

Quick Start - ASP.NET Web Form Pages

Table of Contents

Chapter 3	3-1
Quick Start - ASP.NET Web Form Pages.....	3-1
Quick Start for ASP.NET Web Form Applications.....	3-1
Add New Project	3-2
Generate CRUD Classes for Tables	3-4
Table Information Screen.....	3-4
Generate Table Data Classes	3-5
Test Generated Classes and Web Form Pages	3-7
Chapter Index.....	3-13

Quick Start for ASP.NET Web Form Applications

This document will describe how to start using Haystack for Code Generation to quickly generate a working **ASP.NET Web Form Application**.

From your start menu locate the Haystack folder and click on the **Haystack Code Generator** icon. This will start the Haystack Code Generator for .NET (Figure 1).

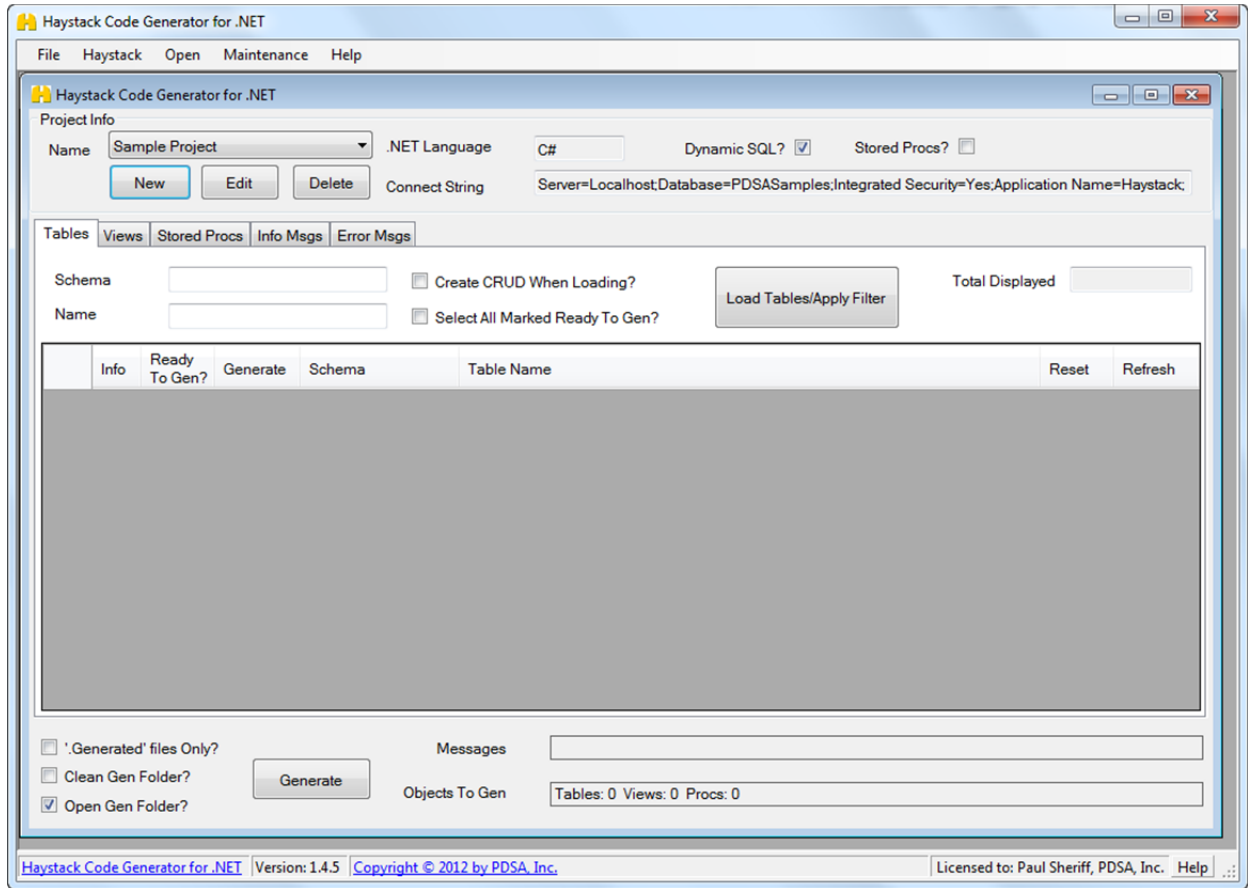


Figure 1: Haystack Main Screen

After starting Haystack you will need to add a new Project. A project is used to generate all objects from a SQL Server or Oracle database.

