Visual Basic 2015
Made Easy

Dr.Liew

Disclaimer

Visual Basic 2015 Made Easy is an independent publication and is not affiliated with, nor has it been authorized, sponsored, or otherwise approved by Microsoft Corporation.

Trademarks

Microsoft, Visual Basic, Excel and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks belong to their respective owners.

Liability

The purpose of this book is to provide basic guides for people interested in Visual Basic 2015 programming. Although every effort and care has been taken to make The information as accurate as possible, the author shall not be liable for any error, Harm or damage arising from using the instructions given in this book.

Acknowledgement

I would like to express my sincere gratitude to many people who have made their contributions in one way or another to the successful publication of this book.
My special thanks go to my children Xiang, Yi and Xun who have contributed their ideas and help in editing this book. I would also like to appreciate the support provided by my beloved wife Kim Huang and my youngest daughter Yuan. I would also like to thank the millions of readers who have visited my **Visual Basic Tutorial** website at **vbtutor.net** for their support and encouragement.

About the Author

Dr. Liew Voon Kiong holds a bachelor's degree in Mathematics, a master's degree in Management and a doctorate in Business Administration. He has been involved in Visual Basic programming for more than 20 years. He created the popular online Visual Basic Tutorial at **www.vbtutor.net**, which has attracted millions of visitors since 1996. It has consistently been one of the highest ranked Visual Basic websites.

To provide more support for Visual Basic students, teachers, and hobbyists, Dr. Liew has written this book to complement the free Visual Basic 2015 tutorial with much more content. He is also the author of the Visual Basic Made Easy series, which includes **Visual Basic 6 Made Easy, Visual Basic 2008 Made Easy, Visual Basic 2010 Made Easy, Visual Basic 2013 Made Easy, Visual Basic 2017 Made Easy** and **Excel VBA Made Easy.** Dr. Liew's books have been used in high school and university computer science courses all over the world.

Chapter 1
Introduction to Visual Basic 2015

- A brief description of Visual Basic 2015
- Getting to know the Visual Basic 2015 Integrated Development Environment

1.1 A Brief Description of Visual Basic 2015

Visual Basic is a third-generation event-driven programming language first released by Microsoft in 1991. The final version of the classic Visual Basic was Visual Basic 6. Visual Basic 6 is a user-friendly programming language designed for beginners. Therefore, It enables anyone to develop GUI Windows applications easily. Many developers still favor VB6 over its successor VB.NET.

In 2002, Microsoft released Visual Basic.NET(VB.NET) to replace Visual Basic 6. Thereafter, Microsoft declared VB6 a legacy programming language in 2008. However, Microsoft still provides some form of support for VB6. VB.NET is a fully object-oriented programming language implemented in the .NET Framework. It was created to cater for the development of the web as well as mobile applications. Subsequently, Microsoft has released many versions of VB.NET. They are Visual Basic 2005, Visual Basic 2008, Visual Basic 2010, Visual Basic 2012, Visual Basic 2013, Visual Basic 2015 and Visual Basic 2017. Although the .NET portion was discarded in 2005, all versions of the Visual Basic programming language released since 2002 are regarded as VB.NET programming language

Visual Basic 2015 was released in the year 2015. It comes as a .NET desktop development component of the Visual Studio Community 2015 integrated development environment(IDE). It is used to build windows desktop applications using the .NET framework. Besides that, Visual Studio Community 2015 also comes with other Windows development tools that include Visual C#, Visual F# , Visual C++, JavaScript, SQL Server, Python , Unity Game and many more.

You can download the free version of Visual Studio Express 2015 for Windows Desktop from the following link:
https://www.visualstudio.com/vs/older-downloads/

After clicking the link about, you will be asked to enter your Microsoft username and password. Next, the following window will appear, as shown in Figure 1.1

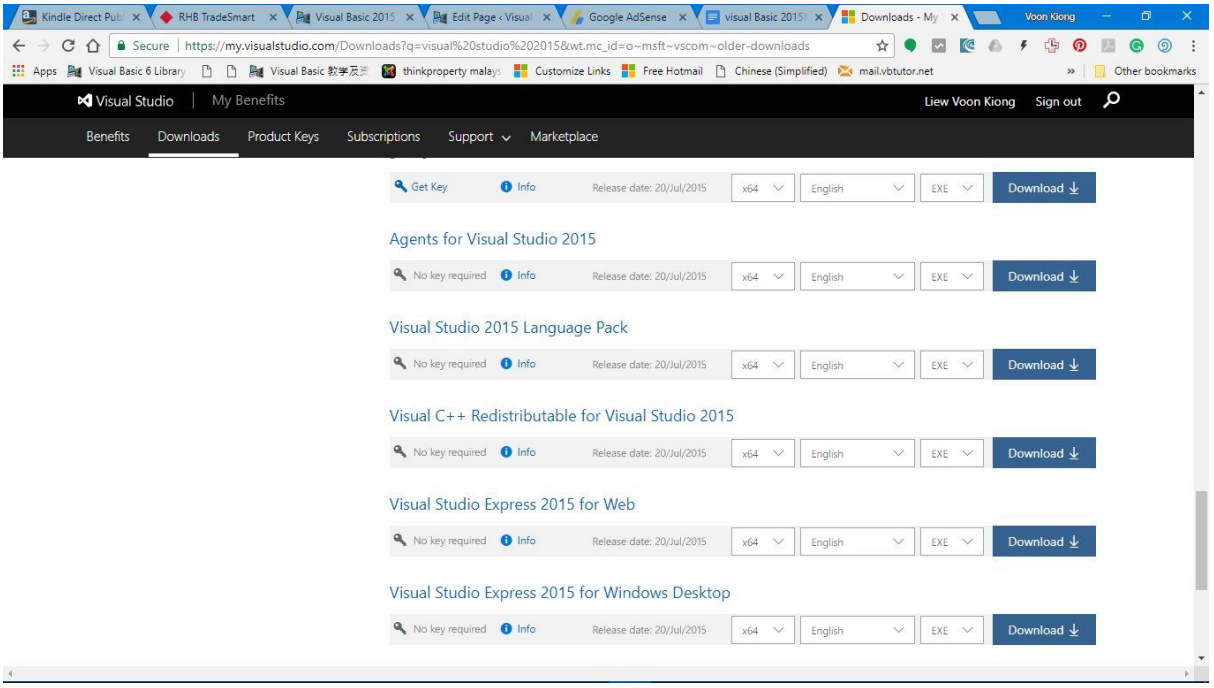

Figure 1.1

Select Visual Studio Express 2015 for Windows Desktop and then click the download button. After the installation file is downloaded, click on it to install VS Express 2015.

