

THE EXPERT'S VOICE® IN WINDOWS 8

Windows 8

MVVM Patterns Revealed

Covers both C# and JavaScript

*UNDERSTAND HOW MVVM
WORKS IN WINDOWS 8*

Ashish Ghoda with Jay Nanavaty

Apress®

Windows 8 MVVM Patterns Revealed

Covers both C# and JavaScript



Ashish Ghoda

Apress®

Windows 8 MVVM Patterns Revealed

Copyright © 2012 by Ashish Ghoda

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN 978-1-4302-4908-5

ISBN 978-1-4302-4909-2 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Ewan Buckingham

Technical Reviewer: Fabio Claudio Ferracchiati

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Louise Corrigan,

Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman,

James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick,

Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Anamika Panchoo

Copy Editor: Lori Cavanaugh

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail, rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

I dedicate this book to my grandparents (Nayansukhray and Kumud Ghoda, Mahavir and Sarala Majmudar), parents (Jitendra and Varsha Ghoda), sister (Kruti Vaishnav), and my lovely family (Pratixa, Gyan, and Anand Ghoda) whose blessings, sacrifice, continuous support, and encouragement enabled me to achieve this dream.

—Ashish Ghoda

Contents at a Glance

About the Author xi

About the Technical Reviewer xiii

Acknowledgments xv

Introduction xvii

■ Chapter 1: Setting Up Windows 8 Application Project..... 1

■ Chapter 2: Implementing the View 13

■ Chapter 3: Implementing the Model 47

■ Chapter 4: Implementing the ViewModel 59

■ Chapter 5: HTML5, JavaScript, and Windows 8 Applications 99

■ Chapter 6: View, Model, and ViewModel Structures in
HTML5 and JavaScript..... 109

Index..... 145

Contents

- About the Author xi
- About the Technical Reviewer xiii
- Acknowledgments xv
- Introduction xvii
- Chapter 1: Setting Up Windows 8 Application Project..... 1
 - The Development Environment 1
 - The Windows 8 Operating System..... 1
 - The Visual Studio Development Platform 2
 - Data Storage..... 3
 - Developer License 3
 - The Windows Store Project Templates 3
 - Creating a Windows Store XAML Project – FinanceHub..... 4
 - Exploring FinanceHub Project 5
 - The Project Structure..... 6
 - The Package Application Manifest File..... 7
 - Setting MVVM Architecture in FinanceHub Project 10
 - Summary..... 11
- Chapter 2: Implementing the View 13
 - Customizing Deployment Package and Runtime Configuration Properties 13
 - Background Color of the Tile 14
 - Application Logo Files and Splash Screen..... 14

Enabling Customization of Application Styles and Themes.....	15
Adding New Resource Dictionary File	16
Enabling Additional Resources.xaml Resource Dictionary at Application Level	17
Customizing Application Theme	17
Developing User Interface	20
The Main Startup Page – MainPage.xaml.....	20
Adding Bottom Application Bar – MainPage.xaml	21
Setting Up Remaining Custom Resources and Styles in Resources.xaml	24
Adding Stocks Page – StocksPage.xaml	25
Adding Stock Details Page – StockInfoView.xaml and StockDetails.xaml.....	32
Adding Add Stock Flyout Control	39
Adding Remove Stock Flyout Control	40
Integrating Flyout Controls with Bottom Application Bar.....	40
Summary.....	45
■ Chapter 3: Implementing the Model	47
Define Serializable and Deserializable Stocks Data Model	48
Define enum to support Add and Remove Stocks Actions	51
Define a Class for Event Arguments of Add and Remove Stocks Actions.....	51
Create a Helper Class to Store and Retrieve Stocks Watchlist in the Local Data Storage	52
Asynchronous Operations and Local Storage Capabilities for Windows 8 Applications.....	52
Implementing Local Storage Helper Class.....	54
Add SimulatedRandomStocksDetail.csv File.....	56
Summary.....	57

■ Chapter 4: Implementing the ViewModel	59
Building MVVM Framework for FinanceHub Application	59
IoC Container Dependency	60
The EventAggregator Class	61
<i>The Navigation Service</i>	65
Implementing the ViewModelLocator Class.....	68
Implementing the <i>DelegateCommand</i> Class.....	69
Implementing ViewModel Classes.....	71
The StocksPageViewModel ViewModel	72
The AddRemoveStockViewModel ViewModel	89
Summary	98
■ Chapter 5: HTML5, JavaScript, and Windows 8 Applications	99
HTML5 and JavaScript	99
Windows 8 Applications Using HTML5 and JavaScript	101
Windows APIs for Windows 8 Store Applications	101
Windows JavaScript Library	101
Best Practices in JavaScript Development.....	103
Global Namespace Pollution and Prevention	103
Use of Self	104
Using the Strict Mode	104
Module Pattern and Self-Executing Functions	105
Using Constructor Initialization.....	105
The Knockout JavaScript Framework	106
Summary	107

