



By: PDSA, Inc.

User Manual for Data Access Layer

Version 2.x

August 2016

Published By: PDSA, Inc.

Copyright © 2010-2016, PDSA, Inc.
All Rights Reserved Worldwide

Written By: Paul D. Sheriff
Technical Editors: Paul D. Sheriff

Every effort has been made to supply complete and accurate information. However, PDSA, Inc. assumes no responsibility for its use, nor for any infringement of the intellectual property rights of third parties which would result from such use.

Copyright © 2010-2016 by PDSA, Inc.

All rights reserved worldwide. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

Printed in the United States of America

Table of Contents

Table of Contents	3
Introduction.....	11
Goals of Haystack	12
System Requirements	13
Supported Databases.....	14
Your Knowledge	14
Installation of Haystack	15
Using Haystack Generated Code.....	15
Technical Support	15
Custom Templates.....	16
PDSA DLLs.....	16
Fee based support.....	16
Payment Options	16
Support Options.....	17
Support Policy.....	17
License Agreement	17
Acknowledgements	18
Chapter 1	1-1
Haystack Overview	1-1
Haystack Overview.....	1-2
Philosophy.....	1-2
Using Haystack Generated Code.....	1-3
Goals of Haystack	1-4
What Haystack Can Create.....	1-4
Business/Manager Class.....	1-4
Entity Class.....	1-4
Search Class	1-5
Validation Class	1-5
View Model Generation	1-5
Stored Procedures.....	1-5
WPF CRUD User Control	1-5
Web Forms CRUD Page	1-5
MVC CRUD Pages	1-6
Web API Services Generation.....	1-6
WCF Services Generation.....	1-6

Why Use Data Access Classes.....	1-6
A Haystack Project.....	1-7
Summary.....	1-8
Chapter Index.....	1-8
Chapter 2.....	2-1
Installation.....	2-1
Installation Procedures.....	2-2
Windows Firewall.....	2-3
Error 1925 or Problems Installing Haystack.....	2-3
Folders Haystack Creates/Accesses.....	2-4
My Documents Folder.....	2-5
C:\Program Data Folder.....	2-6
Change Location of Haystack.exe.config File (if needed).....	2-7
Change Generation Folder (if needed).....	2-7
PDSA Product Activation Screen.....	2-8
Internet Access Required for Activation or Trial.....	2-9
Limitations of Trial Version.....	2-9
Haystack Tips & Tricks Screen.....	2-10
Configure Haystack Screen Step 1.....	2-10
Configure Haystack Screen Step 2.....	2-11
Configure Haystack Screen Step 3.....	2-13
What to do if you don't have DBA rights.....	2-14
Configure Haystack Screen Step 4.....	2-14
Haystack Installation Folder.....	2-16
My Documents Folder.....	2-17
Sample Programs.....	2-17
Registering Haystack.....	2-18
Uninstalling Haystack.....	2-19
Summary.....	2-20
Chapter Index.....	2-20
Chapter 3.....	3-1
Quick Start - MVC.....	3-1
Using Haystack Generated Code in MVC.....	3-2
Quick Start for MVC Applications.....	3-2
Add New Haystack Project for MVC.....	3-3
Generate CRUD Classes for Tables.....	3-5

Table Information Screen	3-6
Generate Table Data Classes	3-7
Create New MVC Project	3-9
Copy Generated Files	3-10
Include Files in Project (VS Bug) - MVC.....	3-13
Add DLLs for MVC	3-14
Fix up the Web.Config File for MVC	3-15
Modify the Global.asax in MVC	3-16
Modify the Home\Index Page in MVC	3-17
Run the MVC Application	3-17
Summary	3-18
Chapter Index.....	3-19
Chapter 4	4-1
Quick Start - WPF	4-1
Using Haystack Generated Code in WPF	4-2
Quick Start for WPF Applications	4-2
Add New Haystack Project for WPF.....	4-3
Generate CRUD Classes for Tables - WPF	4-5
Table Information Screen for WPF	4-6
Generate Table Data Classes for WPF	4-7
Create New WPF Application	4-9
Copy Generated Files for WPF	4-10
Include Files in WPF Project (VS Bug).....	4-11
Add DLLs to WPF Project	4-13
Fix up the App.Config File in WPF	4-15
Modify the App.xaml.cs in WPF.....	4-15
Add WPF User Control to Main Window	4-16
Summary	4-19
Chapter Index.....	4-20
Chapter 5	5-1
Quick Start - Web API	5-1
Using Haystack Generated Code in Web API.....	5-2
Quick Start for Web API Generation.....	5-2
Add New Haystack Project for Web API.....	5-3

