Framework.Database project published to a SQL Server or SQL Express

2. At least one Presentation Tier project, such as Framework.WebApp, published to an IIS Server

Publishing Framework.Database (SSDT) to a SQL Server

This procedure describes publishing the *Framework.Database* SSDT project to your SQL Server. The *Framework.Database* project holds all database objects for the *FrameworkData* database. Including tables, schemas, logins, users, views, stored procedures, etc.

① Build



Publish Database PublishToDev.publish.xml

Target Database Settings

Target database connection: Data Source=DatabaseServer.dev.GenesysSource.com;Integrated Security=Tr

Database name: FrameworkData

Publish script name: Framework.Database.sql

☐ Register as a Data-tier Application

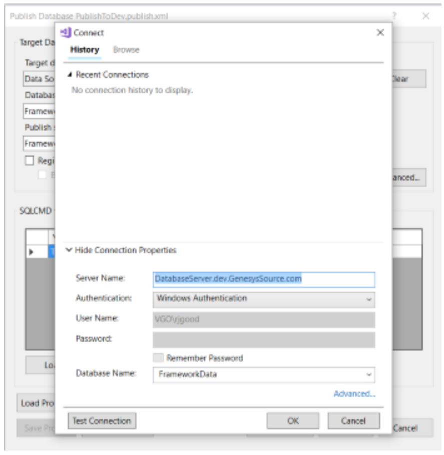
☐ Block publish when database has drifted from registered version

SQLCMD variables

Variable	Value
TargetEnvironment	Dev

1. Open the framework solution you wish to publish, for example: C:\Repos\Framework-for-Mvc.sln
2. Build *Framework.Database* project to ensure no errors
3. Open the Dev publish file \Publish\PublishToDev.publish.xml

② Set SQL Server



Recent Connections

No connection history to display.

Hide Connection Properties

Server Name: DatabaseServer.dev.GenesysSource.com

Authentication: Windows Authentication

User Name: VGO\rjgood

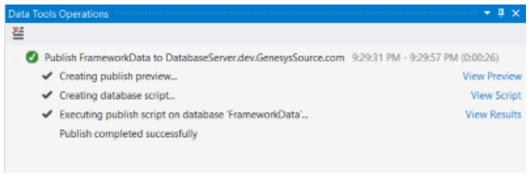
Password:

☐ Remember Password

Database Name: FrameworkData

4. Click *Edit* button
5. Change *Server Name* field to be the name of your SQL Server (i.e. Dev-Sql-16)
6. Click *OK*
7. Click the *Save Profile* button to save your changes

③ Publish



```
Publish FrameworkData to DatabaseServer.dev.GenesysSource.com   9:29:31 PM - 9:29:57 PM (0:00:26)
  ✓ Creating publish preview...
  ✓ Creating database script...
  ✓ Executing publish script on database 'FrameworkData'...
    Publish completed successfully
```

8. Click *Generate Script* to see the change script (no database changes will be applied.)
9. Click *Publish* to apply changes to the *FrameworkData* database

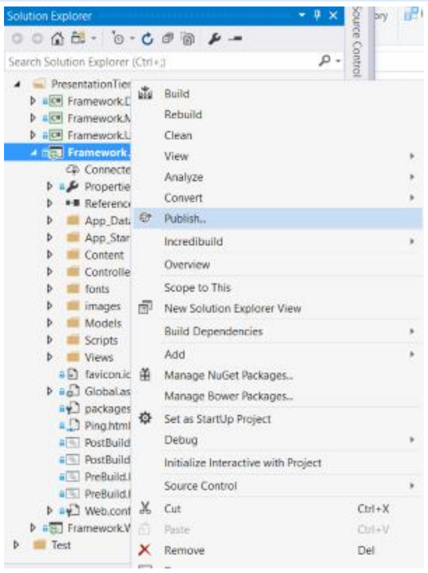
Your *Framework.Database* project is now in sync with your SQL Server

