Getting Started
with the
Design Pattern Framework 4.5

by

Data & Object Factory, LLC
www.dofactory.com

1. Index

1. Index 2
2. Introduction 3
   Copyright and License 3
3. Setup 4
   Versions of Visual Studio and .NET 4
   Installation 4
      1: Gang of Four (GoF) 5
      2: Head First 5
      3: Patterns in Action 6
      4: Spark 6
      5: Plan 6
      6: Visio UML Diagrams 6
4. Database Setup 7
5. Learning about patterns 7
   Exploring Design Patterns 8
      Option A: Gang of Four 8
      Option B: Head First 9
      UML Diagrams 10
   Applying Design Patterns 10
      Step 1: Functionality 11
      Step 2: Solution 11
      Step 3: Layers 12
      Step 5: Data Flow 14
      Step 6: Design Patterns 15
6. What is new in 4.5 15
7. Summary 16

2. Introduction

Welcome to the *Design Pattern Framework ™ 4.5*, a unique package that will get you up to speed with .NET design patterns and practices.

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

If you have used prior versions of the Design Pattern Framework, we have a 'What is new in 4.5' section towards the end 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

3. Setup

Next we'll discuss setup and installation.

Versions of Visual Studio and .NET

The Design Pattern Framework 4.5 requires .NET 4.5 and Visual Studio 2012 *Professional* or better. Visual Studio 2012 *Express* (the free version) is not supported.

Installation

The package comes in a zip file. Simply unzip and you're ready to go. It is suggested you unzip to this location (although any folder will work just fine).

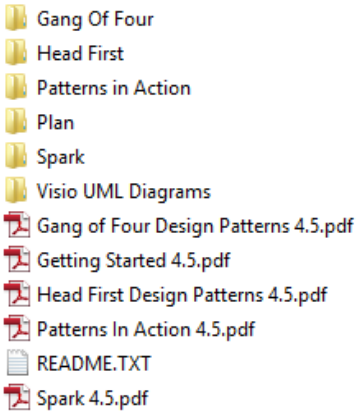
C# Edition: C:\Users\%username%\Documents\DoFactory\DPF\4.5\CS.**
VB Edition: C:\Users\%username%\Documents\DoFactory\DPF\4.5\VB.**

Following installation, you will have 6 subfolders and 5 pdf e-books in these folders. They are:

\Gang of Four\	documented in 'Gang of Four Design Patterns 4.5.pdf'
\Head First\	documented in 'Head First Design Patterns 4.5.pdf'
\Patterns In Action\	documented in 'Patterns in Action 4.5.pdf'
\Spark\	documented in 'Spark 4.5.pdf'
\Plan\	documented in 'Spark 4.5.pdf'
\Visio UML Diagrams\	no separate document

For easy access you could create a desktop shortcut to the root folder of the projects and documents. Simply drag and drop the folder to your desktop. Then give it a short name. It will look like this:

Double click on the shortcut and you will have all projects and documents available to you.



1: Gang of Four (GoF)

The \Gang Of Four\ 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 4.5.pdf' document located in the root folder of the package.

2: Head First

The \Head First\ 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 4.5.pdf' document located in the root folder of the package.

3: Patterns in Action

The \Patterns in Action\ folder contains a pattern-based reference application named 'Patterns in Action'. It demonstrates the use of design patterns in a wide range of applications. The solution contains 8 projects including an MVC app, a Web Forms app, a Windows Forms app, and a WPF app. The default startup project is the *MVC application*, but you can change this to any of the other 3 application types.

Documentation is available in the 'Patterns in Action 4.5.pdf' document in the root folder of the package

4: Spark

The \Spark\ folder contains the Art Shop online store which was built with our new Spark rapid application development (RAD) platform. Spark is an exciting new addition to this package. It is based on a lightweight, pattern-based architecture and a select set of patterns, practices, and tools. Spark will allow you to build your applications easily and quickly. Core patterns include: Repository, Unit of Work, Façade, Active Record, DTO, and CQRS. All of this will be explained in the documentation.

Documentation is available in the 'Spark 4.5.pdf' document in the root folder

5: Plan

The \Plan\ folder contains a starter project that allows you to get up to speed with the Spark platform quickly.

Documentation is available towards the end of the 'Spark 4.5.pdf' document in the root folder

6: Visio UML Diagrams

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

4. Database Setup

All applications in this package are configured to run against a LocalDb database instance. The database files (*.mdf) are located in the \App_Data folders in their respective projects.

