Getting Started

with

Design Pattern Framework^{TM} 4.0

by

Data & Object Factory, LLC
www.dofactory.com

Copyright © 2006 - 2010, Data & Object Factory, LLC.
All rights reserved

1. Index

Copyright © 2006-2010, Data & Object Factory, LLC. All rights reserved.

2. Introduction

Welcome to the *Design Pattern Framework* ^{TM} *4.0*, a unique package that will help you join the ranks of expert .NET developers and architects.

This document will show you how to get started with design patterns and pattern architectures. It discusses setup and installation followed by a section on how to optimize your learning experience with this extensive resource.

For those who have used prior versions of the Design Pattern Framework, a ‘What is new in 4.0’ section can be found at the bottom of this document. If you are new to the Framework please continue reading.

But first, we ask that you review our simple copyright and license notice.

Copyright and License

Copyright (C) 2006-2010 Data & Object Factory, LLC. All rights reserved.

If this is a single-user license the product can only be used by a single user. If you purchased the 16-user license then the product can be used by up to 16 developers concurrently. Site licenses are restricted to a single facility at a given physical address.

You are allowed to use the source code in your own business application development, but remember that this product and the source code are intended for educational purposes only.

THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.

3. Setup

Next, we discuss setup and installation.

Versions of Visual Studio and .NET

The Design Pattern Framework 4.0 requires Visual Studio 2010 and .NET 4.0. Please note that Visual Studio 2010 *Express* (the free version) is not supported. Older versions of the Design Pattern Framework (.NET 3.5, .NET 2.0 and .NET 1.1) are available from our website.

To run the Framework it is necessary that you have SQL Server Express installed.

To enable the Silverlight Patterns application you must have the Silverlight 4 Tools for Visual Studio 2010 installed (download from Microsoft) and the Silverlight 4 Toolkit (download from CodePlex).

Installation

Run the 'Design Pattern Framework 4.0 **.msi' installer program (where ** = CS for the C# edition, and VB for the VB.NET edition). This will install the complete package with all necessary files. We recommend that you follow the suggested folders in the installation program. They are:

Windows XP

C# Edition: C:\Documents and Settings\%username%\My Documents\DoFactory Patterns 4.0 CS.**

VB Edition: C:\Documents and Settings\%username%\My Documents\DoFactory Patterns 4.0 VB.**

Vista / Windows 7

C# Edition: C:\Users\%username%\Documents\DoFactory Patterns 4.0 CS.**

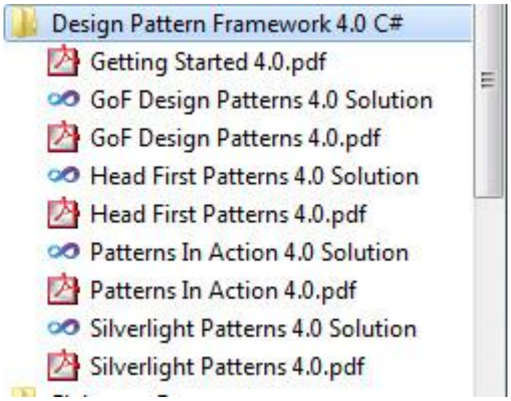
VB Edition: C:\Users\%username%\Documents\DoFactory Patterns 4.0 VB.**

Copyright © 2006-2010, Data & Object Factory, LLC. All rights reserved.

Following installation, you will have 5 subfolders and 4 pdf e-books in these folders. They are:

\Gang of Four\	documented in 'Gang of Four Design Patterns 4.0.pdf'
\Head First\	documented in 'Head First Design Patterns 4.0.pdf'
\Patterns In Action\	documented in 'Patterns in Action 4.0.pdf'
\Silverlight Patterns\	documented in 'Silverlight Patterns In Action 4.0.pdf'
\Visio UML Diagrams\	no separate document

Also installed are these Windows menu items for easy access.