10: Publishing Framework.Database (SSDT) to SQL Server

Publishing Framework.WebApp (MVC) to an IIS Web Server

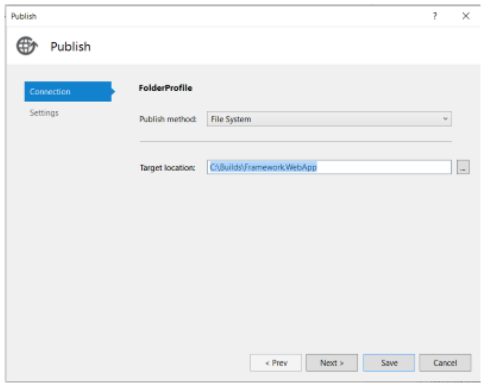
This procedure outlines how to publish the *Framework.WebApp* ASP.NET MVC project from Visual Studio to the IIS Web Server.

① Build



1. Open the MVC solution you wish to publish, for example: C:\Source\Framework-for-Mvc.sln
2. Build the solution to ensure no errors
3. Right-click the project and click *Publish*

② Set Target Path



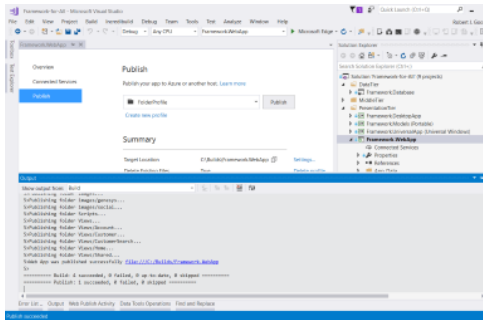
FolderProfile

Publish method: File System

Target location: C:\Builds\Framework.WebApp

4. Click the *Settings...* link in the *Publish* window
5. Change Target Location to be the path to your web project (default is local drive), for example:
\\Dev-Web-16\WebSites\Framework.WebApp
6. Click *Save*

③ Publish



7. Click *Publish* to publish the project to your development web server

The MVC project has now been published to your IIS Web Server.

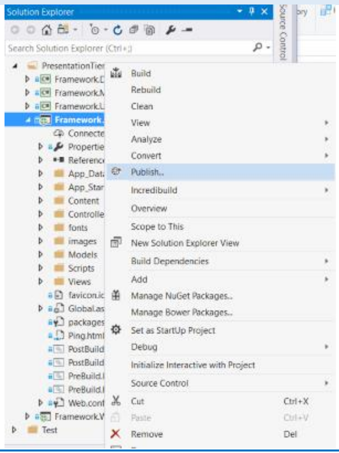
11: Publishing Framework.WebApp (MVC) to IIS Web Server

Publishing Framework.WebServices (Web API) to an IIS Web Server

This procedure outlines how to publish the *Framework.WebServices* ASP.NET Web API project from Visual Studio to the IIS Web Server.

①

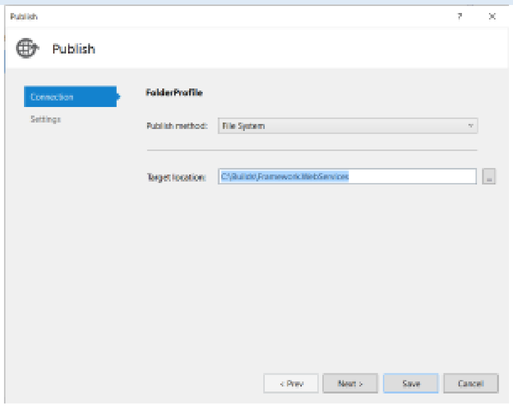
Build



1. Open the MVC solution you wish to publish, for example: C:\Repos\Framework-for-Mvc.sln
2. Build the solution to ensure no errors
3. Right-click the project and click *Publish*

②

Set Target Path



Publish method: File System

Target location: C:\Builds\Framework.WebServices

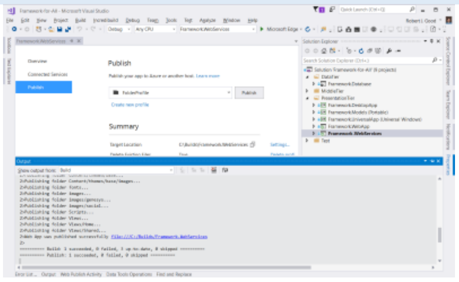
4. Click the *Settings…* link in the *Publish* window
5. Change Target Location to be the path to your web project (default is local drive), for example:

\\Dev-Web-16\WebSites\Framework.WebServices

6. Click *Save*

③

Publish



7. Click *Publish* to publish the project to your development web server

The Web API project has now been published to your IIS Web Server.