Add New Project

Click on the “New” button beneath the Project Info Name Combo Box (Figure 2) to add a new project to Haystack.

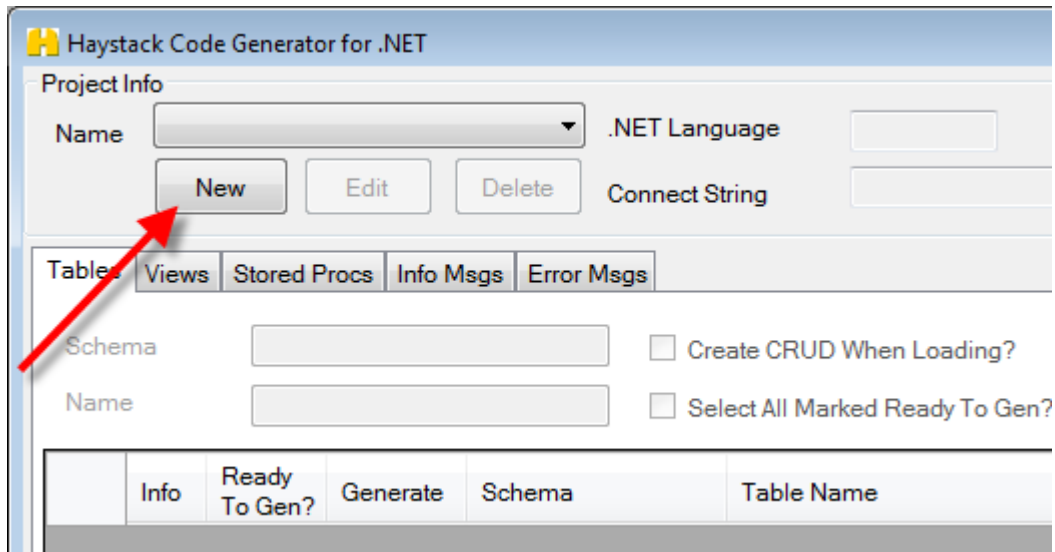


Figure 2: Click "New" to create a new Project

You will now be presented with the **Project Information** screen. On this screen is where you will create the project that points to a database.