■ Chapter 6: View, Model, and ViewModel Structures in HTML5 and JavaScript.....	109
Setting up FinanceHub JavaScript Windows 8 Application Project.....	109
Exploring and Setting up FinanceHub Project	110
Adding Application Logo Files and Splash Screen.....	111
Implementing the View	111
Adding StockPages.html Page Control	112
Adding StockDetails.html Page Control.....	114
Enhancing default.html Page.....	118
Implementing Converters	121
Create FinanceHub Application Branding	121
Implementing the Model	122
Define Stock Class.....	123
Add SimulatedRandomStocksDetail.csv File	124
Implementing the ViewModel.....	125
Define StocksPageViewModel Class and Bind to the Views.....	125
Implementing Local Storage Helper Class.....	132
Define AddandRemoveStockViewModel Class and Bind to the Views.....	133
Implementing the State Persistence	141
The SaveStocks Function	141
The LoadStocks Function	142
Check State Persistence.....	143
Summary	144
Index.....	145

About the Author



Ashish Ghoda has been awarded a British Computer Society (BCS) Fellowship and is a senior IT executive with over 15 years IT leadership experience, enterprise architecture, application development, and technical and financial management.

He is a director at a Big Four auditing firm and teaches at New Jersey Institute of Technology (NJIT) and University of Maryland University College (UMUC) as adjunct assistant professor/faculty. He is also founder and president of Technology Opinion LLC providing IT services to organizations and IT community.

Ashish Ghoda is an author/co-author and technical reviewer of multiple books and articles on Microsoft platforms including *XAML Developer Reference* (Microsoft Press) and *Introducing Silverlight 4* (Apress).

He has a master's degree in information systems from NJIT and has earned Microsoft and TOGAF certifications.

About the Technical Reviewer

Fabio Claudio Ferracchiati is a senior consultant and a senior analyst/developer using Microsoft technologies. He works for Brain Force (www.brainforce.com) in its Italian branch (www.brainforce.it). He is a Microsoft Certified Solution Developer for .NET, a Microsoft Certified Application Developer for .NET, a Microsoft Certified Professional, and a prolific author and technical reviewer. Over the past ten years, he's written articles for Italian and international magazines and coauthored more than ten books on a variety of computer topics.

Acknowledgments

I would like to thank Ewan Buckingham, the lead editor of this book, who has given me opportunity to write this fast-paced guide for the new Microsoft Windows 8 application development platform.

It was a journey for me, as I learned the new concepts while writing this book and there were some unpredicted events like super storm Sandy that caused some delays. However, the support from the coordinating editor, Anamika Panchoo, and excellent feedback from the technical reviewer, Fabio Claudio Ferracchiati, and Ewan helped me to finish this book to a high standard and on time. I also would like to thank you the whole Apress team for their support during different phases of this book writing project.

Jay Nanavaty has been working with me since 2008 on different projects, including working on parts of my earlier books. This time I am happy to see him growing and becoming a significant contributor to this book. I am looking forward to seeing him becoming a co-author of my future books. I wish him good luck in his future success.

Introduction

Microsoft Windows 8 provides a new versatile platform to develop Windows 8 store applications with the “modern” touch-optimized user interface concepts running on different set of Windows 8 devices—PCs, tablets, and mobile phones.

This book provides step-by-step instructions for developing a FinanceHub Windows 8 store application using XAML and C# and HTML5, JavaScript, and CSS3 following the Model-View-View-Model (MVVM) design pattern.

This book is a fast-paced guide for how to create Windows 8 apps for PCs, tablets, and mobile phones. Whether you use C# or JavaScript, XAML, or HTML5 and CSS3, this book teaches you how to use the MVVM pattern to bring elegance, power, speed, and reusability to your Windows 8 apps. Experience in XAML and C# or HTML5, CSS3, and JavaScript would be helpful learn these concepts quickly.

Before we dive in to developing the FinanceHub Windows 8 application, let’s quickly get an overview of the following:

- Model-View-View-Model (MVVM) pattern
- Windows 8 store applications
- The FinanceHub Windows 8 store application, which we are going to develop as part of this book
- The book structure

The Model-View-View-Model (MVVM) Pattern

We have established design patterns such as Model View Controller (MVC) and Model View Presenter (MVP) patterns that enable the application design to provide separation between what an application looks like (the user interface look and feel), and how the application acts on user actions (implementing the business logic and integrating with the data source). With the introduction of XAML-based frameworks and their capabilities of data binding, data templates, commands, and behaviors, an enhanced MVVM design pattern emerged, which can leverage the aforementioned XAML-based frameworks capabilities and can provide a separation between the presentation and business layers.