12: Publishing Framework.WebServices (Web API) to IIS Web Server

Tech and Code Aspects of the *Genesys Framework*

The Genesys Framework is a .NET Framework and Core stack containing C#, EF, SSDT, T-SQL, MVC, Web API, WPF, UWP, JavaScript, CSS and HTML. This section aims to explain some key tech aspects, to enable you to run and alter the projects with minimal learning curve.

Database Connections in App_Data

All projects communicate to SQL Server and SQL Express through the ***"DefaultConnection"*** connection string located in the following two files:

\App_Data\ConnectionStrings.Debug.json

\App_Data\ConnectionStrings.Release.json

Both include identical entries, but configured for different environments. For example: Dev SQL Server might be called DatabaseServer.dev.GenesysSource.com while Production is called DatabaseServer.prod.GenesysSource.com.

Here is an example of the **DefaultConnection** connection string:

```
"DefaultConnection": "data source=DatabaseServer.test.GenesysSource.com; initial
catalog=FrameworkData; user id=TestUser; password=57595709-9E9C-47EA-ABBF-4F3BAA1B0D37;
Multipleactiveresultsets=True; Application Name=GenesysFramework;"
```

13: DefaultConnection connection string

Web Service Connections in App_Data

All native-client project, such as *Framework.UniversalApp (UWP) and Framework.DesktopApp (WPF)* communicate to their Web API back-ends through the ***"MyWebService"*** application setting located in the following two files:

\App_Data\AppSettings.Debug.json

\App_Data\AppSettings.Release.json

Both include identical entries, but configured for different environments. For example: Dev might be called sampler.dev.GenesysSource.com while Production may be called sampler.GenesysSource.com.

Here is an example of the **MyWebService** app setting:

```
"MyWebService": "http://sampler.GenesysSource.com/Genesys-Framework-for-WebAPI/v1"
```

14: MyWebService application setting

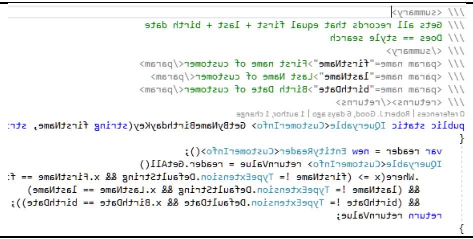
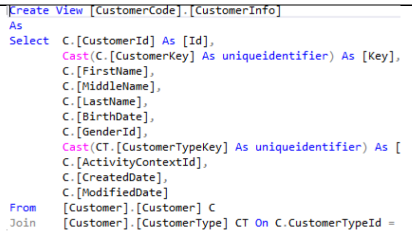
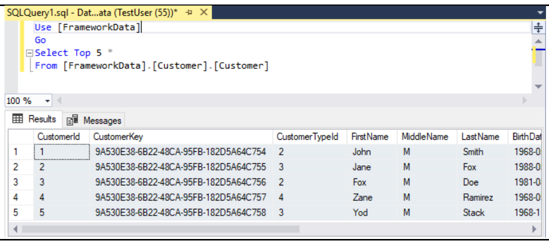
Framework.DataAccess pulls data through Framework.Database SQL Views

The *Framework.DataAccess* project employs EF Core for SQL Server data access. By default, *Framework.Database* is configured to:

- Select data through a (1) SQL View
- Insert, Update and Delete data through three (3) SQL Stored Procedures

Why? Because your code is now Loosely-coupled to your data through SQL Views.

