



By: PDSA, Inc.

Version 1.x

April 2012
Published By: PDSA, Inc.

**Copyright © 2010-2012, 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 © 2010-2012 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.....	12
Goals of Haystack	13
System Requirements	15
Supported Databases.....	15
Your Knowledge	16
Installation of Haystack	16
Technical Support	16
Custom Templates.....	17
PDSA DLLs.....	17
Fee based support.....	17
Payment Options	17
Support Options.....	18
Support Policy.....	18
License Agreement	18
Ownership of the Software	19
Use and Redistribution Rights.....	19
Publication of Software.....	20
Restrictions.....	20
Warranty.....	20
U.S. Government Restricted Rights.....	21
General.....	21
Acknowledgements	22
Chapter 1	1-1
Haystack Overview	1-1
Haystack Overview.....	1-2
Philosophy.....	1-2
Goals of Haystack	1-3
What Haystack can Create.....	1-4
Data Access Classes.....	1-4
Business Rule Classes.....	1-4
Entity Classes	1-4
Stored Procedures.....	1-5
XML Data Classes	1-5

XAML CRUD User Control	1-5
ASP.NET CRUD User Control / Page	1-5
View Model Generation	1-5
WCF Services Generation	1-5
Why use Data Access Classes	1-6
A Haystack Project	1-6
Summary	1-7
Chapter Index	1-7
Chapter 2	2-1
Installation	2-1
Installation Procedures	2-2
PDSA Product Activation Screen	2-2
Limitations of Trial Version	2-3
Haystack Tips & Tricks Screen	2-4
Configure Haystack Screen Step 1	2-4
Configure Haystack Screen Step 2	2-5
Configure Haystack Screen Step 3	2-7
What to do if you don't have DBA rights	2-8
Configure Haystack Screen Step 4	2-8
Haystack Installation Folder	2-10
Sample Programs	2-11
Registering Haystack	2-13
Uninstalling Haystack	2-15
Summary	2-16
Chapter Index	2-16
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
Summary	3-12
Chapter Index	3-13

Chapter 4	4-1
Quick Start - ASP.NET MVC	4-1
Quick Start for ASP.NET MVC Applications.....	4-1
Add New Project.....	4-2
Generate CRUD Classes for Tables	4-4
Table Information Screen	4-4
Generate Table Data Classes	4-5
Test Generated Classes and MVC Pages.....	4-6
Summary	4-12
Chapter Index.....	4-13
Chapter 5	5-1
Quick Start - Silverlight.....	5-1
Quick Start for Silverlight Applications	5-1
Add New Project.....	5-2
Generate CRUD Classes for Tables	5-4
Table Information Screen	5-5
Generate Table Data Classes	5-5
Test Generated Classes and Web Form Pages.....	5-7
Copy Generated Files into Silverlight Web Project.....	5-8
Finish the Silverlight Project	5-12
Add WCF Service Reference.....	5-14
Add Views and ViewModels	5-15
Summary	5-18
Chapter Index.....	5-19
Chapter 6	6-1
Quick Start - WPF	6-1
Quick Start for WPF Applications	6-1
Add New Project.....	6-2
Generate CRUD Classes for Tables	6-4
Table Information Screen	6-4
Generate Table Data Classes	6-5
Test Generated Classes and User Controls.....	6-6
Summary	6-13
Chapter Index.....	6-14
Chapter 7	7-1

- Overview of an N-Tier Application 7-1
 - What is N-Tier? 7-2
 - Service Oriented Architecture 7-2
 - A Tiered Approach to Development 7-4
 - Implementation of N-Tier 7-5
 - Why N-Tier is a Good Choice 7-5
 - Why use Data Access Classes..... 7-6
 - Advantages of Separating Services 7-7
 - Disadvantages of separating services..... 7-7
 - Reusable Components 7-8
 - The Data Layer Component 7-8
 - The Data Classes (Data Model) 7-9
 - The Business Rules Component 7-9
 - Advantages of a Business Rules Component 7-10
 - Disadvantages of a Business Rules Component 7-10
 - How to achieve N-Tier..... 7-10
 - Map a Database Object to a Class..... 7-11
 - Goals of Data Classes 7-12
 - Benefits of Data Classes 7-12
 - Summary 7-13
 - Chapter Index..... 7-13
- Chapter 8 8-1
- Overview of the PDSA Data Model 8-1
 - The PDSA Data Model 8-1
 - Definitions..... 8-3
 - Overview of the Generated Code..... 8-3
 - PDSA Data Layer 8-4
 - PDSADataManager Class 8-4
 - GetDataSet Method 8-5
 - PDSA Data Providers Configuration Section..... 8-6
 - ExecuteSQL Method 8-7
 - Other Methods..... 8-9
 - Summary 8-10
 - Chapter Index..... 8-10
- Chapter 9 9-1
- Entity and Validation Classes..... 9-1
 - Entity and Validation Classes Defined 9-2