You can easily change from LocalDb to SQL Server. Start by creating a new SQL Server database. Then run the relevant script (named Action.sql or Art.sql) against this database. The script files are located in a solution folder named /Solution Items/Sql Server. This will create the database *and* the data. Then finally change the connection string in the relevant web.config or app.config files. The application(s) are now ready to run against SQL Server.

5. Learning about patterns

Design patterns are advanced object-oriented techniques. It is assumed that you have experience programming in C# or VB and that you're familiar with OO concepts such as classes, objects, interfaces, encapsulation, inheritance, etc.

To understand design patterns you need to know what they are and why they exist. On our website we explain patterns like so:

Design patterns are 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 says is that developers do not want to keep re-inventing the wheel for problems that occur over and over again. So, they come up with 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). Furthermore, when a developer talks about a 'wheel' all other developers know exactly what he/she means: 'wheel' has become part of the vocabulary of developers.

It takes time and experience before a pattern is recognized or ‘discovered’. But once it is proven and documented, other developers will benefit because they don’t have to go through the same trial-and-error process. This greatly accelerates productivity of .NET developers.

Exploring Design Patterns

There is plenty of information on design patterns available in books and on the Internet (wikis, forums, etc.). However, these sources do not provide solid examples that show ***when, where, and how*** these patterns are used in real-world apps. Most examples are simple and do not represent the real world. This makes it hard for .NET developers to learn about design patterns and know how to apply these in their own projects.

The *Design Pattern Framework 4.5* is designed to address this problem. First, it includes structural, real-world, and .NET optimized versions of all the Gang of Four design patterns. Secondly, this release includes two full-stack reference applications: one called *Patterns in Action 4.5* and the other *Spark 4.5*; the latter implements a stylish Art Shop where customer can purchase art reproductions.

Below are two possible starting points on your journey to learn about design patterns.

Option A: Gang of Four

The 23 Gang of Four (GoF) design patterns are considered 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. The pattern solution with 69 projects is located under the /Gang of Four folder and is described in the 'Gang of Four Design Patterns 4.5.pdf'.

The GoF design patterns are 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 4.5.

We suggest you start with the most popular patterns. In the 'Gang of Four Design Patterns 4.5.pdf' document you will find for each of the 23 patterns their frequency-of-use. Popular patterns include Factory, Singleton, Facade, Adapter, and Observer. In that same 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 Step A, 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 should get you up to speed quickly with patterns. Please note that the 'Head First' book itself is not included in this package.

The .NET Head First design patterns are located in the /Head First folder and the associate documentation is found in 'Head First Design Patterns 4.5.pdf'.

UML Diagrams

To support the study of GoF patterns (both Step A and Step B), you will find visual representations of the pattern classes in a UML class diagram for each of these 23 patterns in the /Visual UML Diagrams/ folder. To open these files you will need a copy of the Visio (Viewer or the full version).

Applying Design Patterns

Once you're familiar with the concept of design patterns and have explored the commonly used GoF design patterns, you are ready to move on to the next two categories of patterns: the Enterprise patterns and the Model View patterns.

These patterns are best studied in the context of a real-world app. To this end, this package comes with 2 comprehensive applications: they are called *Patterns in Action 4.5* and *Spark 4.5*. It is best to start with *Patterns in Action* followed by *Spark*.

These applications demonstrate when, where, and how design patterns (GoF, Enterprise, and Model View) are used in a multi layered architecture. The use of patterns is demonstrated throughout the entire application stack, that is, anywhere in the application. The Patterns in Action solution supports 4 different UI platforms: MVC, Web Forms, Windows Forms, and WPF. Interestingly, all UI platforms consume the exact same Service, Business, and Data layers.

New in this release is *Spark 4.5. Spark* is a light-weight application platform that allows developers to build applications quickly and easily. It uses a limited set of pattern, practices and tools, which helps in creating very fast and agile solutions.

Art Shop is the name of the reference application built with Spark. It is an online store where users purchase art reproductions by famous classic painters, such as, Van Gogh,

Monet, and Cezanne. The Art Shop is a comprehensive, full-stack solution that demonstrates the use of Spark in building modern, responsive MVC web apps.

Understanding and internalizing all the details of *Patterns in Action* and *Spark* 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 our reference applications over and over again when they need architectural guidance and sample code for their own .NET projects.