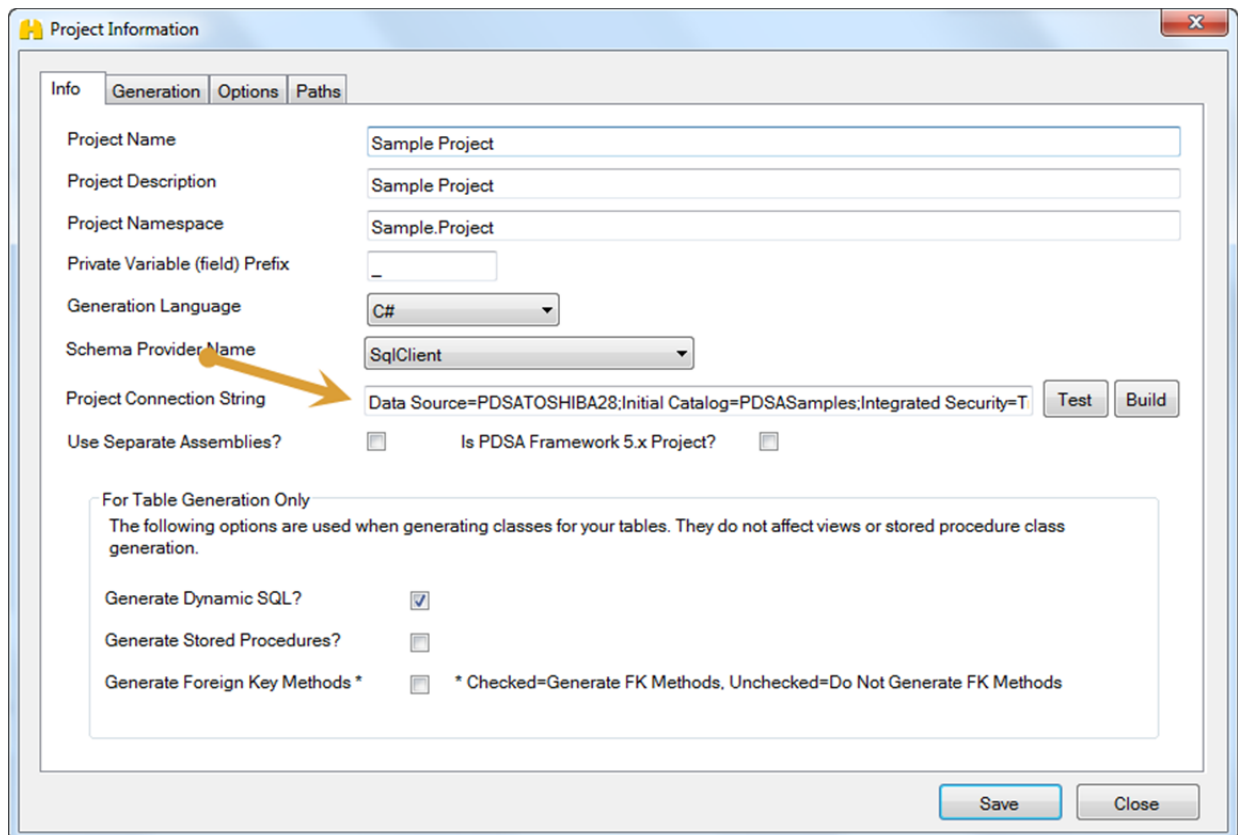


Figure 3: Project Information (Info Tab)

All you need to create a project is on this first tab (Figure 3). For this quick start, just fill in a connection string to one of your databases, or if you installed the

PDSASamples database when you installed Haystack, just leave this existing connection string.

Click the **Save** button to save this new project into the Haystack database.

Generate CRUD Classes for Tables

After you have created a project and set the connection string to a valid server and database/catalog, you are now ready to read in the list of tables in the catalog or schema. Click on the **Load Tables/Apply Filter** button (Figure 4) to load in all the tables.

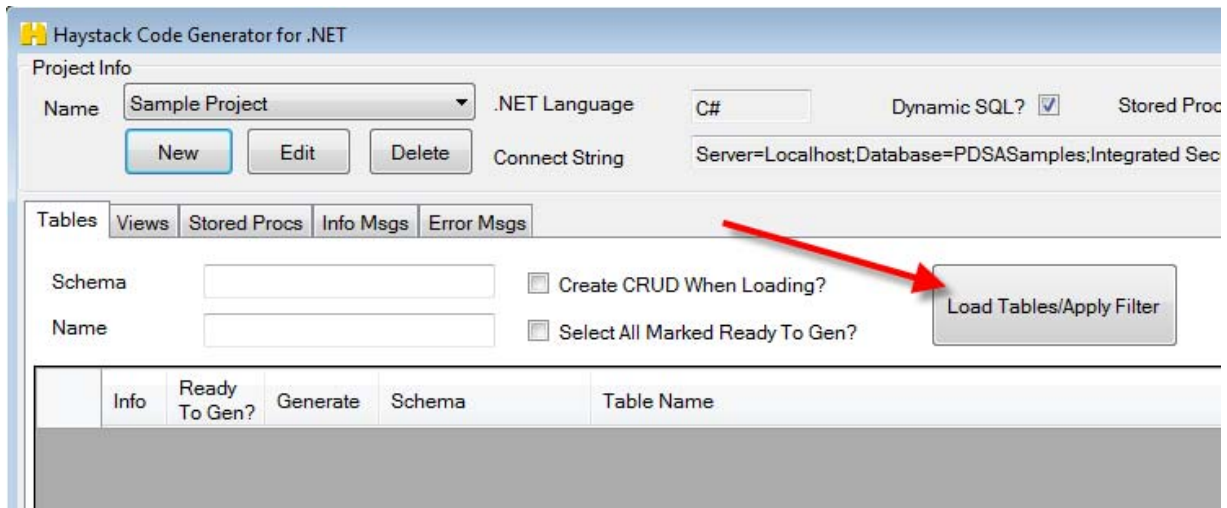


Figure 4: Load Tables

After clicking the **Load Tables/Apply Filter** button the grid on the lower part of the screen will be filled in with the names of the tables that match the filter.

At this point you need to click on the **Info** button to the left of a table to load all of the columns for the table and display the Table Information Screen (Figure 5). You must open the table to load all columns in order to generate classes for a table.