Validation Defined	9-2
A Simple Product Entity Class	9-3
Entity Base Class	9-4
Validator Class	9-6
Validator Class Files	9-7
PDSAProperty Class	9-11
Generated Validator Class.....	9-12
Using the Validator and Entity Classes	9-15
Validate Method.....	9-15
CheckBusinessRules Method.....	9-17
PDSA.Validation Namespace Overview.....	9-18
Summary	9-19
Chapter Index.....	9-19
Chapter 10	10-1
Using the Data and Manager Classes	10-1
Manager Class	10-2
Methods and Properties in Generated Manager Class.....	10-3
BuildCollection Method	10-3
Data Classes Overview	10-5
Properties of a Data Class.....	10-5
Methods of a Data Class	10-7
Data Classes Dynamic SQL and Stored Procedure Methods.....	10-8
Load Method.....	10-8
LoadByPK Method.....	10-9
GetDataSet Method.....	10-10
GetDataTable Method	10-11
Insert Method.....	10-12
Update Method	10-14
Delete Method	10-16
DeleteByPK Method	10-18
GetDataReader Method.....	10-19
RowCount Method	10-22
Working with Nulls.....	10-22
Data Classes with Dynamic SQL Only	10-23
Load using WHERE.....	10-23
RowCount Using a Where	10-26
Custom SELECT Statement.....	10-28
Custom WHERE Clause.....	10-29

Custom ORDER BY Clause	10-30
Custom INSERT Statement.....	10-31
Custom UPDATE Statement	10-33
Custom DELETE Statement.....	10-35
View Classes.....	10-37
Execute Stored Procedure Classes	10-37
Execute Stored Procedure with Output Variables	10-40
Read Only Stored Procedures	10-42
Relationships.....	10-42
Using Where Filters	10-45
Validate Data.....	10-48
DataObject Validate Method.....	10-48
DataObject CheckBusinessRules Method.....	10-49
Using Audit Tracking	10-50
Using Transactions.....	10-56
Summary	10-61
Chapter Index.....	10-61
Chapter 11	11-1
XML Data Classes	11-1
XML Data Classes.....	11-1
Product.xml	11-2
Methods in Generated XML Data Classes	11-2
BuildCollection Method.....	11-3
The Code for BuildCollection	11-4
Samples	11-6
Generating XML Data Classes.....	11-8
Summary	11-13
Chapter Index.....	11-13
Chapter 12	12-1
Generating Code.....	12-1
Code Generation	12-2
Project Information Screen.....	12-3
Project Information Screen (Generation Tab).....	12-5
Project Information (Options Tab)	12-5
Project Information (Paths Tab).....	12-7
Step-by-Step Scenarios	12-9

Table Data Classes using Dynamic SQL	12-9
Create Dynamic SQL Project.....	12-9
Load Tables.....	12-10
Create CRUD when Loading?	12-11
Select all Marked Ready to Gen?	12-11
Read Columns and Generate Default CRUD Statements	12-11
Generate Code for Dynamic SQL Classes	12-13
Table Data Classes using Stored Procedures.....	12-15
Create Stored Procedure Project.....	12-16
Load Tables.....	12-17
Read Columns and Generate Default CRUD Stored Procedures	12-17
Generate Code for Table Stored Procedure Classes	12-19
Table Data Classes using Dynamic SQL and/or Stored Procedures	12-21
View Data Classes	12-22
Stored Procedure Data Classes	12-22
Stored Procedures that Execute Data Modification Statements.....	12-23
Entity Class to Map To.....	12-23
Stored Procedures that Return Data	12-24
Summary	12-26
Chapter Index.....	12-26
Chapter 13	13-1
Haystack Templates.....	13-1
Templates.....	13-1
Templates.....	13-2
Code Snippets.....	13-2
Template Token Overview.....	13-2
Iteration Tokens	13-3
General Tokens.....	13-3
Remove Tokens	13-7
Current Database Object Tokens.....	13-8
XML Data Class Generation Tokens.....	13-14
Looping Tokens.....	13-14
Summary	13-17
Chapter Index.....	13-18
Chapter 14	14-1
Haystack XML Files	14-1