Design Pattern Framework 4.0 C#
Getting Started 4.0.pdf
GoF Design Patterns 4.0 Solution
GoF Design Patterns 4.0.pdf
Head First Patterns 4.0 Solution
Head First Patterns 4.0.pdf
Patterns In Action 4.0 Solution
Patterns In Action 4.0.pdf
Silverlight Patterns 4.0 Solution
Silverlight Patterns 4.0.pdf

As an alternative to these menus you can create shortcuts on your desktop. Just right-click on the menu items and select Send To > Create Shortcut (desktop).

1: Gang of Four (GoF)

This folder contains 69 Gang-of-Four design pattern projects, i.e. Structural, Real world, and .NET optimized.

Documentation is available in the 'Gang of Four Design Patterns 4.0.pdf' document.

Copyright © 2006-2010, Data & Object Factory, LLC. All rights reserved.

2: Head First

This folder contains 46 Head First design pattern projects. These are .NET versions of the Java based examples from the Head First Design Patterns book (published by O'Reilly Media).

Documentation is available in the 'Head First Design Patterns 4.0.pdf' document.

3: Patterns in Action

This folder contains a large reference e-commerce application, named 'Patterns in Action'. It demonstrates the use of design patterns in a real-world environment using Web, Win Forms, and WPF UI platforms. The application contains 20 projects including an ASP.NET Web Forms application, an ASP.NET MVC 2 application, a Windows Forms application, a WPF application, WCF hosting and WCF service projects, and several class libraries.

The default startup project is the *ASPNET Web Forms Application*, but you can change this to the *ASP.NET MVC Application (or Test version),* the *Windows Forms Application* or the *WPF Application*..

'Patterns in Action' uses the Microsoft ASP.NET 4.0 built-in web server (code named Cassini). When running you will see 3 ASP.NET Development Servers popup. If you want to use IIS instead, you will need to create 3 new IIS virtual directories, one for the web application, and two for the web services. Also, adjust the WCF configuration and project Start Options accordingly.

Documentation is available in the 'Patterns in Action 4.0.pdf' document.

4: Silverlight Patterns

This folder contains the new Silverlight Patterns in Action reference application. Functionally it is the same as the Patterns in Action application, but the architecture is

Copyright © 2006-2010, Data & Object Factory, LLC. All rights reserved.

very different (using WCF RIA Services + Entity Framework). As mentioned earlier, it requires that you have Silverlight 4.0 SDK and the Silverlight Toolkit installed.

Documentation is available in the ‘Silverlight Patterns 4.0.pdf’ document.

5: Visio UML Diagrams

This folder contains 23 Visio diagrams of the Gang of Four (GoF) design patterns.

4. Database Setup

The ‘Patterns in Action’ application is configured to run the Entity Framework against a local SQL Express database named *Action.mdf*. The database files are located under project ‘WCF.ActionServer’ in the \App_Data\ folder.

You can change to SQL Server by creating a new SQL Server database (for example, named *Action*) and run the *action.sql* script (located in a folder named \Solution Items\SQL Server\). Then, in web.config (in project ‘WCF.ActionServer’) under <appSettings> change the ConnectionStringName to *EntityFramework.SqlServer*. The application will now run against SQL Server.

Alternatively, you can use MS Access by changing the same web.config as follows: set the DataProvider to *System.Data.OleDb* and set the ConnectionStringName to *ADO.NET.Access*.

The application uses the Microsoft Membership services. By default, the Membership database is configured to use SQL Express. If you wish you can change this to SQL Server. Please note that MS Access does not support Membership, so when running against MS Access, you will also need access to either SQL Express or SQL Server for Membership services.

With SQL Server and SQL Express, you have the additional option to change data access from Entity Framework to either LINQ-to-SQL or ADO.NET. For details consult the ‘Patterns In Action 4.0.pdf’ document.

The ‘Silverlight Patterns’ application is configured to run against its own copy of a local SQL Express database named *Action.mdf*. This database, together with the membership database (aspnetdb.mdf), is located under the standard \App_Data\ folder. Changing it to SQL Server requires similar steps as described for ‘Patterns in Action’.

5. Learning about patterns

What are Design Patterns

Design patterns are advanced object-oriented techniques. To get the most out of this package, it is best that you are familiar with object-oriented programming in C# or VB. It is assumed you're familiar with concepts such as classes, objects, interfaces, encapsulation, inheritance, polymorphism, etc.