Table Information Screen

On this screen you may modify any of the SELECT, WHERE, ORDER BY, INSERT, UPDATE or DELETE statements. You can read more about this

screen in another chapter. But to start all you have to do right now is to click on the **Save Table Info** button. This will save all the columns for the table and all the SELECT, WHERE, etc. statements into the Haystack database so that it can now be generated.

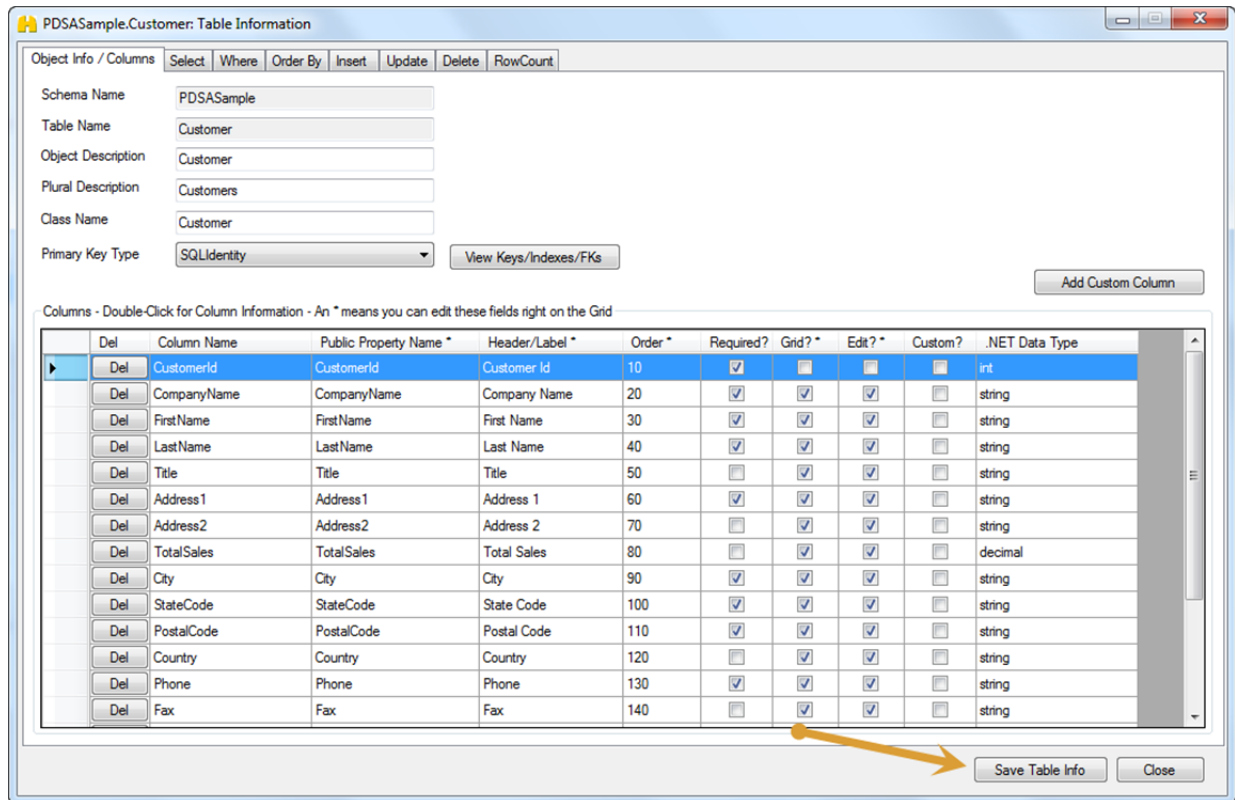


Figure 5: Table Information Screen

Generate Table Data Classes

After saving the table information you will notice that the **Generate** check box is now checked (Figure 6). You may now click on the **Generate** button to generate the CRUD classes for the table selected.