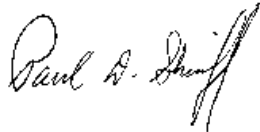
Overview of Haystack XML Files.....	14-2
Reason for XML Files	14-3
??_DotNetTypes XML File	14-3
??_TemplateGroups XML File.....	14-4
DatabaseServers XML File.....	14-4
Languages XML File.....	14-5
??_DBTypes XML File.....	14-6
??_DBTypesExcludes XML File	14-6
??_SPSettings XML File.....	14-7
??_Templates XML File.....	14-7
Prefixes XML File	14-7
ReservedWords XML File.....	14-8
Relationships between XML Files	14-8
.NET Language XML Files	14-8
Template Group XML Files.....	14-9
Database Servers XML Files.....	14-9
Database Servers and SQL Generation XML Files.....	14-10
Add New Code Generation Template	14-11
.NET Language Templates	14-11
Database Templates	14-12
Add New Default Dynamic SQL Statement.....	14-12
Add New Default Stored Procedure	14-12
Modifying .NET Type Defaults.....	14-13
Adding New Database Data Types	14-13
Prefix Replacement XML.....	14-13
Reserved Words XML	14-13
Summary	14-14
Chapter Index.....	14-14
Chapter 15	15-1
ASP.NET Deployment.....	15-1
Deploying an ASP.NET Web Project	15-1
Summary	15-3
Chapter Index.....	15-4
Chapter 16	16-1
MVVM Made Simple	16-1
Overview	16-1
Why MVVM?	16-2

The Sample	16-2
The Product Class	16-3
The ProductManager Class	16-5
The Brute Force Approach	16-6
Data Binding Basics in XAML.....	16-7
Eliminate More Code	16-9
Binding Other Properties	16-9
A Simple ViewModel	16-12
DataCollection and DetailData Properties	16-13
Populating the Data in the View Model Class.....	16-15
Hooking Into the View Model Class in the Code-Behind	16-16
Event Procedures in the Window Class	16-17
Summary	16-20
Chapter Index.....	16-20

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 VB.NET 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 loop at the end of the last name.

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 in VB.NET and C#
- 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
 - Add additional CRUD statements and WHERE and ORDER BY statements when using Dynamic SQL.
 - Generate business rules for each column in the table based on the schema
 - Allow you to add some business rules to Haystack that is generated each time.
 - 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.
 - You may add on additional WHERE and ORDER BY statements
- 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 Classes for XML Files that are single level (like a table)
 - Creates a data class that can read and write element or attribute based XML files.
 - You can add business rules for the XML files.
- 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 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 Silverlight 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
 - 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 Silverlight user control.
 - All data access is called from the ViewModel class.
 - The ViewModel interacts with the generated WCF Service Layer
- Generate ASP.NET Add/Edit/Delete User Controls and Pages
 - Creates a detail user control/page that generates a Label and TextBox for each field in the table.
 - Automatically display validation rule errors on bad input

- Uses an ObjectDataSource bound to the generated view model class.
- Creates a GridView bound to the ObjectDataSource for displaying all records.
- A View Model class is generated that is used from within all events in the user control. All data access is called from the ViewModel class.

System Requirements

Below are the assumptions about your computing environment.

- You are running Visual Studio.NET VS.NET 2010 with SP1 or higher
- You are using SQL Server 2005/2008 express or higher for storing the Haystack meta-data.
- You are using SQL Server 2005 or above to generate classes from.
- You are using Oracle 10g or above to generate classes from.
- You are running Windows XP Professional, Windows 2003 Server, Windows Vista or Windows 7. You may use either 32-bit or 64-bit.
- To run a Demo 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 2005 and higher
- Oracle 10g and higher (tables and views only, stored procedures and packages coming soon)

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# and/or VB.NET
- You have a good understanding of WPF, Silverlight and/or ASP.NET.
- 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.

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 calling tech support.</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#, VB, SQL Server, WPF, Silverlight and/or ASP.NET 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 E. 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

The Haystack software is licensed on a per-user basis. This means that each developer using this product must have a valid license and activation key. You may contact PDSA, Inc. to request additional licenses. Each license is attached to a particular login id and machine name on each machine. If you do not have a login id and machine name on your machine you will be asked to supply one when you register the product. Registration takes place over the internet, so you must have an internet connection to register this software.

You may not change the license or take the license to another machine as this would be in violation of the license agreement.

Ownership of the Software

The enclosed software program (“Software”) and the accompanying written materials are owned by PDSA, Inc. or its suppliers and are protected by United States copyright laws, by laws of other nations and by international treaties. You must treat the Software like any other copyrighted material except that you may make one copy of the Software solely for backup or archival purpose, and you may transfer the software to a permanent storage device for archival purposes.

Use and Redistribution Rights

You have the right to redistribute any code you generate from Haystack, as well as any of the DLLs from Haystack needed to run the generated code. You may not redistribute the Haystack product to any other user, nor does your end user have any rights to use the DLLs in their development efforts.