As a first step to understand design patterns you need to know what they are and why they exist. On our website we explain it as follows:

Design patterns are recurring solutions to software design problems you find again and again in real-world application development. Patterns are about design and interaction of objects, as well as providing a communication platform concerning elegant, reusable solutions to commonly encountered programming challenges.

Essentially what this means is that developers do not want to keep re-inventing the wheel for problems that are common and re-occurring. So, they describe the 'wheel pattern' that will solve all problems that require the 'ability to easily move over the ground'. Clearly, there are different types of wheels (for cars, bikes, wheelbarrows, etc), but they all have in common that the solution requires a device that is circular with a center axle (this is the pattern).

It takes time and experience before patterns are recognized or 'discovered'. But once they are proven and documented, other developers can start benefitting because they don't have to go through the same trial-and-error process. This greatly accelerates productivity of .NET developers. Another benefit is that your code will become more effective, more robust, and more elegant.

Copyright © 2006-2010, Data & Object Factory, LLC. All rights reserved.

Exploring Design Patterns

You'll find plenty of information on design patterns in books and on the Internet (wikis, forums, etc). However, what is missing are pattern examples in the context of real-world applications, that is, ***when, where, and how design patterns are used***. Most examples are simple and not representative of the real world. This makes it hard for .NET developers to learn about design patterns and use them in their own projects. Our *Design Pattern Framework 4.0* is designed to address this very issue. Below are two possible starting points for you to start exploring and learning about design patterns.

Option A: Gang of Four

The 23 Gang of Four (GoF) design patterns are considered to be the foundation of all patterns. They are called Gang of Four because they were first published in a book, titled *Design Patterns*, written by 4 co-authors. If you are new to design patterns, we recommend you start with these patterns. The pattern solution with 69 projects is located under folder \Gang Of Four\ and is described in file 'Gang of Four Design Patterns 4.0.pdf'.

In our code samples, each of the GoF design patterns is available in 3 forms: *structural*, *real-world* and *.NET optimized*. The structural form follows the original, somewhat abstract, definition with original class names. The real-world follows the same structure, but is applied to an easy to understand real-world problem. Finally, the .NET optimized form solves the same problem but exploits the latest, most effective programming techniques available in .NET 4.0.

We suggest you start with the most popular patterns. In 'Gang of Four Design Patterns 4.0.pdf' you will find for each of the 23 patterns their frequency-of-use. Popular patterns include Factory, Singleton, Facade, Adapter, and Observer. In this document we also explain when and where each pattern is used and how Microsoft uses the pattern in their .NET Framework libraries.

Option B: Head First

Alternatively, you could start by exploring the 'Head First' design pattern path. These patterns are based on a book titled *Head First Design Patterns.* Many developers really enjoy this book. It explains design patterns in a light-hearted, easy-to-understand manner. The 'Head First' patterns in the book are the same as the 23 GoF patterns discussed in Option A above, but they are written for Java developers and all code examples are in Java.

As part of this Design Pattern Framework you have received .NET versions of these Head First Java examples. This will allow you to read about the patterns in the book while referencing our .NET Head First design pattern code examples. This combination of an easy-to-read book with .NET code samples will get you up to speed quickly with design patterns. The .NET Head First design patterns are located under folder \Head First\ and associated documentation is found in 'Head First Design Patterns 4.0.pdf'.

UML Diagrams

To support the study of GoF patterns (both Option A and Option B), you will find visual representations of the pattern classes in a UML class diagram for each of these 23 patterns in the \Visio UML Diagrams\ folder.

Applying Design Patterns

Once you're familiar with the concept of design patterns and have explored the more commonly used GoF design patterns, we suggest you move on to the 'Patterns in Action' reference application. This comprehensive e-commerce application demonstrates ***when, where, and how design patterns are used in today's modern 3-***

tier architectures. 'Patterns in Action' demonstrates modern pattern architectures for an entire application – i.e. top to bottom. The .NET solution is located under folder \Patterns in Action\. The associated documentation is called 'Pattern in Action 4.0 pdf'.