The MVVM pattern is mainly derived from the concept of MVC design pattern. Thus, if you are familiar with the MVC design pattern then you will see many similarities in the MVVM and MVC design patterns concepts. The MVVM pattern creates separation between these layers by allowing the user to define and implement clear role and responsibilities for the presentation and business layers and perform integration between these layers.

Figure 1 demonstrates the three main core classes—the View Class, the ViewModel Class and the Model class—of the MVVM design pattern.

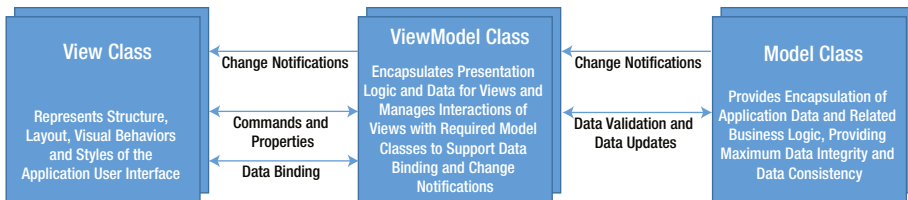


Figure 1. Core Classes of the MVVM Design Pattern

- **View Class**—defines the applications' user interface. It represents the structure, layout, visual behaviors, and styles of the application user interface. For Windows 8 store applications you can define Views of your application in XAML or HTML5. The application can contain one or more View classes as per the application requirements and application design.
- **ViewModel Class**—encapsulates the presentation logic and data for the view. It manages the interactions of views with the required model classes to support data binding and the change notification events by implementing required properties and commands in the ViewModel. The ViewModel class(es) do not contain any user interface. Application can contain one or more ViewModel classes as per the application requirements and application design. We will implement ViewModel classes required for the FinanceHub application, to be developed as part of this book, using C# (for the XAML-based views) and JavaScript (for HTML5-based views).
- **Model Class**—provides encapsulation of the application data and related business logic, allowing for maximum data integrity and data consistency. Basically it implements application data model, integrates with the data sources, and implements required business logic for data retrieval and validation in order to provide data integrity and data consistency. The Model class(es) do not contain any user interface. The application can contain one or more Model classes as per the application requirements and application design. We will implement Model classes required for the FinanceHub application, to be developed as part of this book, using C# (for the XAML-based views) and JavaScript (for HTML5-based views).

Windows 8 Store Applications

Microsoft released the latest version of the operating system Windows 8, which is designed to provide the next generation touch-optimized applications that can run seamlessly on different Windows devices from PCs and tablets, to Windows 8 mobile phones supporting different form factors. These touch-optimized, next-generation Windows 8 applications are called Windows 8 Store Applications.

The Windows 8 platform introduces a new Windows Runtime (WinRT) framework that introduces the new set of APIs and user controls for XAML and JavaScript (WinJS) to support Windows 8 store applications development. Windows 8 platform also introduces new user interface concepts, such as Charm bar and App bar, which should be considered as part of your application design.

The Windows 8 development platform enables you to develop Windows 8 store applications not only using .NET languages such as Visual Basic .NET and C# and XAML to define the presentation layer but also using more generic languages such as C++ and JavaScript and HTML5 and CSS3 to define the presentation layer. You must install Windows 8 and Visual Studio 2012 to develop Windows 8 applications.

Please note that in this book I will be using “Windows 8 store application” and “Windows 8 application” words interchangeably.

The FinanceHub Windows 8 Store Application

We are going to develop a FinanceHub Windows 8 store application using XAML and C# and HTML5, CSS3 and JavaScript using Visual Studio Windows store default XAML and JavaScript application project templates to demonstrate how to best implement MVVM design pattern for Windows 8 apps.

The FinanceHub application allows you to add/remove stocks to create a stocks watchlist to monitor your favorite stocks. The application contains two main views:

- The first view is a home page, showing your favorite stocks with the latest updates in the stock price.
- The second view is a stock details view, which provides the detailed information of the selected stock.

It will also implement a Windows 8 appbar to allow add and remove one or more stocks from your watchlist.

Figure 2 shows the home page and Figure 3 shows the stocks details page of the FinanceHub Windows 8 application.