SQL Views for Selects

Framework.DataAccess C# EF Code	*Framework.Database* SQL View	*FrameworkData* SQL Table
		

```
/// <summary>
/// Gets all records that equal first + last + birth date
/// Does == style search
/// </summary>
/// <param name="firstName">First name of customer</param>
/// <param name="lastName">Last Name of customer</param>
/// <param name="birthDate">Birth Date of customer</param>
/// <returns></returns>
0 references | Robert J. Good, 6 days ago | 1 author, 1 change
public static IQueryable<CustomerInfo> GetByNameBirthdayKey(string firstName, str
{
    var reader = new EntityReader<CustomerInfo>();
    IQueryable<CustomerInfo> returnValue = reader.GetAll()
      .Where(x => (firstName != TypeExtension.DefaultString && x.FirstName == f
        && (lastName != TypeExtension.DefaultString && x.LastName == lastName)
        && (birthDate != TypeExtension.DefaultDate && x.BirthDate == birthDate));
    return returnValue;
}
```

```
Create View [CustomerCode].[CustomerInfo]
As
Select  C.[CustomerId] As [Id],
        Cast(C.[CustomerKey] As uniqueidentifier) As [Key],
        C.[FirstName],
        C.[MiddleName],
        C.[LastName],
        C.[BirthDate],
        C.[GenderId],
        Cast(CT.[CustomerTypeKey] As uniqueidentifier) As [
        C.[ActivityContextId],
        C.[CreatedDate],
        C.[ModifiedDate]
From    [Customer].[Customer] C
Join    [Customer].[CustomerType] CT On C.CustomerTypeId =
```

SQLQuery1.sql - Dat...ata (TestUser (55))*

```
Use [FrameworkData]
Go
Select Top 5 *
From [FrameworkData].[Customer].[Customer]
```

100 %

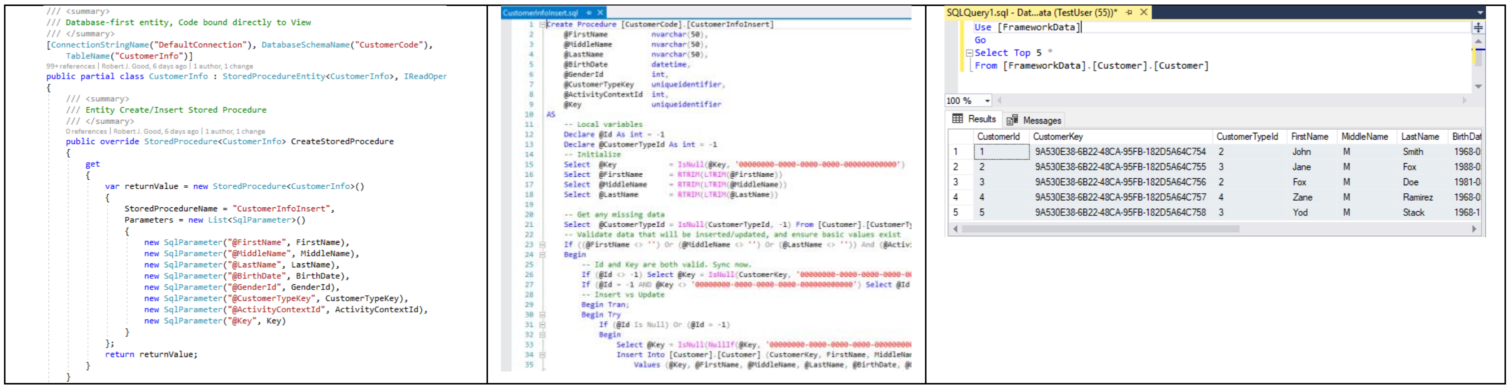
Results | Messages

	CustomerId	CustomerKey	CustomerTypeId	FirstName	MiddleName	LastName	BirthDat
1	1	9A530E38-6B22-48CA-95FB-182D5A64C754	2	John	M	Smith	1968-0
2	2	9A530E38-6B22-48CA-95FB-182D5A64C755	3	Jane	M	Fox	1988-0
3	3	9A530E38-6B22-48CA-95FB-182D5A64C756	2	Fox	M	Doe	1981-0
4	4	9A530E38-6B22-48CA-95FB-182D5A64C757	4	Zane	M	Ramirez	1968-0
5	5	9A530E38-6B22-48CA-95FB-182D5A64C758	3	Yod	M	Stack	1968-1