1.2 The Visual Studio 2015 Integrated Development Environment

When you launch Microsoft Visual Studio 2015 Express, you will be presented with the Start Page of Microsoft Visual Studio Community 2015, as shown in Figure 1.2

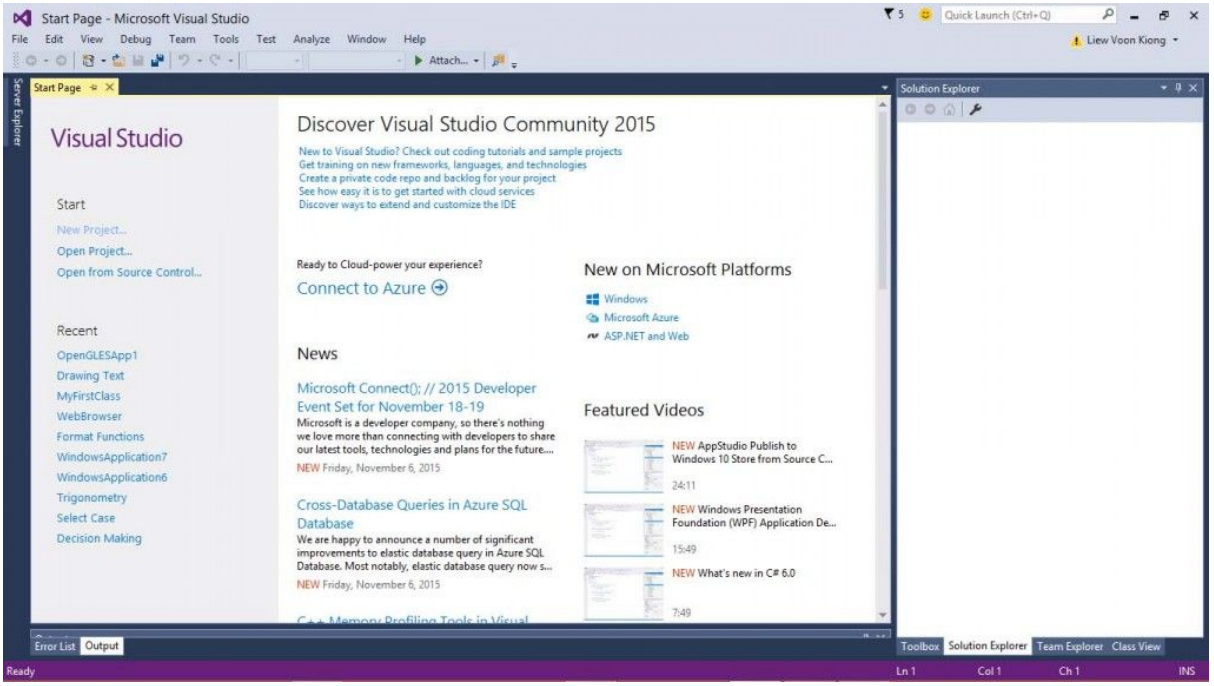

Figure 1.2 Visual Studio Express 2015 IDE

The Visual Studio 2015 start page comprises a few sections, the Start section and the Recent section on the left pane and Discover Visual Studio Community 2015 on the right pane. In the start page, you can either start a new project, open a project or open a recent project. You can also check for the latest news in Visual Studio Community 2015. The Start Page also consists of a menu bar and a toolbar where you can perform various tasks by clicking the menu items.

1.3 Creating a New Project in Visual Studio 2015

To start a new Visual Studio Express 2015 project, click on New Project under the Start section to launch the Visual Studio 2015 New Project page as shown in Figure 1.3. You can also choose to open a recent project:

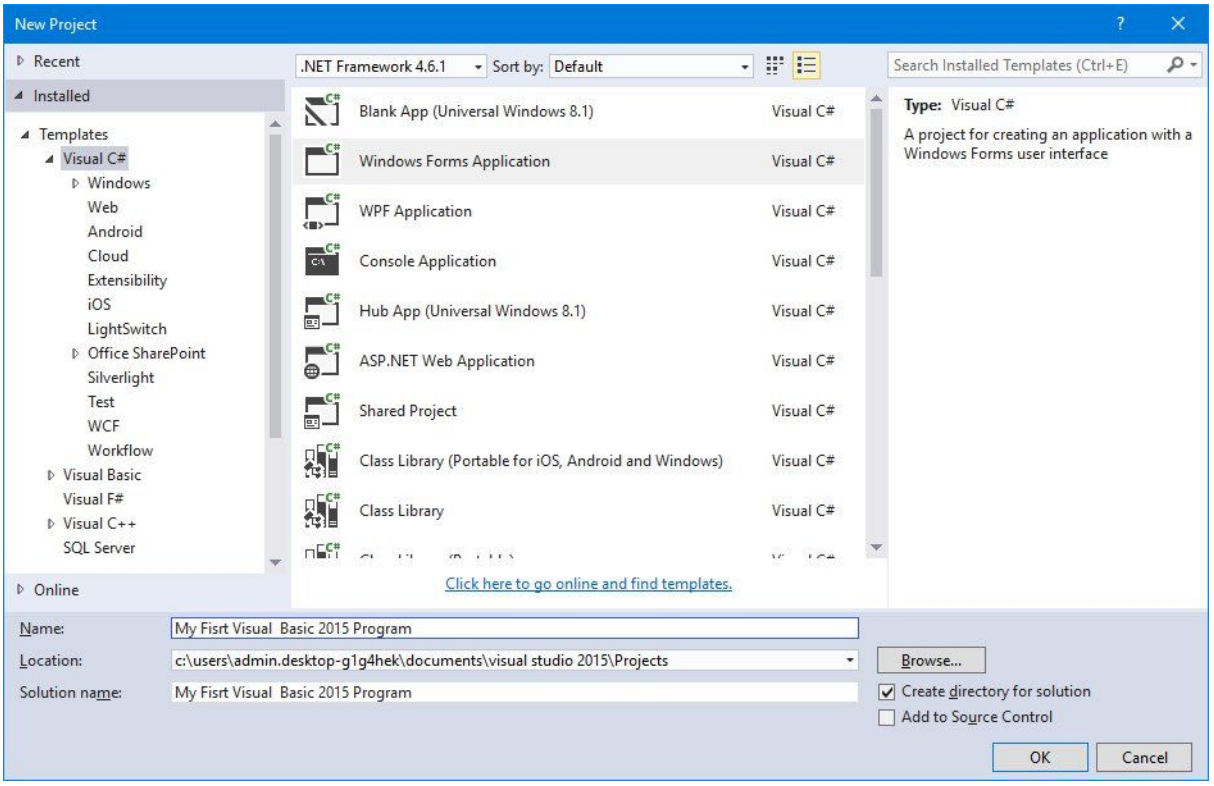

Figure 1.3 Visual Studio 2015 Express New Project Page

The New Project Page comprises a few templates, among them are Visual Basic, Visual C# , Visual C++ and others. Since we are only learning Visual Basic 2015, we shall select Visual Basic. Visual Basic 2015 offers you several types of projects that you can create; they are *Blank Apps*, *Windows Forms Application, WPF Application, Console Application ,Class Library(.NET Framework), Shared Project* and more. Since we are only learning how to create windows desktop applications, we shall select Windows Forms Application.

At the bottom of this dialog box, you can change the default project name WindowsApplication1 to some other name you like, for example, My First Visual Basic 2015 Program. After renaming the project, click OK to continue. The Visual Basic Express 2015 IDE Windows will appear, as shown in Figure 1.4. Visual Basic Express 2015 IDE comprises a few windows, the Form window, the Solution Explorer window and the Properties window. It also consists of a toolbox which contains many useful controls that allows a programmer to develop his or her Visual Basic 2015 programs.

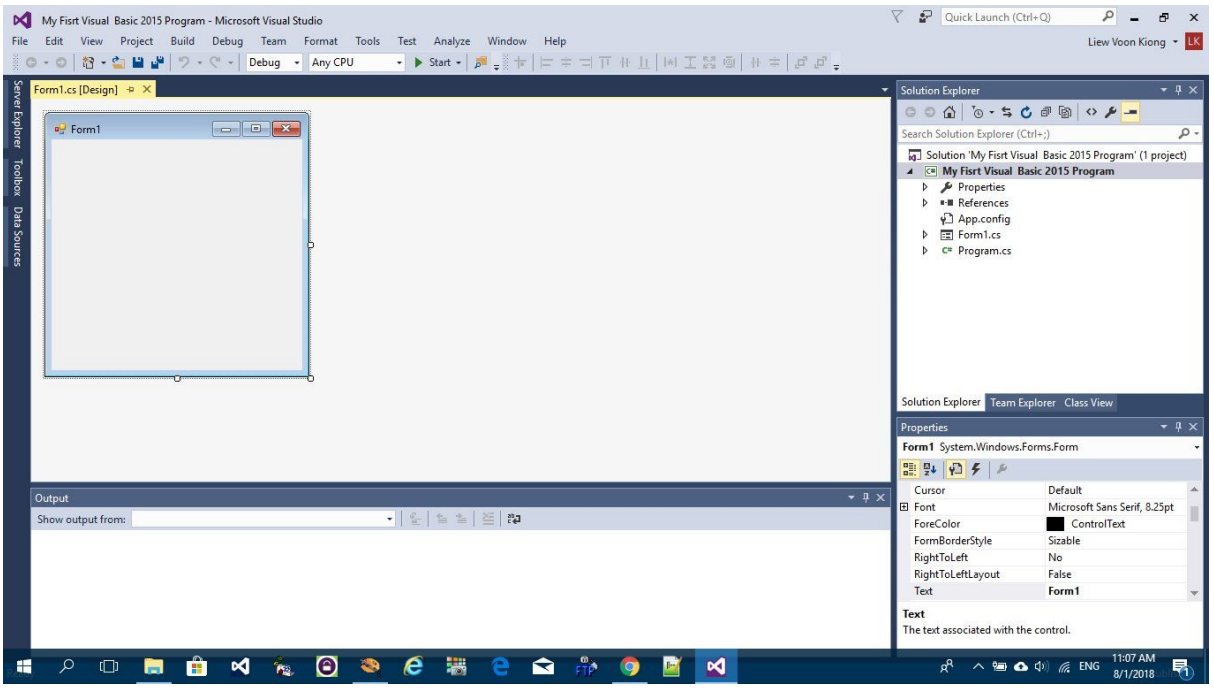

Figure 1.4 The Visual Basic 2015 Express IDE

The Toolbox is not shown until you click on the Toolbox tab. When you click on the Toolbox tab or use the shortcut keys Ctrl+Alt+x, the common controls Toolbox will appear, as shown in Figure 1.5. You can drag and move your toolbox around and dock it to the right, left ,top or bottom of the IDE.