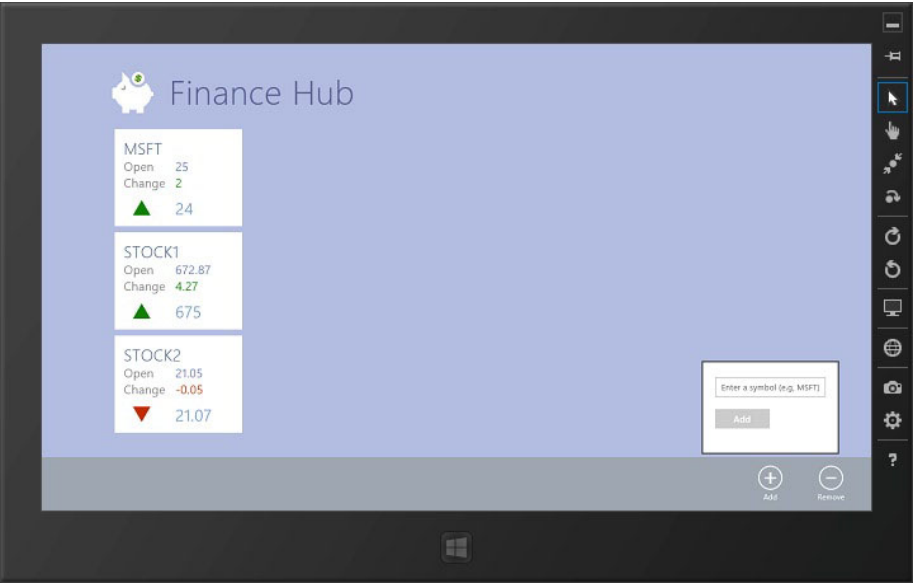


Figure 2. Home Page of the FinanceHub Windows 8 store application displaying the stocks watchlist and an appbar with Add Stock flyout

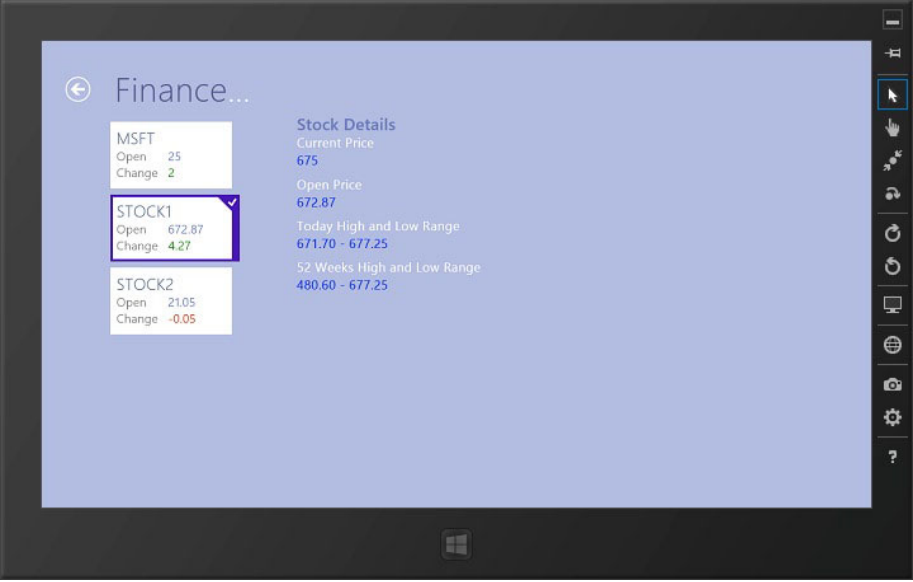


Figure 3. Stock Details Page of the FinanceHub Windows 8 store application displaying selected stock information

The Book Structure

This book contains six chapters.

- The first four chapters are focused on developing the FinanceHub Windows 8 application using XAML and C# following MVVM design pattern.
- Chapters 5 and 6 will describe how to develop the FinanceHub Windows 8 application using HTML5, CSS3, and JavaScript following MVVM design pattern.

Chapter 1: Setting Up Windows 8 Application Project

The first chapter provides information on how to set up your development environment to start the Windows 8 application development. It will also create a skeleton of the FinanceHub XAML Windows 8 application that would require the user to follow and implement the MVVM design pattern.

Chapter 2: Implementing the View

The second chapter will implement required view classes to build the user interface (home page, stock details view, appbar, and add and remove stocks view) of the FinanceHub application following the hierarchical navigation pattern. Along with building the user interface, we will explore some of the key enhancements made in the XAML and some of the new properties introduced in WinRT.

Chapter 3: Implementing the Model

The third chapter will implement lightweight model classes that defined the serializable and deserializable stocks data model and will also implement pre-requisites to implement features of Add and Remove stock actions. During the implementation we will explore and use some of the new features introduced in the .NET Framework 4.5.

Chapter 4: Implementing the ViewModel

The fourth chapter will implement the ViewModel classes to support stocks information binding to StocksPage, StockDetails, and RemoveStock views, and application navigation. It will also implement action commands to support stock selection and add and remove stock actions. We will also revisit the earlier implementation of the project to remove the hard-coded information and bind it to the ViewModel properties and commands.