This use and redistribution rights are on a non-exclusive, royalty free basis, provided that:

- Each developer using the program owns a license of the software;
- You do not distribute the Haystack product or any DLL that Haystack uses exclusively, except those DLLs used to support your generated code.
- You agree to indemnify, hold harmless, and defend PDSA, Inc. and its suppliers or contractors from and against any claims or lawsuits, including attorneys’ fees, that arise or result from the use or distribution of or other activities relating to your software;
- If you claim a copyright, you must add a clause starting “Portions of this program are "Copyright 2010 by PDSA, Inc.” and that you do not claim ownership of those portions of the software.

Publication of Software

You are prohibited from publishing the Haystack source code on any medium, including but not limited to, electronic mail, online services, blogs, internet newsgroups, and/or printed matter. The Haystack source code may not be included in freeware or shareware applications in a non-compiled state.

Restrictions

Under no circumstances may you attempt to reverse engineer this product. You may not rent or lease the Software, but you may transfer the Software and the accompanying written materials on a permanent basis provided you retain no copies, the recipient agrees to the terms in this Software License, and you have not used any of the Haystack source code in an application. Ownership transfers must be reported to PDSA, Inc. in writing. You must also deactivate your product using the appropriate mechanisms in the software.

Warranty

Except as specifically stated in this agreement, the software and software documentation is provided and licenses “As Is” without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

PDSA, Inc. warrants for a period of 30 days from the date of delivery to you that (1) the Software will substantially conform to the Software documentation and specifications published by PDSA, Inc., and (2) the media on which the Software is furnished and the documentation will be free from defects in materials and workmanship under normal use. PDSA, Inc. does not warrant that the functions contained in the Software will meet your requirements or that the operation of the software will be either error free or appear precisely as described in the Software documentation. You are responsible for selecting the Software to achieve your intended results and for the installation, use and results obtained from the Software. The Software must be used with the computer hardware and operation system for which it is designed, and in accordance with the Software documentation and specification requirements.

PDSA, Inc. will replace defective media and documentation and correct substantial Software non-conformities at no charge, provided you return the item with dated proof of payment (in case of defect) or otherwise notify PDSA, Inc. in writing (in case of substantial Software non-conformity) within 30 days from the date of delivery to you. If PDSA, Inc. is unable to replace defective media or documentation or correct substantial Software non-conformities, PDSA, Inc. will refund the license fee. These are your sole remedies for any breach of warranty. You or your dealer may return the enclosed registration card to PDSA, Inc. in order for these remedies to be available to you and for you to be eligible for customer support and service during the warranty period.

In no event will PDSA, Inc. be liable to you for any damages, including without limitation loss profits or revenues, loss of data, business interruption loss, recovery or substitution costs, or claims by third parties, or other indirect incidental or consequential damages, arising out of the use of or inability to use the software, even if PDSA, Inc. or a PDSA, Inc. dealer has been advised of the possibility of such damages. In no case shall PDSA, Inc.'s liability exceed the amount of the license fee paid by you for the software.

This limited warranty gives you specific legal rights. Some jurisdictions provide other rights, and some jurisdictions do not allow the exclusion or limitation of implied warranties or limitation of liability for incidental or consequential damages. As a result, the above limitations and/or exclusions may not apply to you.

U.S. Government Restricted Rights

The software and software documentation are provided with Restricted and Limited Rights. Use, duplication or disclosure by the Government is subject to restrictions as set forth in DFAR 252.277-7013(c)(1)(ii) or FAR 52.227.19 as applicable, and additional restrictions set forth in this agreement. Contractor/Manufacturer is PDSA, Inc. located at 17852 E. 17th Street, Suite 205, Tustin, CA 92780.

General

This agreement is governed by the laws of the state of California. The prevailing party in any action or proceeding brought in connection with a breach of this agreement will be entitled to reimbursement by the other party for costs and reasonable attorneys' fees.

Any questions concerning this Agreement should be referred in writing to PDSA, Inc. located at 17852 E. 17th Street, Suite 205, Tustin, CA 92780.

You acknowledge that you have read this agreement, and by opening this software package indicate your acceptance of its terms and conditions. You

also agree that it is the complete agreement between us and that it supersedes any prior information or representations you received relating to the subject matter of this agreement. No PDSA, Inc. dealer, distributor representative or employee is authorized to make any modifications or additions to this agreement.

THE WARRANTY AND REMEDIES SET FORTH IN THIS LIMITED WARRANTY ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESSED OR IMPLIED.

Acknowledgements

Thanks to all the employees of PDSA, Inc for their help in developing this product. In addition we had several beta testers and others that made significant suggestions and helped us test this product.

- Susan Daffron
- Ashraf Ali
- Thomas Beyerlein
- Dale Mitbo
- David Coyer
- George Hornberger