'Patterns in Action' is a comprehensive 3-tier application that supports 4 different UI platforms (ASP.NET Web Forms, ASP.NET MVC, Windows, and WPF). Each of these UI platforms consumes the exact same WCF Web services layer (the Application Façade Pattern). Understanding all the details of the application will take some time and it is best not to expect this to happen overnight. Many of our users tell us that they keep going back to this reference application over and over again for when they need detailed architectural guidance on their own .NET projects. To get the most out of the application we suggest the following steps (details follow later):

Begin by exploring the functionality of the application, that is, navigate through the application and get a feel of what the end-user can do; they are fairly self-explanatory. After that, you can explore the high level organization of the solution and projects including the different layers and their relationships. Next, you should dig a little deeper and start exploring the individual projects and the roles they play, such as, Log, DataObjects, BusinessObjects, etc.

Once you are familiar with the different projects, you need to understand the data flow, that is, how the data gets from the database all the way to the UI and then back again. The debugger is of great help in tracing the data. By the time you understand the data flow you will have a pretty good understanding of the overall 3-tier architecture. With that, you are ready to explore the individual GoF patterns that are scattered throughout the application, such as Proxy, Factory, Singleton, Strategy, and Observer. Try to understand what the purpose is of these patterns and how they are implemented. Finally, you can explore the numerous newer patterns, including the Enterprise patterns, the SOA and Messaging Patterns, as well as the Model-View design patterns.

If you are developing with Silverlight or are planning to, then the Silverlight Patterns application will be of great interest to you. It is a separate Visual Studio Solution and is located under folder \Silverlight Patterns\. The associated documentation is available in 'Silverlight Patterns 4.0 pdf'.

The Silverlight Patterns application has the same functionality as Patterns in Action, but is built on a very different architecture using WCF RIA Services and its associated patterns and practices. Also here, it is best to start exploring the functionality first, then the Visual Studio solution, and only then the code base itself, with design patterns, etc.

Let's review these steps in some more detail.

Step 1: Functionality

This is where you find out what the application actually does. 'Patterns in Action' is an e-commerce application that allows users to shop for electronic products. We suggest you start with the ASP.NET Web Forms Application. Ensure that this project is set to be the startup project (the default setting). Start browsing, searching, and sorting products. View product details and add these to the shopping cart. Add several products with different quantities to the cart. Change the cart by changing quantities (and recalculate), removing items, and setting different shipping options.

The shopping experience is in a public area. The administration area is secure and requires the user to login. Login (credentials are provided) and then browse, add, edit, delete customers. View customer's orders and order details. Once you're done you can logout.

Next, change the startup project to the ASP.NET MVC Application. Run this application and confirm that look-and-feel and functionality are exactly the same as the Web Forms application.

After that, change the startup project to Windows Forms Application. Run the application and login. Select different customers on the left-hand tree control and see that the orders and order details adjust accordingly. Add a new customer, and edit, delete customers. When you are done you can logout and exit.

Finally, change the startup application to WPF Application. Login and manipulate customers by hovering over, double clicking, and right clicking on the customer images. Right click will allow you to add, edit, and delete customers. Double clicking on a customer will open the order and order details window. Once completed, you can logout and exit.

It is very important that you realize that the four UI platforms consume the *exact same* WCF service layer (the Application Façade pattern). At a high-level, this entire solution is built around 3 layers: 1) the Presentation Layer, 2) the Business Layer, and 3) the Data Layer. By switching startup projects you only change the Presentation Layer, not the underlying layers or database.

You may find that the applications are running a bit slow. The reason for this is the WCF Hosting configuration (via HTTP). To keep the configuration simple, we opted to have all three applications use the same protocol, but in a real-world production environment you will most likely configure these differently (ideally using IIS7).

As mentioned the 'Silverlight Patterns' application is functionally the same, but architecturally very different. The Silverlight client consumes methods that are made available through WCF RIA Services and its architecture is closer to that of a 'Client-Server' model. Architecture and patterns of this application are described in the documentation.