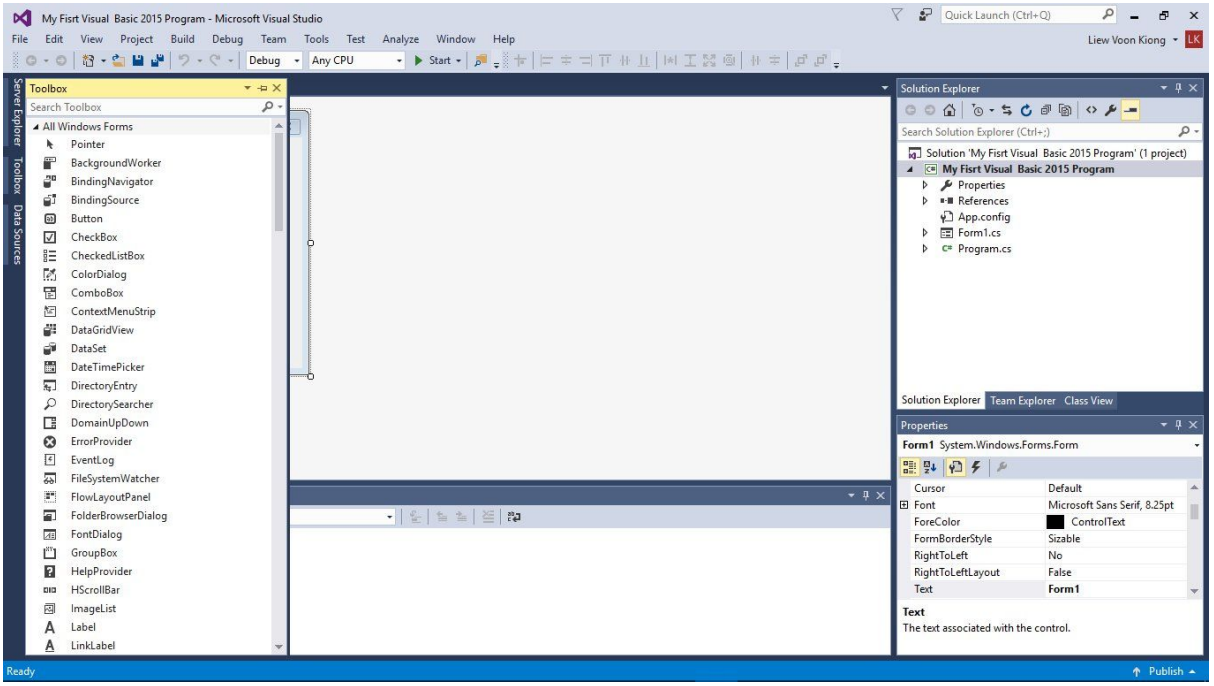

Figure 1.5 Visual Basic 2015 Express Tool Box

Next, we shall proceed to show you how to create your first program. First, change the text of the form to 'My First Visual Basic 2015 Program' in the properties window; it will appear as the title of the program. Next, insert a button and change its text to OK and its name to BtnOK. The design interface is shown in Figure 1.6

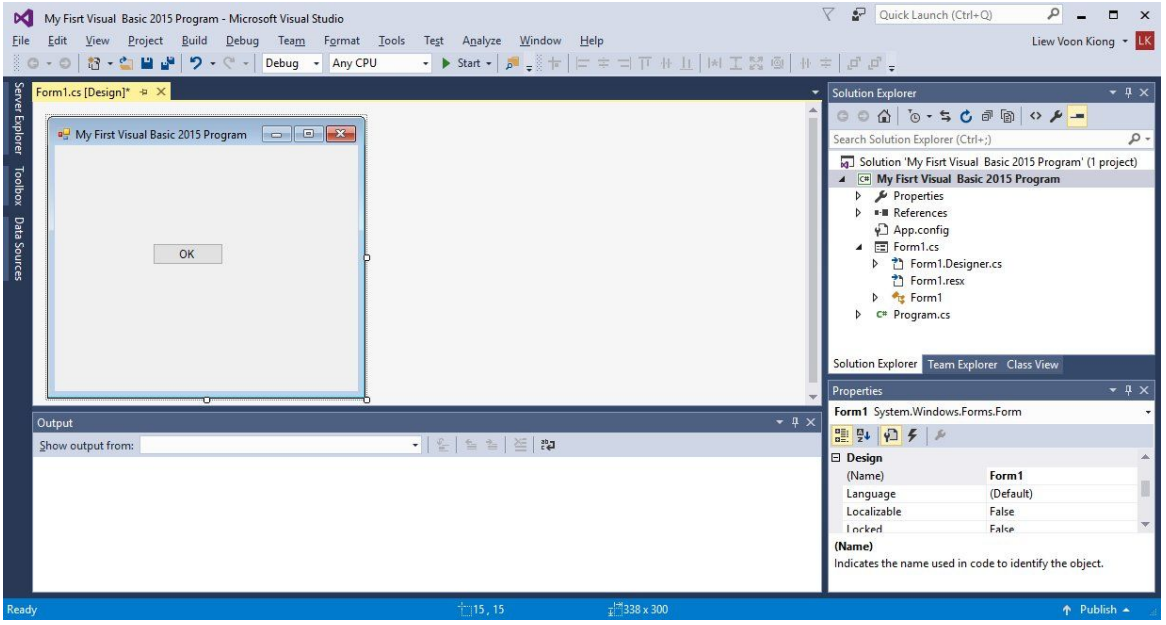

Figure 1.6 The Design Interface

Now click on the OK button to bring up the code window and enter the following statement between **`Private Sub`** and **`End Sub`** procedure, as shown in Figure 1.6:

```
MsgBox("My First Visual Basic 2015 Program")
```

Now click on the Start button on the toolbar or press F5 to run the program then click on the OK button, a dialog box that displays the "My First Visual Basic 2015 Program" message will appear, as shown in Figure 1.7. The function **`MsgBox`** is a built-in function of Visual Basic 2015 which can display the text enclosed within the brackets