- Generate CRUD Classes for Tables - Web API..... 5-5
- Table Information Screen for Web API..... 5-6
- Generate Table Data Classes for Web API..... 5-7
- Create New Web API Project..... 5-9
- Copy Generated Files for Web API..... 5-10
- Include Web API Files in Project (VS Bug) 5-13
- Add DLLs for Web API 5-14
- Fix up the Web.Config File for Web API..... 5-15
- Modify the Global.asax for Web API 5-15
- Run the Web API Project 5-16
- Summary 5-18
- Chapter Index..... 5-19
- Chapter 6 6-1
- Quick Start - Angular..... 6-1
 - Using Haystack Generated Code for Angular 1 6-2
 - Quick Start for Angular Generation 6-2
 - Add New Haystack Project for Angular 1 6-3
 - Generate CRUD Classes for Tables – Angular 1..... 6-5
 - Table Information Screen for Angular 1 6-6
 - Generate Table Data Classes for Angular 1 6-7
 - Create New Angular 1 Project..... 6-9
 - Copy Generated Files for Angular 1..... 6-10
 - Include Angular Files in Project (VS Bug) 6-13
 - Add DLLs to Angular 1 Project..... 6-14
 - Fix up the Web.Config File for Angular 1 6-15
 - Modify the Global.asax for Angular 1 6-15
 - Run the Angular 1Project 6-16
 - Summary 6-18
 - Chapter Index..... 6-19
- Chapter 7 7-1
- Quick Start - jQuery..... 7-1
 - Using Haystack Generated Code for jQuery 7-2
 - Quick Start for jQuery Generation 7-2
 - Add New Haystack Project for jQuery 7-3
 - Generate CRUD Classes for Tables - jQuery 7-5

Table Information Screen for jQuery	7-6
Generate Table Data Classes for jQuery	7-7
Create New Web API Project	7-9
Copy Generated Files for jQuery.....	7-10
Include jQuery Files in Project (VS Bug)	7-13
Add DLLs for jQuery	7-14
Fix up the Web.Config File for jQuery	7-15
Modify the Global.asax for jQuery	7-15
Prepare to Run the HTML Page with jQuery	7-16
Modify [ClassName].html File	7-17
Summary	7-20
Chapter Index.....	7-21
Chapter 8	8-1
PDSA Data Access Layer Reference.....	8-1
Two PDSA Data Layers.....	8-3
Initialize Data Access Layer for Use in Your Projects	8-3
Add Required DLLs	8-3
Add Connection String.....	8-3
Initialize the Data Access Layer.....	8-4
Overview of Data Access Layer Classes.....	8-4
Files Generated	8-7
Entity Class.....	8-8
Manager Class.....	8-8
Search Class	8-8
Validator Class.....	8-8
Entity Class	8-8
Properties in Entity Class.....	8-9
Methods in Entity Class	8-9
Manager Class	8-9
Properties in Manager Class	8-10
Methods in Manager Class	8-10
Search Class	8-11
Validator Class	8-12
Properties in Validator Class	8-12
Methods in Validator Class	8-12
Summary	8-14
Chapter Index.....	8-14

Chapter 9	9-1
PDSA Data Access Layer Usage	9-1
Using Haystack Generated Code	9-3
Data Access Layer Samples	9-3
Tables	9-3
View	9-4
Stored Procedures	9-4
Select / Search / Sort	9-4
Get All Records	9-4
Get Subset of Columns	9-5
Add a WHERE Clause	9-6
Add Additional WHERE Clauses	9-7
Load a Single Record	9-9
Scalar Values	9-10
Built-In RowCount() Method	9-10
Row Count Using a WHERE Clause	9-10
Row Count Using IS NULL	9-11
Total Sales Using SUM()	9-12
Transform Data	9-13
Get Customers as JSON	9-13
Get Customers as XML	9-14
Get Customers as Excel String	9-14
Data Validation	9-15
Built-In Validation (Data Annotations)	9-15
Validate and Throw Exception	9-15
Custom Validation	9-16
Using Data Modification Action Property	9-17
Modify Data	9-18
Insert Data	9-18
Update Data	9-18
Delete Data	9-19
Bulk Updates and Delete	9-20
Custom Update	9-20
Update TotalSales Where Null	9-21
Delete Where TotalSales Equals a Value	9-22
Transactions	9-23
Simple Transaction	9-23
Transaction with Exception	9-26
Order Header/Detail Transaction	9-26