To get the most out of these two reference applications we suggest the following learning path:

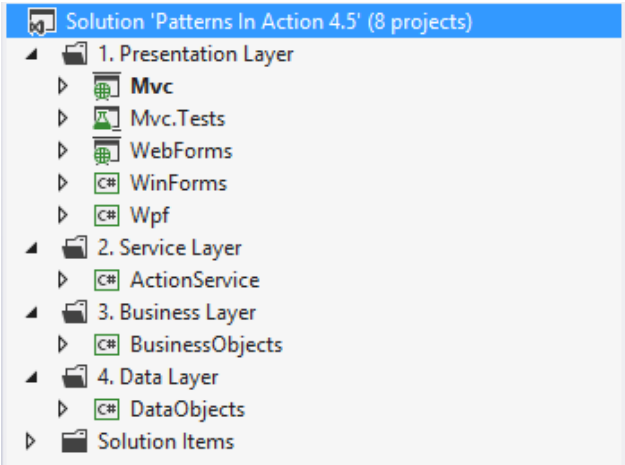
Step 1: Functionality

Begin by exploring the functionality of the applications, that is, navigate through the pages and the forms and get a feel of what the user can do. Most of it is fairly self-explanatory. Don't forget to login as an Admin as the administration pages cover large areas of functionality. The admin login for both apps is *debbie@company.com* with password *secret123*.

The *Patterns in Action* solution supports 4 different UI platforms: MVC, Web Forms, Windows Forms and WPF. Select any of these as a startup project and run through these as well. The *Spark* solution focuses on MVC and REST.

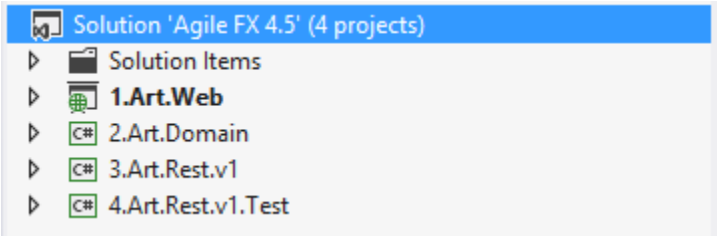
Step 2: Solution

Once you understand what these applications do, explore their solutions and their projects. The *Patterns in Action* solution has 8 projects in a folder structure that clearly expresses the project and layer relationships. See the next figure for this.



Each layer has its own folder. They are numbered 1 through 4. The presentation layer at the top supports and contains 4 different UI platforms (MVC, WebForms, WinForms, and WPF). There is also a testing project (Mvc.Tests). The bottom three layers each have their own individual projects.

The objective of the Spark platform is to allow developers to build beautiful applications quickly and easily. The Spark solution has just 4 projects, 2 of which are for REST and REST testing.

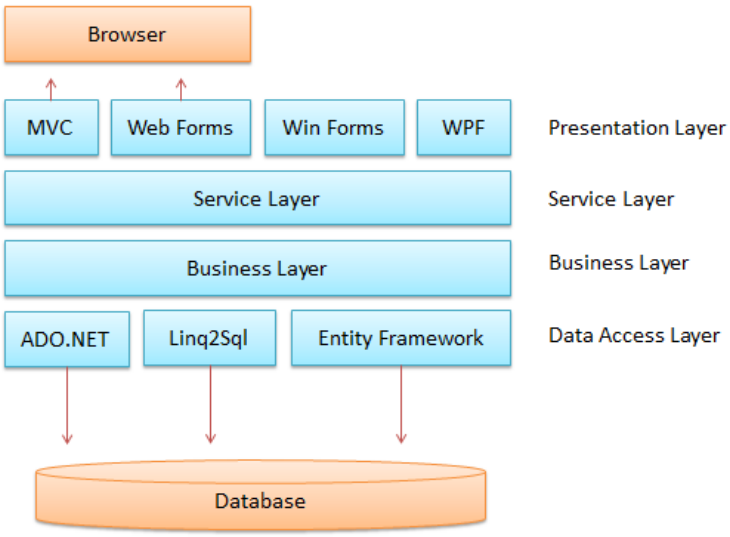


The main application, which is the Art Shop, has just 2 projects: Art.Web and Art.Domain. The Presentation layer is located in Art.Web and the remaining three layers, Service, Business, and Data Access, are all located in the Art.Domain project.

We'll look at layers next.

Step 3: Layers

To give you a head start here is an overview of the Patterns in Action architecture and layers.



It shows the 4 layers you commonly encounter in .NET app development: Presentation, Service, Business, and Data Access.