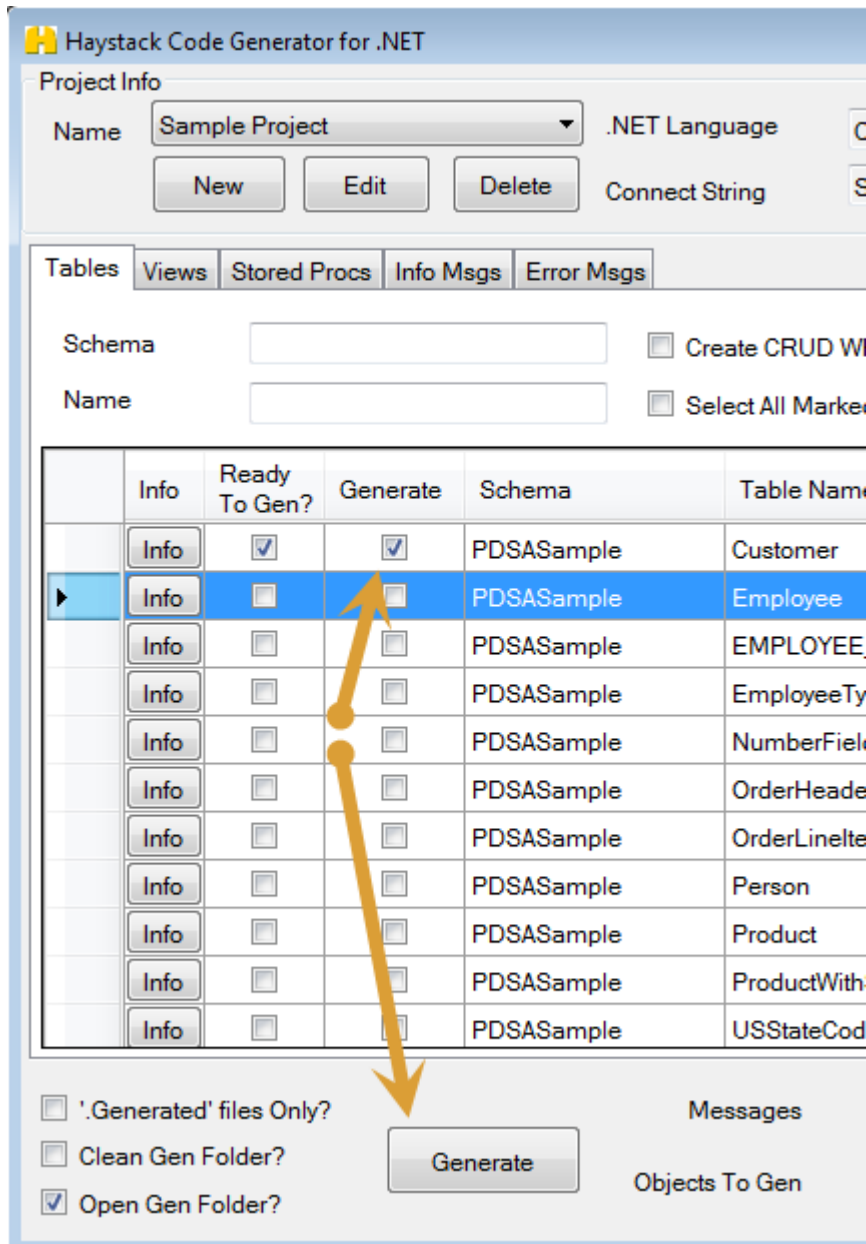


Figure 6: Generate Code

After the generation completes, a Windows Explorer window will open where all of the code was generated.

Test Generated Classes and Web Form Pages

Once the classes and Web Form Pages are generated you are ready to put them into a project and try them out.

Open Visual Studio 2010 (or later).

Click on **File | New | Project** from the menu

Choose the **Web** Templates and click on the **ASP.NET Web Application** template.

Fill in the **Name:** field with **Sample.Project** (or whatever namespace you used in the Project Information screen in Haystack).

<p>NOTE: If you are using Visual Basic, you need to name the WPF application the same name you used in the Namespace in the Haystack project you setup. This is normally Sample.Project.</p>
--

Browse to where you wish to save this project.

Click the **OK** button

In the Windows Explorer window where the code was generated, copy the entire contents of the **ASP.NET-WebForms-Pages** folder into the root of your new web application project (Figure 7).

When prompted to overwrite the Web.Config file or any other folders, choose **Yes**.