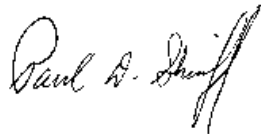
Stored Procedure Usage	9-29
Get Products No Input/Output Parameters	9-29
Get Products Input Parameters	9-29
Get Products Input/Output Parameters	9-30
Insert a Product	9-30
Product Insert with Output Parameter	9-31
View Usage	9-32
Summary	9-33
Chapter Index.....	9-33
Chapter 10	10-1
Relationships Using the Data Access Layer	10-1
Create Child Table References	10-2
Relationships in PDSASamples Database	10-2
Generated Relationship Code	10-3
One to Many Relationships in View Model.....	10-5
WPF Sample.....	10-6
MVC Sample.....	10-6
Load Child Tables using View Model	10-6
MVC Sample.....	10-6
Load Child Tables Using Property in Entity Class	10-9
Modify Generated Code.....	10-11
WPF Sample.....	10-12
Combo Box on Edit Page for Foreign Key Tables.....	10-15
WPF Sample.....	10-15
Load Child Tables in WPF Using Manager Class	10-16
Summary	10-21
Chapter Index.....	10-21
Chapter 11	11-1
Haystack Code Generator Reference	11-1
Code Generation	11-2
Project Information Screen	11-3
Project Information Screen (Generation Tab).....	11-4
Project Information (Options Tab).....	11-5
Project Information (Paths Tab).....	11-7
Step-by-Step Scenarios	11-9
Table Data Classes using Dynamic SQL	11-9
Create Dynamic SQL Project.....	11-9

Load Tables.....	11-10
Create CRUD when Loading?	11-11
Select all Marked Ready to Gen?	11-11
Read Columns.....	11-11
Generate Code for Dynamic SQL Classes.....	11-12
Table Data Classes using Stored Procedures	11-15
Create Stored Procedure Project	11-15
Load Tables to Generate Stored Procs for.....	11-16
Read Columns and Generate Default CRUD Stored Procedures	11-17
Generate Code for Table Stored Procedure Classes.....	11-18
View Data Classes	11-20
Stored Procedure Data Classes.....	11-21
Stored Procedures that Execute Data Modification Statements	11-21
Entity Class to Map To	11-22
Stored Procedures that Return Data.....	11-23
Summary	11-25
Chapter Index.....	11-25
Chapter 12	12-1
Deploying Haystack Applications	12-1
Runtime License Creator Utility	12-2
Get the Assembly Name.....	12-3
Get the Assembly Guid.....	12-3
Creating a Guid	12-4
Create License File.....	12-6
Add License File to Your Project.....	12-6
WPF Applications	12-7
MVC Applications	12-7
Web API/WCF Applications.....	12-9
Summary	12-10
Chapter Index.....	12-10

Introduction

Thank you for purchasing Haystack Code Generator for .NET. This tool will help you be more productive when creating your Service Oriented (N-Tier) C# and Angular applications. Like any good product it is never done. We are continually enhancing this product. You can be assured that you will receive all of the upgrades for free for your first year. After that there is a nominal fee that is explained under Support Options in this document. In addition, we would like to ask that if you make any additions, that you send those along to us so we can include them in the next upgrade. We hope you find this tool as useful as we do, and look forward to hearing from you.

Sincerely,

A handwritten signature in black ink that reads "Paul D. Sheriff". The signature is written in a cursive style with a prominent initial "P".

Paul D. Sheriff
President, PDSA, Inc.

Goals of Haystack

By using this product you will create N-Tier (Service Oriented) applications in a fraction of the time it would normally take without using any code generation tools. Look at all the features of this very useful tool:

- Generate code for C# applications
- Generate classes for Tables
 - Generate Select, Search, Insert, Update and Delete statements
 - Use Dynamic SQL and/or Stored Procedures for CRUD logic
 - Generate the stored procedures for CRUD logic for a table
 - Generate Data Annotations for business rules
 - Detect foreign key relationships and generate methods to allow you to select data from these relationships.
- Generate classes for Views
 - Read Only Class for calling the View.
- Generate classes for Stored Procedures
 - Data modification stored procedures
 - Stored procedures that return data but have no input parameters
 - Stored procedures that return data with input parameters
 - Stored procedures that return data with input and output parameters
 - Maps all input and output parameters to properties in class
 - Maps all columns returned to properties in class
 - Automatically fills in the value of output parameters to properties in class
- Generate WPF Add/Edit/Delete User Controls
 - Creates a detail user control that generates a TextBlock and TextBox for each field in the table.
 - Automatically display validation rule errors on bad input
 - If there is a foreign key relationship in the table, then a combo box is generated for the foreign key.

- Create a ListView user control for displaying records.
 - Bind the ListView to the detail user control
 - A View Model class is generated that is bound to the WPF user control.
 - All data access is called from the ViewModel class.
- Generate Web API Service Classes
 - Get, Put, Post and Delete Methods
- Generate WCF Service Classes
 - Creates Interfaces for Services.
 - Creates Services to call Data Classes
 - Create Response Objects that are returned from services
 - Creates .SVC files to call the Services
- Generate Angular Add/Edit/Delete Pages
 - A bootstrap, Angular CRUD page is generated
 - Communicates via generated Web API methods
 - HTML 5 and Bootstrap
- Generate HTML/jQuery Add/Edit/Delete Pages
 - A bootstrap CRUD page is generated
 - Communicates via generated Web API methods
 - HTML 5 and Bootstrap
- Generate MVC Add/Edit/Delete Pages
 - Generates Controllers for each table
 - Generates View Model that is called from the Controller for all data access and business rule validation
 - Generates Create, Delete, Details, Edit and Index pages

System Requirements

Below are the assumptions about your computing environment.