15: SQL Views for Selects

SQL Stored Procedures for Inserts, Updates and Deletes

Framework.DataAccess C# EF Code	*Framework.Database* SQL SP	*FrameworkData* SQL Table



```
/// <summary>
/// Database-first entity, Code bound directly to View
/// </summary>
[ConnectionStringName("DefaultConnection"), DatabaseSchemaName("CustomerCode"),
    TableName("CustomerInfo")]
99+ references | Robert J. Good, 6 days ago | 1 author, 1 change
public partial class CustomerInfo : StoredProcedureEntity<CustomerInfo>, IReadOper
{
    /// <summary>
    /// Entity Create/Insert Stored Procedure
    /// </summary>
    0 references | Robert J. Good, 6 days ago | 1 author, 1 change
    public override StoredProcedure<CustomerInfo> CreateStoredProcedure
    {
        get
        {
            var returnValue = new StoredProcedure<CustomerInfo>()
            {
                StoredProcedureName = "CustomerInfoInsert",
                Parameters = new List<SqlParameter>()
                {
                    new SqlParameter("@FirstName", FirstName),
                    new SqlParameter("@MiddleName", MiddleName),
                    new SqlParameter("@LastName", LastName),
                    new SqlParameter("@BirthDate", BirthDate),
                    new SqlParameter("@GenderId", GenderId),
                    new SqlParameter("@CustomerTypeKey", CustomerTypeKey),
                    new SqlParameter("@ActivityContextId", ActivityContextId),
                    new SqlParameter("@Key", Key)
                }
            };
            return returnValue;
        }
    }
```

CustomerInfoInsert.sql

```
Create Procedure [CustomerCode].[CustomerInfoInsert]
    @FirstName          nvarchar(50),
    @MiddleName         nvarchar(50),
    @LastName           nvarchar(50),
    @BirthDate          datetime,
    @GenderId           int,
    @CustomerTypeKey    uniqueidentifier,
    @ActivityContextId  int,
    @Key                uniqueidentifier
AS
    -- Local variables
    Declare @Id As int = -1
    Declare @CustomerTypeId As int = -1
    -- Initialize
    Select  @Key            = IsNull(@Key, '00000000-0000-0000-0000-000000000000')
    Select  @FirstName      = RTRIM(LTRIM(@FirstName))
    Select  @MiddleName     = RTRIM(LTRIM(@MiddleName))
    Select  @LastName       = RTRIM(LTRIM(@LastName))

    -- Get any missing data
    Select  @CustomerTypeId = IsNull(CustomerTypeId, -1) From [Customer].[CustomerTy
    -- Validate data that will be inserted/updated, and ensure basic values exist
    If ((@FirstName <> '') Or (@MiddleName <> '') Or (@LastName <> '')) And (@Activ
    Begin
        -- Id and Key are both valid. Sync now.
        If (@Id <> -1) Select @Key = IsNull(CustomerKey, '00000000-0000-0000-0000-00
        If (@Id = -1 AND @Key <> '00000000-0000-0000-0000-000000000000') Select @Id
        -- Insert vs Update
        Begin Tran;
        Begin Try
            If (@Id Is Null) Or (@Id = -1)
            Begin
                Select @Key = IsNull(NullIf(@Key, '00000000-0000-0000-0000-00000000
                Insert Into [Customer].[Customer] (CustomerKey, FirstName, MiddleNa
                    Values (@Key, @FirstName, @MiddleName, @LastName, @BirthDate, @
```

SQLQuery1.sql - Dat...ata (TestUser (55))*

```
Use [FrameworkData]
Go
Select Top 5 *
From [FrameworkData].[Customer].[Customer]
```

100 %

Results | Messages

	CustomerId	CustomerKey	CustomerTypeId	FirstName	MiddleName	LastName	BirthDat
1	1	9A530E38-6B22-48CA-95FB-182D5A64C754	2	John	M	Smith	1968-0
2	2	9A530E38-6B22-48CA-95FB-182D5A64C755	3	Jane	M	Fox	1988-0
3	3	9A530E38-6B22-48CA-95FB-182D5A64C756	2	Fox	M	Doe	1981-0
4	4	9A530E38-6B22-48CA-95FB-182D5A64C757	4	Zane	M	Ramirez	1968-0
5	5	9A530E38-6B22-48CA-95FB-182D5A64C758	3	Yod	M	Stack	1968-1

16: SQL Stored Procedures for Inserts, Updates and Deletes

Why the *Genesys Framework*?