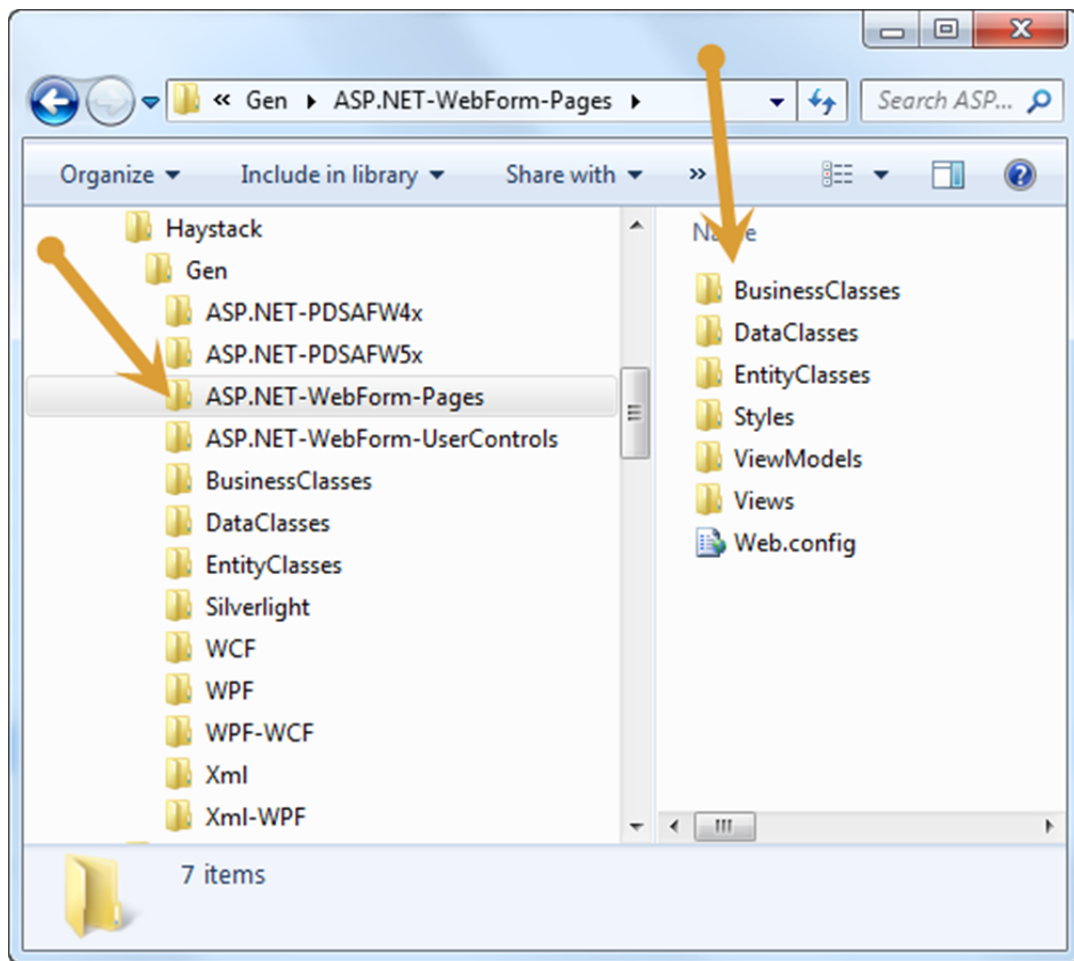


Figure 7: Copy all folders from the ASP.NET-WebForm-Pages folder into your project

Your Solution Explorer window in your Web application should now look like the Figure 8:

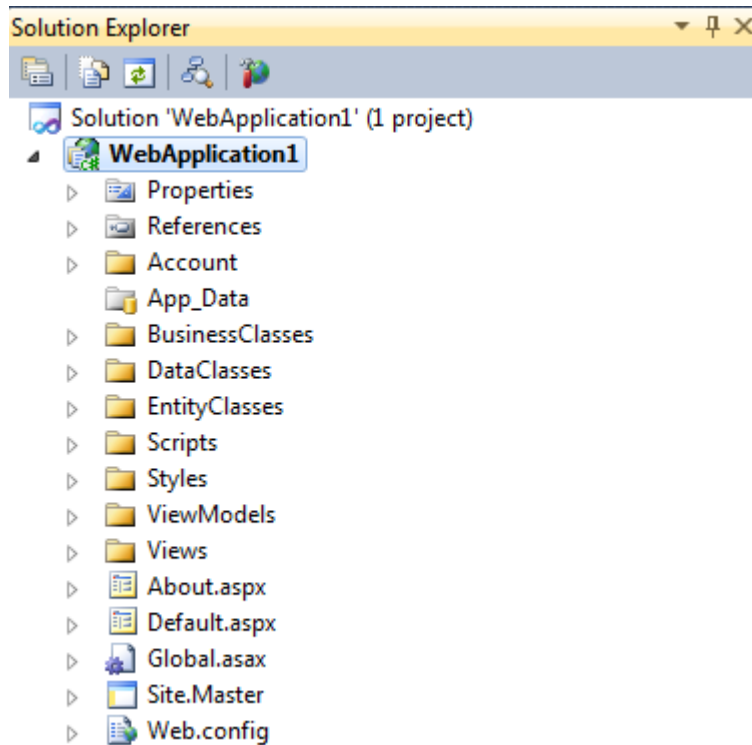


Figure 8: Your Visual Studio project after adding generated code

You now need to add some resources to your ASP.NET web application in order for the generated code to work.

From Haystack, click the **Open | Install Folder for Resources** menu as shown in Figure 9.

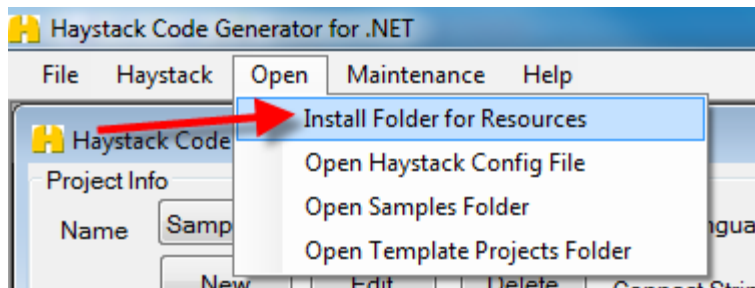


Figure 9: Your Visual Studio project after adding generated code

Locate the folder named `_Resources-For-ASP.NET-WebForms` (Figure 10).

Copy in the **Images** folder into the root of your web project

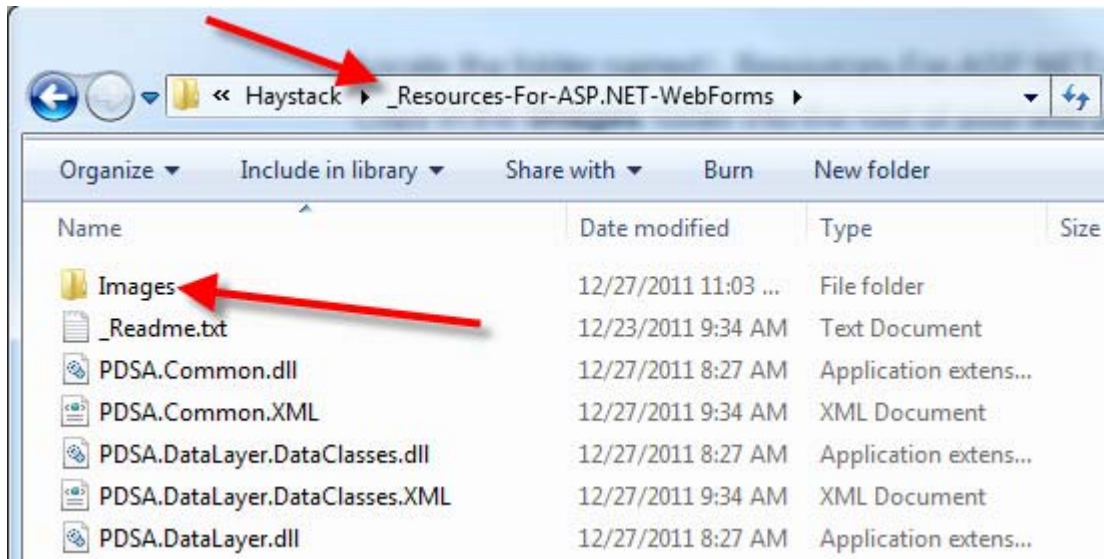


Figure 10: Locate the resources to add to your Web application.

Now back in your Visual Studio web application right mouse click and choose **Add Reference...** from the context menu.

Click the **Browse** tab.

Navigate to the folder where you installed Haystack which is typically **C:\Program Files\Haystack** or **C:\Program Files (x86)\Haystack** and under the folder named **_Resources-For-ASP.NET-WebForms** select all of the DLLs that you find there and click the OK button as shown in Figure 11.