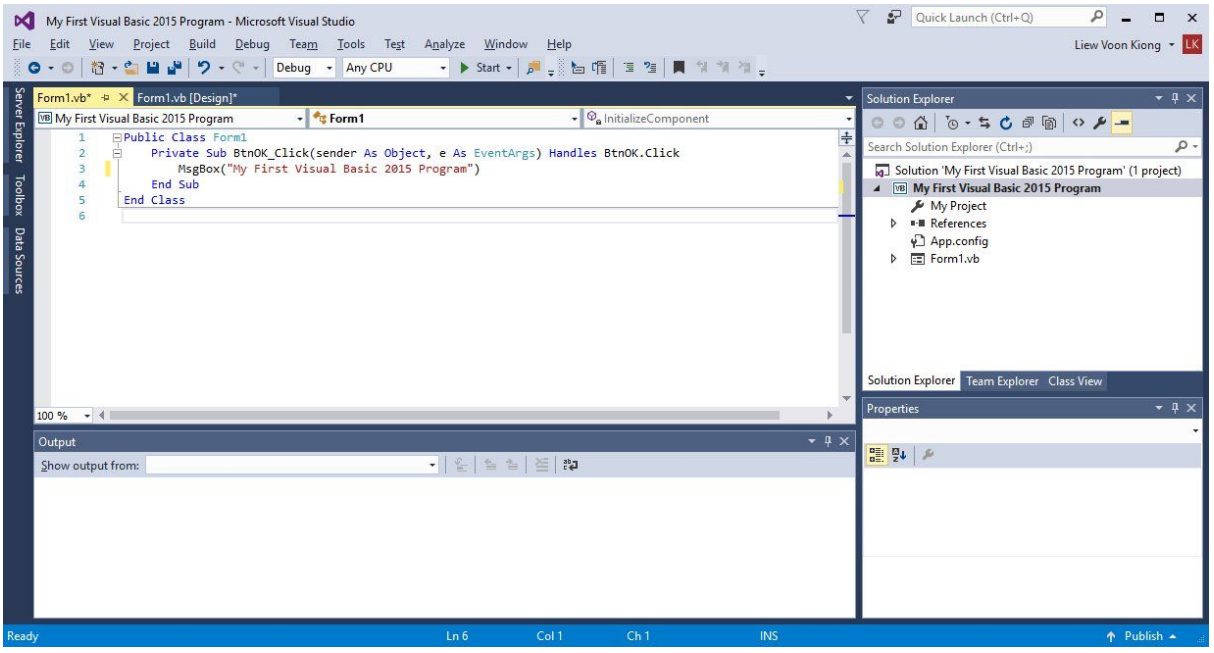

Figure 1.7 Visual Basic 2015 Code Window

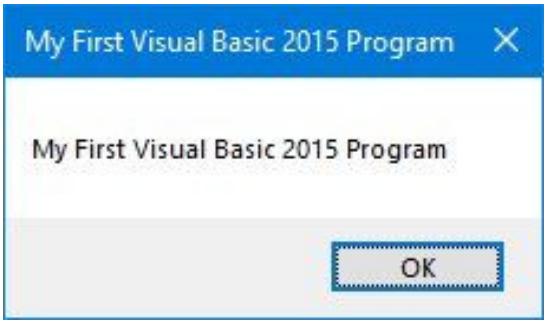

Figure 1.8 The Runtime Interface

Summary

- In section 1.1, you have learned about the history of Visual Basic 2015
- In section 1.2, you have learned how to install and launch Visual Basic Studio Express 2015
- In section 1.3, you have learned how to launch the new project dialog and the Visual Basic Express 2015 IDE. You have also learned how to write your first program.

Chapter 2
Designing the Interface

- Customizing
- Adding controls
- Setting Control Properties

As Visual Basic 2015 is a GUI-based programming language, the first step in developing an application is to build a graphical user interface. To build a graphical user interface, you need to customize the default form by changing its properties at design phase and at runtime. After customizing the default form, you may proceed to add controls from the toolbox to the form and then customize their properties.

2.1 Customizing the Form

When you start a new Visual Basic 2015 project, the VS2015 IDE will display the default form along with the Solution Explorer window and the Properties window for the form as shown in Figure 2.1. The name of the default form is Form1. The

properties window displays all the properties related to Form1 and their corresponding attributes or values. You can change the name of the form, the title of the form using the text property, the background color, the foreground color, the size and more. Try changing the following properties:

Property	**Value**
Name	MyForm
Text	My First Visual Basic 2015 Program
BackColor	Aqua
ForeColor	DarkBlue
MaximizeBox	False

In fact, you do not have to type in the color manually, you can indeed select a color from the color drop-down list that comprises three tabs, Custom, Web, and System, as shown in the Figure 2.1. Clicking on the drop-down arrow will bring out a color palette or a list of color rectangles where you can select a color.

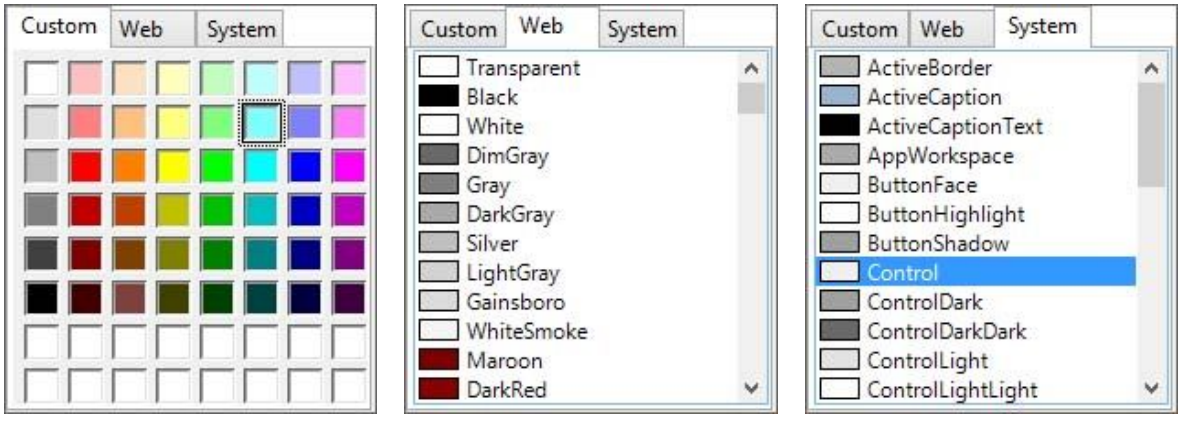

Figure 2.1

Another method of setting the colors is to manually type in the RGB color code or the hex color code. The values of R, G and B ranges from 0 to 255, therefore, by varying the values of the RGB we can obtain different colors. For example, a RGB value of 128, 255, 255 yield the cyan color.

On the other hand, the hex color code system use a six-digit, three-byte hexadecimal number to represent colors. The bytes represent the red, green and blue components of the color. One byte represents a number in the range 00 to FF (in hexadecimal notation), or 0 to 255 in decimal notation. For example, **#0000ff** represents the cyan color. However, when you type in the hex color code in the properties window of VS2015, it automatically converts the color to RGB color or the color name. Figure 2.2 shows a list of Hex color codes and the corresponding colors.

color	code	color	code	color	code	color	code	color	code
	eeeeee		ffffcc		ffccff		ff99ff		ff66ff
	dddddd		ffff99		ffcccc		ff99cc		ff66cc
	cccccc		ffff66		ffcc99		ff9999		ff6666
	bbbbbb		ffff33		ffcc66		ff9966		ff6633
	aaaaaa		ffff00		ffcc33		ff9933		ff6633
	999999		ccffff		ffcc00		ff9900		ff6600
	888888		ccffcc		ccccff		cc99ff		cc66ff
	777777		ccff99		cccccc		cc99cc		cc66cc
	666666		ccff66		cccc99		cc9999		cc6699
	555555		ccff33		cccc66		cc9966		cc6666
	444444		ccff00		cccc33		cc9933		cc6633
	333333		99ffff		cccc00		cc9900		cc6600
	222222		99ffcc		99ccff		9999ff		9966ff
	111111		99ff99		99cccc		9999cc		9966cc
	ff0000		99ff66		99cc99		999966		996699
	ee0000		99ff33		99cc33		999933		996633
	cc0000		66ffff		66ccff		6699ff		6666ff
	33ffff		00ffff		00ccff		3399ff		3366ff
	ff00ff		cc00ff		00ee00		0000ff		6600ff

Figure 2.2 Hex Color Codes

The design interface is shown in Figure 2.3 and the runtime interface is shown in Figure 2.4. In the runtime interface, notice that the title has been changed from Form1 to My First Visual Basic 2015 Program, background changed to aqua color , the text OK color is dark blue and the window cannot be maximized.