The *Genesys Framework* was built out of frustration with the Copy-paste Anti-pattern in our daily software engineering lives. Boomerang bugs, bloated classes, inconsistent coding standards made development slow and tedious. Most software engineers know of good practices, some have even built reusable stacks…but inevitably the project would not be approved or completed.

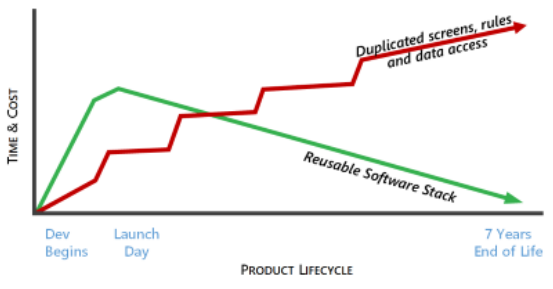
We set out to make code reuse fast, easy and with a minimal learning curve.

Why build reusable code?

Code reuse is an important theme in many of todays accepted software practices, such as N-tier and Object-oriented programming (OOP.)

Typically, reusable software stacks and services have low technical debt and are cheaper to maintain over time. Reusable code "settles" over time and costs decrease. Your return on investment (RoI) is greater with reusable software stacks

Conversely, the code duplication method tends to cost more over time, with high technical debt in the form of maintenance time and costs spiking per each duplicated item. Your costs go up over time, until the software is rewritten or retired.



The Genesys Framework offers n-tier, reusable business objects, with a low learning curve. Reusability without the cost of doing it yourself, and without the uncertainty of an untested new code base.

Why code full-stack, cross-platform business objects?

Microsoft .NET classes have a unique characteristic…they can run almost anywhere on any popular platform and run in any software tier. This allows a .NET entity class, like a Customer entity, to enjoy a 100% strongly-typed stack and consistency in properties and validation rules…in web sites, web services, native apps, CLR stored procedures and in class libraries.

With cross-platform full-stack entity objects, spelling errors and type errors show immediately as a compile error…in a stored procedure, in a data access C# file, in a MVC controller…everywhere that entity is used. Typing is maintained through the stack:

Any SQL Data -> Framework.Database -> Framework.DataAccess -> Framework.Models -> Any .NET App

Genesys Framework takes advantage of run-anywhere to enable any business object to run in Web, Services, Desktop and Mobile.

Take the Customer entity as an example:

- *CustomerInfo.cs:* Heavy Data Access Object (DAO) based on Entity Framework database-first. Supports CRUD-to-SQL methods of *Create*(), *Read*(*Expression*), *Update*(), *Delete*().
- *CustomerModel.cs*: Lightweight screen and transport models. This class is cross-platform and runs in MVC, Web API, UWP, WPF, Xamarin iOS, Xamarin Android, CLR Stored Procedures.
- *CrudViewModel<CustomerModel>*: MVVM ViewModel with CRUD-to-Services methods such as *CreateAsync*(), *ReadAsync*(*Expression*), *UpdateAsync*() and *DeleteAsync*().
- *customer.Serialize()*: JSON string is returned from any class that inherits *CrudEntity* or *ModelEntity*. This JSON can be controller generated and used by client-side web applications.

Getting Help

Have a question? Have a problem? Contact us anytime…

Contact Genesys Source	Help and Guidance	On Social
GENESYSSOURCE 22431 Antonio, Suite B160-843 Rancho Santa Margarita, CA 92688 +1 949.544.1900 www.genesyssource.com *help@genesyssource.com*	[Docs] docs.genesyssource.com [FAQs] Frequently Asked Questions [Q+A] Question and Answer [+/-] Report an Issue [Zip] Download Genesys Framework [Azure] Try Cloud Dev Environment Free	http://twitter.com/GenesysSource http://facebook.com/GenesysSource http://GenesysSource.com/blog http://GenesysSource.com/news/rss/1