- You are running Visual Studio 2015 or later
- You are using SQL Server 2008/2012 or later for storing the Haystack meta-data.

- You are using SQL Server 2008 or later to generate classes from.
- You are running Windows 7 or later.
- To run our trial version of Haystack, you must have an internet connection.

Supported Databases

You may generate data classes for tables, views, and stored procedures from the following databases:

- SQL Server 2008 and higher

Your Knowledge

Haystack helps you create N-Tier data classes and standard add, edit, delete forms. The use of these classes does assume a fairly good knowledge of .NET. The following are some of the assumptions we will make about your knowledge:

- You have a good understanding of C#
- You have a good understanding of one of the technologies you are generating for:
 - Web API
 - Angular
 - HTML5/jQuery
 - MVC
 - WCF Web Services
- You understand Object Oriented Programming principles
- You know how to use a relational database
- It is highly recommended that you read the chapters in this manual thoroughly.

If you unfamiliar with the above concepts, you might also purchase some mentoring services from PDSA, Inc. (www.pdsa.com).

Installation of Haystack

See Chapter 2 for full installation instructions.

Using Haystack Generated Code

Haystack generates generic .NET code that can be used in any version of .NET from 4.52 and later. The generated code relies on a few things in order to work.

1. Some PDSA .dlls need to be added to your Visual Studio project
2. A reference to System.Linq.Dynamic.dll (included in Haystack)
3. Configuration entries need to be added to your Web.Config or App.Config file in your Visual Studio project.
4. To distribute your .NET application you will need to create a runtime license from Haystack. A runtime license can only be generated from a purchased copy of Haystack, not the trial version.

Technical Support

Because of the complexity of software development, PDSA, Inc. does not provide free technical support related to the use of the generated source code in your applications, nor the creation of custom templates. Technical support is available for how to use Haystack, program bugs in Haystack, the sample source code, and the DLLs that are a part of this product and the accompanying documentation.

<p>TIP: It is strongly recommended that you read the owners' manual thoroughly before reaching out to our tech support team.</p>

Custom Templates

If you would like help designing custom templates for your particular situation, please contact us via email or phone. We are available to help you accomplish this task. PDSA, Inc. charges \$200/hour billed in 15 minute increments for the creation of custom templates for Haystack.

PDSA DLLs

Several PDSA DLLs are supplied for use with Haystack generated code. In the demo version of Haystack, these DLLs can only be run from within the Visual Studio environment. When you purchase the full version of Haystack you will be able to distribute these DLLs for use in a stand-alone EXE version of your application.

NOTE: PDSA does not supply the source code for these DLLs with Haystack. Source code can be purchased for an additional fee. Please contact PDSA for more information.

Fee based support

Fee-based technical support for C#, Visual Basic, SQL Server, WPF, Angular, HTML5, jQuery, MVC and Web Forms questions is available via telephone or email. PDSA, Inc. charges \$200/hour billed in 15 minute increments for this type of support (fees subject to change without notice). If we don't answer your question, then you don't pay.

Payment Options

You can pre-pay by check or money order, and we accept Visa and MasterCard. Please have the following ready when you call with a credit card:

- Credit Card Number
- Name on Card
- Expiration Date
- Address of where you receive the bill for this credit card

Support Options

The following support options are available to registered users of Haystack. This technical support is only available on the Haystack product itself.

- Send us an email with your support question to Support@pdsa.com.
- Fax us your question at (714) 734-9793.
- You can mail us a letter at 17852 17th Street, Suite 250, Tustin CA, 92780.
- You can telephone us at (714) 734-9792. You will receive 5 free support calls. After that you must pay our normal tech support fees as published on our web site.

Support Policy

The following options are available to you for support on Haystack for the first year.

- Automatic notification of upgrades and bug fixes
- 5 free phone calls
- Unlimited email on issues related to this product

Your first year's support on Haystack is included in the purchase price. For each year after we charge 20% of the retail price of the product.

NOTE: Major releases of Haystack are subject to an additional upgrade charge.

License Agreement

PLEASE READ THE EULA.RTF or EULA.TXT file in the installation folder of Haystack for our complete License Agreement.

Acknowledgements

Thanks to all the employees of PDSA, Inc for their help in developing this product.