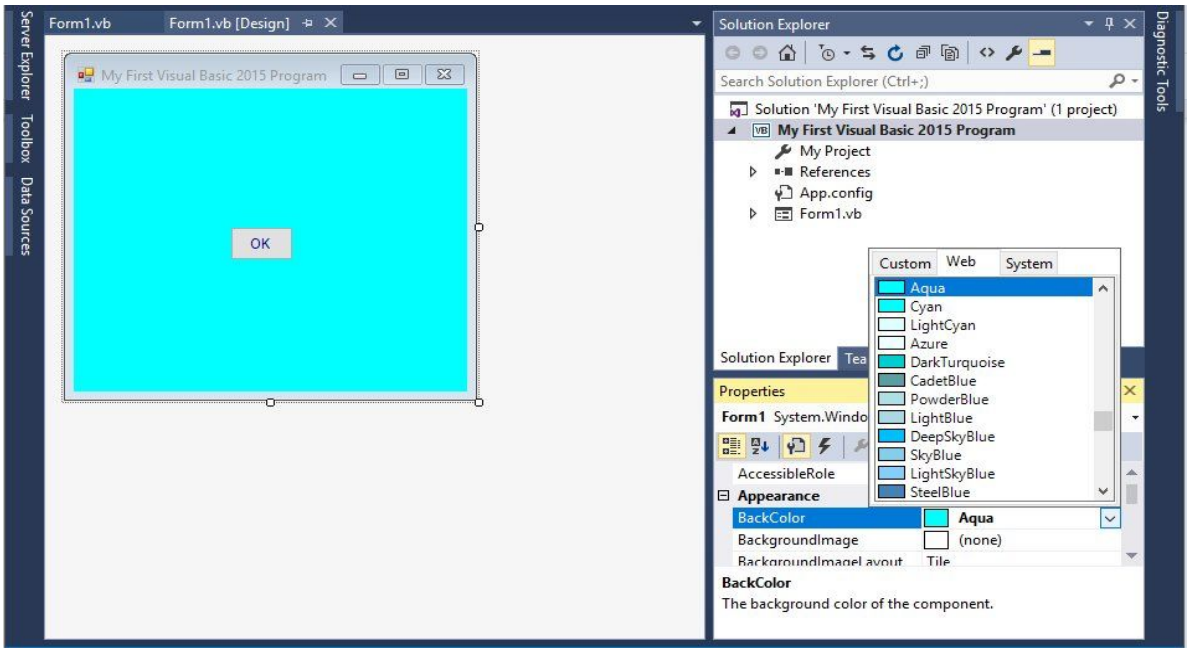

Figure 2.3

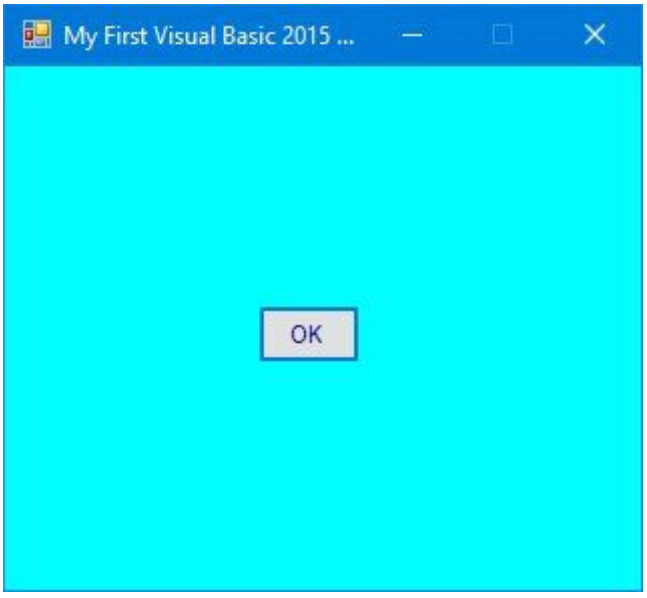

Figure 2.4

You can also change the properties of the form at run-time by writing the relevant codes. The default form is an object and an instant of the form can be denoted by the name **Me.** The property of the object can be defined by specifying the object's name followed by a dot or period:

```
ObjectName.property
```

For example, we can set the background of the form to blue using the following code:

```
Me.BackColor=Color.Blue
```

In addition, you can also use the **FromArgb** method to specify the color using the RGB codes, as follows:

```
Me.BackColor = Color.FromArgb(0, 255, 0)
```

To achieve the same interface as shown in Figure 2.3, type in the following code by clicking the form to enter the code window:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
Me.Text = "My First Visual Basic 2015 Program"
Me.BackColor = Color.Cyan
Me.MaximizeBox = False
Me.MinimizeBox = True
End Sub
```

In place of cyan, you can use RGB code as follows:

```
Me.BackColor = Color.FromArgb(0,255,255)
```

Press F5 to run the program and you will get the exact interface as that shown in Figure 2.4.

In addition, you can also specify the size, the opacity and the position of the default form using the code, as follows:

```
Private Sub Form1_Load(sender As Object, e As EventArgs Handles
```

```
MyBase.Load
Me.Text = "My First VB2015 Project"
Me.BackColor =Color.Beige
Me.MaximizeBox = False
Me.MinimizeBox = True
Me.Size = New Size(400, 400)
Me.Opacity = 0.85
Me.CenterToParent()
End Sub
```

The runtime interface is as shown in Figure 2.5. Notice that background is translucent as the Opacity is set to 0.85 or 85%.

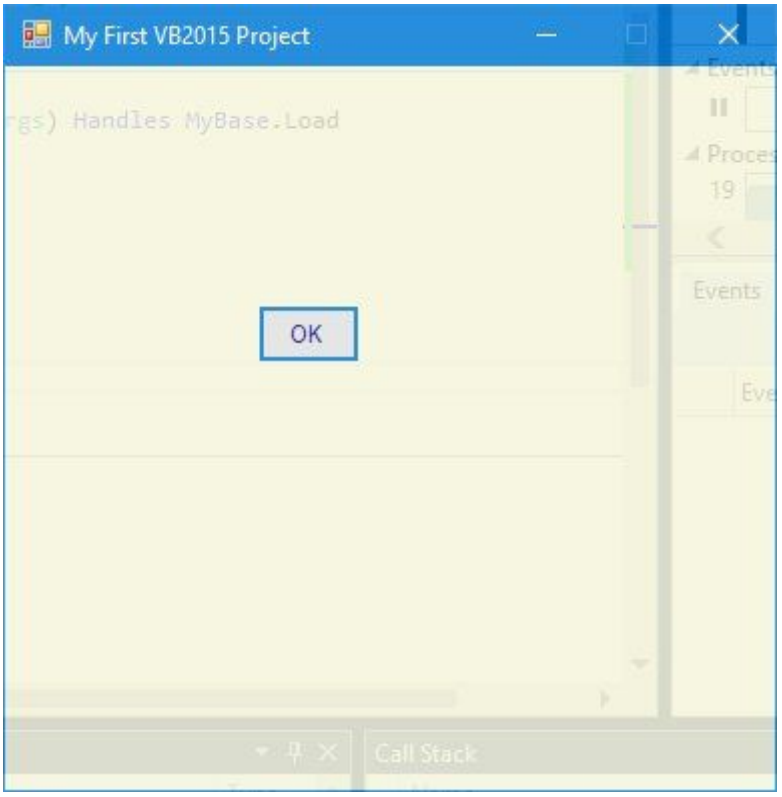

Figure 2.5

2.2 Adding Controls to the Form

In section 2.1, we have learned how to build an initial interface in Visual Basic 2015 by customizing the default form. Next, we shall continue to build the interface by adding some controls to the form. The controls are objects that consist of three elements, namely properties, methods, and events.They can be added to the form from the Toolbox. Among the controls, the most common ones are the button, label, textbox, listbox, combobox, picture box, checkbox, radio button and more. The controls can be made visible or invisible at runtime. However, some controls will only run in the background and cannot be seen at runtime, one such control is the timer.

The Toolbox is usually hidden when you start Visual Basic 2015 IDE, you need to click View on the menu bar and then select Toolbox to reveal the tool box, as shown in Figure 2.6. You can also use shortcut keys Ctrl+w+x to bring out the toolbox.

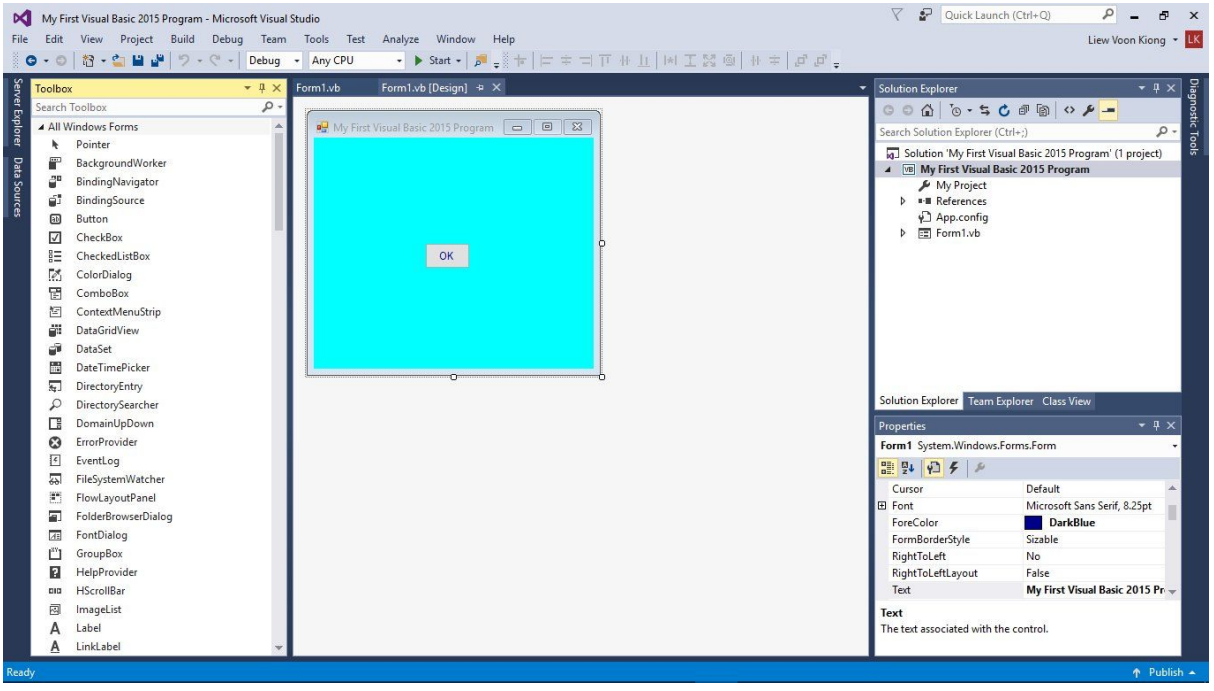

Figure 2.6: Toolbox

You can position the toolbox by dragging it anywhere you like while its status is set to float. You can also dock the toolbox by right-clicking on the tool box and choose dock from the pop-up menu. The docked Toolbox that appears side by side with the Solution Explorer, and as one of the tabbed windows together with the Form Design window and the code window, as shown in Figure 2.7.