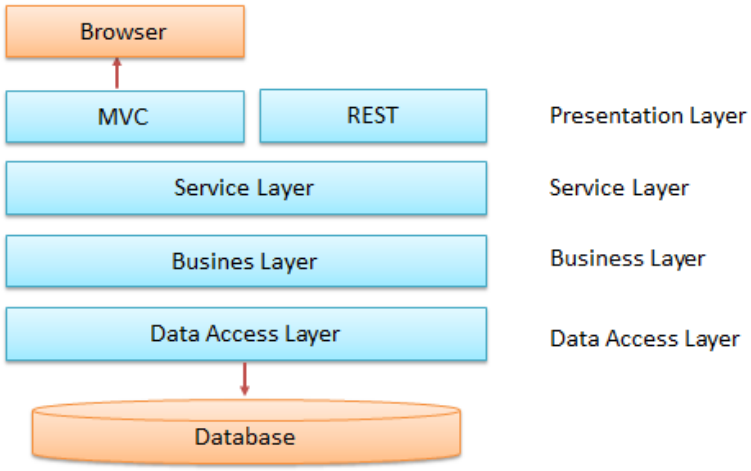
The Presentation layer support 4 different UI platform; they are MVC, WebForms, Windows Forms and WPF. Two of these are web apps that use the browser.

The diagram clearly shows that all 4 UI platforms must pass through the same Service layer to get data in and out of the database. This is a very important concept and clearly demonstrates the power and flexibility of pattern based architectures, that is, 4 different platforms all of which consume the same Service layer.

Three different data access technologies are available for data access: ADO.NET, Linq 2 Sql, and Entity Framework. They represent the Data Access Layer. All three talk to the same database.

In *Patterns in Action* any UI platform can be combined with any Data Access technology. For example you can combine MVC with ADO.NET, WPF with Entity Framework, or Windows Forms with Linq-to-Sql. Having the correct architecture and design patterns make all this possible.

Here is a similar diagram for the *Spark* platform:



It is somewhat simpler than Patterns in Action because its focus is on providing a simple environment that allows developers to quickly build lightweight and fast applications. Only a select set of patterns, practices, and tools are used. Notice that Spark also supports REST which allows the outside world to communicate with the application.

The documentation will explain all of the details.

Step 5: Data Flow

Once you have a feel of the different projects and layers, you need to understand the data flow or the execution path: in other words, how does the data get from the database all the way to the UI and then back again.

The debugger is a great tool to help in tracing the execution through the different layers. Here is how this works. Suppose your startup project is MVC. You start with a breakpoint in an action method. When the breakpoint hits, you trace the execution path down through the Service, Business and Data Access layers, and then all the way back up to the Presentation layer until the view is ready to be rendered. Do this for both GET and

POST requests, that is, requests where data is selected from the database and requests in which data is inserted or updated in the database.

By the time you understand the data flow and execution paths, you will have a very good understanding of the overall functionality and architecture of the applications.

Step 6: Design Patterns

At this point you are ready to explore the individual patterns that are scattered throughout the applications. These include GoF, Enterprise, and the MV* (Model View) patterns. Both applications offer you the opportunity to study literally dozens of patterns throughout the code. All of these are documented in the accompanying documents. Many are flagged in the code with a //** (double asterisk) comment (or ' ** in VB).

6. What is new in 4.5

.NET development is constantly changing and the latest release of this package is a reflection of the recent trend towards simplicity and rapid, agile development. Here is a summary of what's new in the Design Pattern Framework 4.5.

New in 4.5:

1. MVC 4 with Razor, Inversion of Control, Testing, and more.
2. REST with the new Web API
3. Repository and Unit Of Work patterns
4. Active Record and CQRS Pattern
5. Caching Practices with Lazy load
6. SimpleMembership Services
7. Spark™ application platform and Art Shop reference application
8. Emphasis on JavaScript and jQuery client development

Enhancements in 4.5:

1. All applications have new look and feel
2. Simpler and easier setup and configuration

3. Convention over Configuration
4. Use of LocalDb over Sql Express

Features dropped in 4.5:

1. WCF and SOA: replaced with Web API and REST
2. Membership: replaced with SimpleMembership
3. Silverlight app
4. MS Access support

Note: All prior versions of the Design Pattern Framework (1.1, 2.0, 3.5, and 4.0) are available to you for free (see your download page on dofactory.com). So if you need any of the above dropped items, they are available in version 4.0 -- please note that 4.0 requires Visual Studio 2010.

7. Summary

The *Design Pattern Framework 4.5* is a unique and comprehensive package that teaches you about design patterns and their use in real-world applications.

We are confident that you will be able to apply many of these patterns and practices in your own projects and build wonderful .NET applications!