Step 2: Overview Level

This is where you learn about the organization of the 'Patterns in Action' solution which consists of 20 different projects and 5 different layers (although the architecture is still referred to as 3-tier). The layers are arranged in folders that have numbers so that they display in a logical order, that is, from top to bottom. The presentation tier (Layer 1) contains 4 different UI platforms: ASP.NET Web Forms, ASP.NET MVC, Windows Forms and WPF. The service tier is represented by the WCF Hosting and WCF Services (Layers 2 and 3). There are two different services: ActionService (application services) and ImageService (for presenting customer and product images). The service layer typically handles data validation, authentication, authorization, and transactions.

The next two layers are Business Layer (Layer 4) and Data Layer (Layer 5). The business layer contains business objects with business rules. The data layer performs database access and demonstrates the use of 3 different access technologies: ADO.NET, LINQ-to-SQL, and Entity Framework. You configure the database, as well as the data access technology used, in the web.config file under project 'WCF.ActionServer'

Step 3: Detail Level

Here you focus on the individual projects. It is best to start at the bottom and then move up until you reach the UI layer. Start with DataObjects, then BusinessObjects, then Action Service, then.WCF.ActionServer, and finally one of the presentation tier platforms. The references folder in each project shows you the dependencies on other projects. The 'Patterns in Action 4.0 pdf' guide describes each project in more detail.

Step 4: Data Flow

By this time you should have a reasonable understanding of the data flow. This means that you know how the application takes the data from the database, the moves it via the different layers to the UI layer and then back again to the database (usually to save or update changes made by the user). It is important that you understand the different data container types that hold the data as it travels through the layers. These containers include: Entities, Business Objects, and Data Transfer Objects.

Remember, that in your own projects it may not be necessary to use all container types presented in this reference application; in many cases, a handful of Business Objects is all you need.

Copyright © 2006-2010, Data & Object Factory, LLC. All rights reserved.

As mentioned before, the debugger is a wonderful tool to follow the data as it travels through the different layers.

Step 5: GoF Design Patterns

The Gang of Four design patterns are located throughout the Framework projects, but mostly in the Data Layer and Framework projects. It is suggested that you identify these (browse the code and explore the code comments), study the source code, and with your prior understanding (based on Option A or B) you will be getting a good handle on when, where, and why they are used.

Step 6: Modern Design Patterns

The newer and more recently discovered design patterns are also located throughout the application. The Enterprise Patterns are generally located where the GoF are, that is, in the Data Layer and Framework projects. The SOA and Message patterns are located in the WCF service layer.

The Model-View patterns are always located in the presentation layer: MVC in the ASP.NET MVC application, MVP in the Windows Forms application, and MVVM in both the WPF and Silverlight applications. MVVM is probably the last pattern to study because it is one of the most complex patterns and requires that you are familiar with WPF / Silverlight commands, visual state management, and data binding.

6. What is new in 4.0

Here is a brief summary of what's new in the Design Pattern Framework 4.0.

New in 4.0 are:

1) ASP.NET MVC 2 Reference Application
    - Routing, Areas, Unit testing, Mocking, Dependency Injection, more…
2) Silverlight 4 Application
    - MVVM, Managed Extension Framework, Charting, more…
3) Entity Framework 4 (in Patterns in Action and Silverlight Patterns)
4) Repository Pattern

Enhancements in 4.0 include:

1) ASP.NET Web Forms (and MVC)
    - New Look & Feel
    - SEO (Search Engine Optimization)
2) ADO.NET data access
    - Improved performance
    - Improved security (prevents SQL Injection)
    - Note: ADO.NET changes are very significant.
3) Improved code effectiveness (throughout all projects)

Details of these and other features are discussed in the respective pdf documents.

7. Summary

As you can see, the *Design Pattern Framework 4.0* is a unique and comprehensive package that captures and presents design patterns in a real-world setting with several .NET reference applications. We are hopeful that you will be able to apply many of these patterns and practices in your own projects and build .NET applications that are robust, easy-to-maintain, elegant, and, above all, successful.