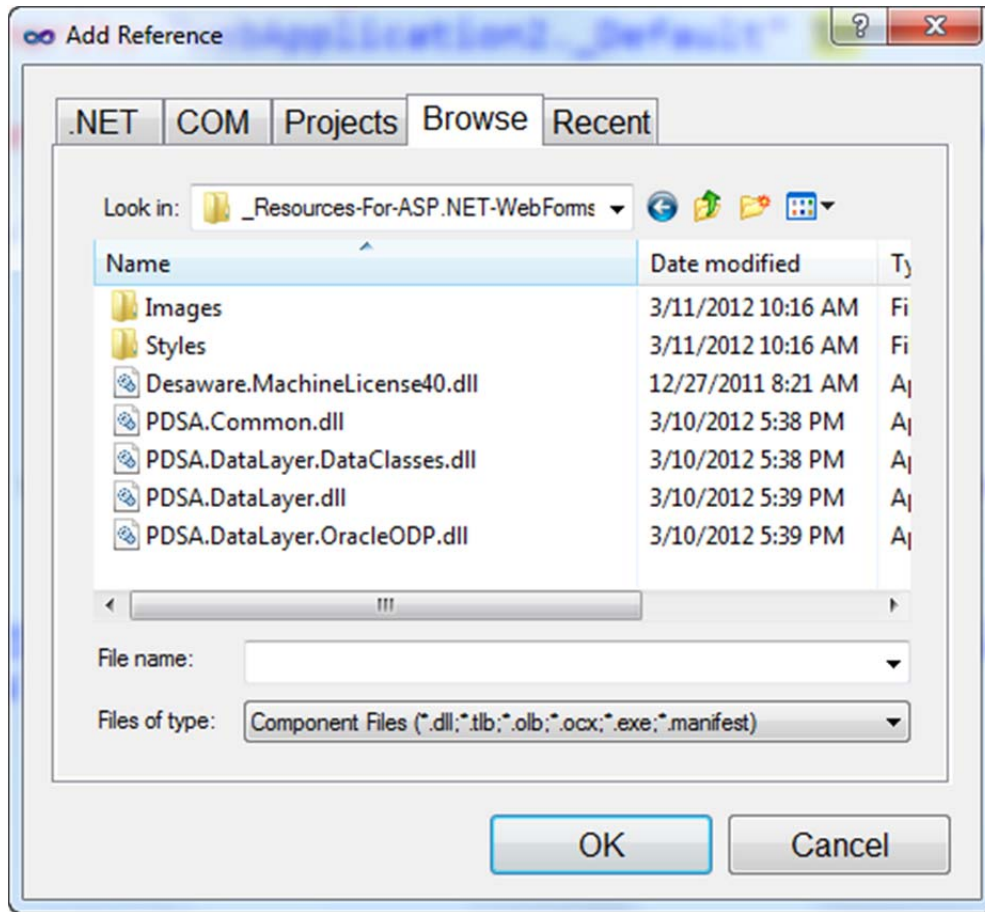


Figure 11: Reference the PDSA DLLs

Now within the **Views** folder in your Visual Studio project, find one of the generated views, right mouse click on and select **Set As Start Page**. Press F5 to run the application and your new view should appear.

NOTE: If you receive an error about a Namespace not being found, this is normally because the Namespace you created in the Project screen in Haystack does not match the name of the application you created. Simply perform a search and replace on this name to fix this.

Congratulations! You created a complete List, Search, Add, Edit and Delete page in just a matter of minutes.

Summary

In this chapter you learned how to quickly generate CRUD classes and put those classes into an ASP.NET project and test out the generated classes and Web Form Pages.

<p>NOTE: The PDSA DLLs ONLY run in VS.NET when you are using the DEMO version of Haystack. In order to make them run from a website you need to purchase the full version of Haystack.</p>

Chapter Index

A

Add New Project, 3-2

G

Generate Button, 3-5

Generate CRUD Classes for Tables, 3-4

Generate Table Data Classes, 3-5

I

Info Button, 3-4

L

Load Tables/Apply Filter Button, 3-4

Q

Quick Start for ASP.NET Web Form
Applications, 3-1

S

Save Table Info Button, 3-5

T

Table Information Screen, 3-4

Test Generated Classes and Web Form
Pages, 3-7

Tutorial for ASP.NET Web Form
Applications, 3-1