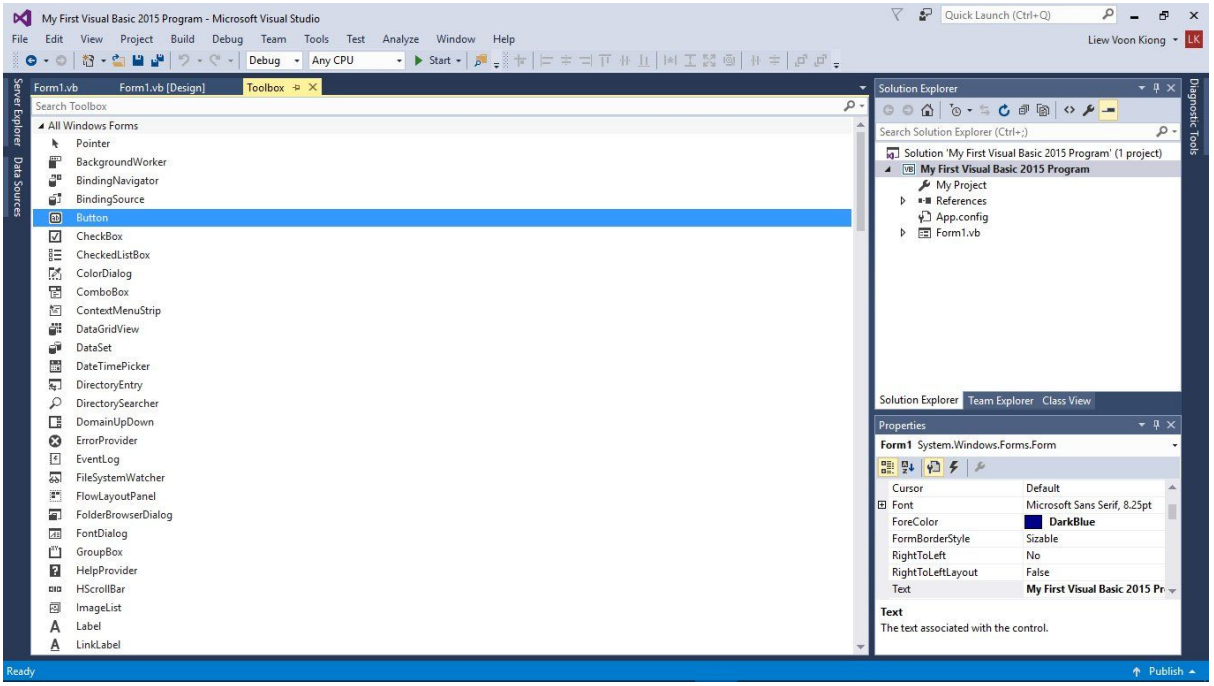

Figure 2.7

You can also dock the tool box at the bottom, below the default form, as shown in Figure 2.8. Further, you may also pin the tool box to the side bar or the bottom bar by clicking on the pin icon on the menu bar of the toolbox.

How and where you want to position your tool box is entirely up to you but we strongly suggest that you place the tool box alongside or at the bottom of the default form so that it is easy for you to add controls from the tool box into the form. You should never cover the form with the toolbox because it will be difficult to add controls to the form.

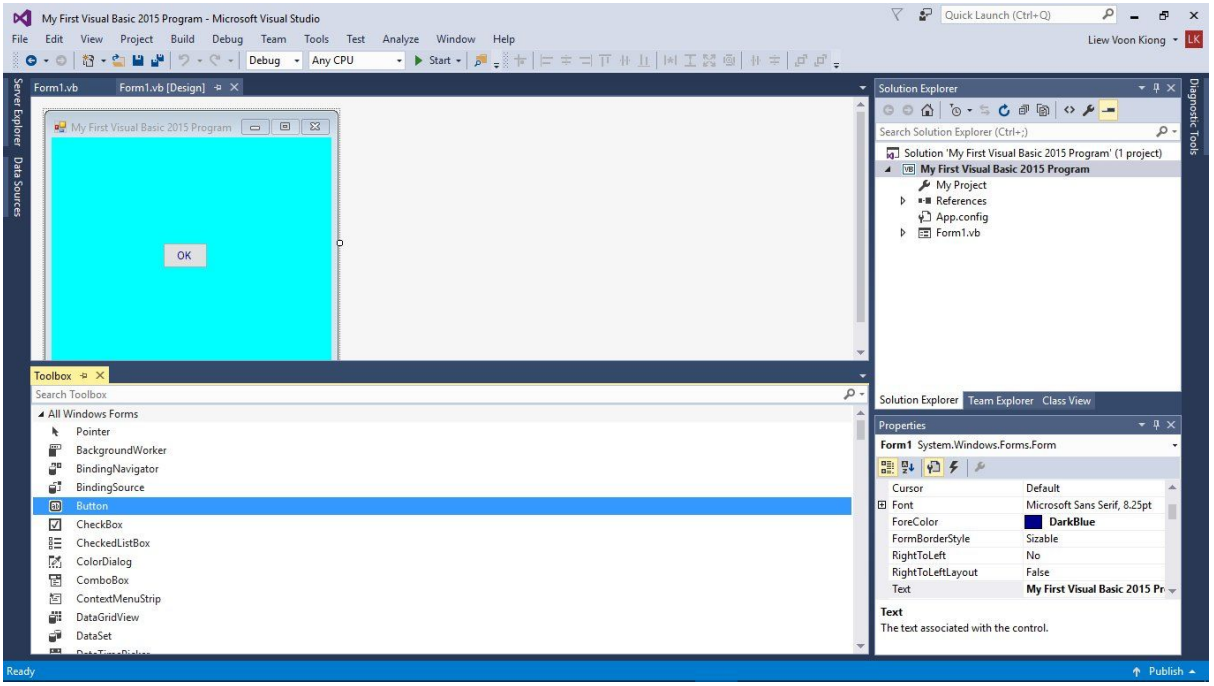

Figure 2.8

Adding a control to the form is an easy task, what you need to do is double click it or drag it onto the form. You can drag the control around in the form and you can also resize it.

To demonstrate how to add the controls and then change their properties, we shall design a picture viewer. First, change the title of the default form to Picture Viewer in its properties window. Next, insert a picture box on the form and change its background color to white. To do this, right click the picture box and select properties in the popup menu, then look for the **`BackColor`** Property as shown in the properties window in Figure 2.9. Finally, add two buttons to the form and change the text to View and Close in their respective properties windows. The picture viewer is not functional yet until we write code for responding to events triggered by the user. We will deal with the programming part in the coming chapters.

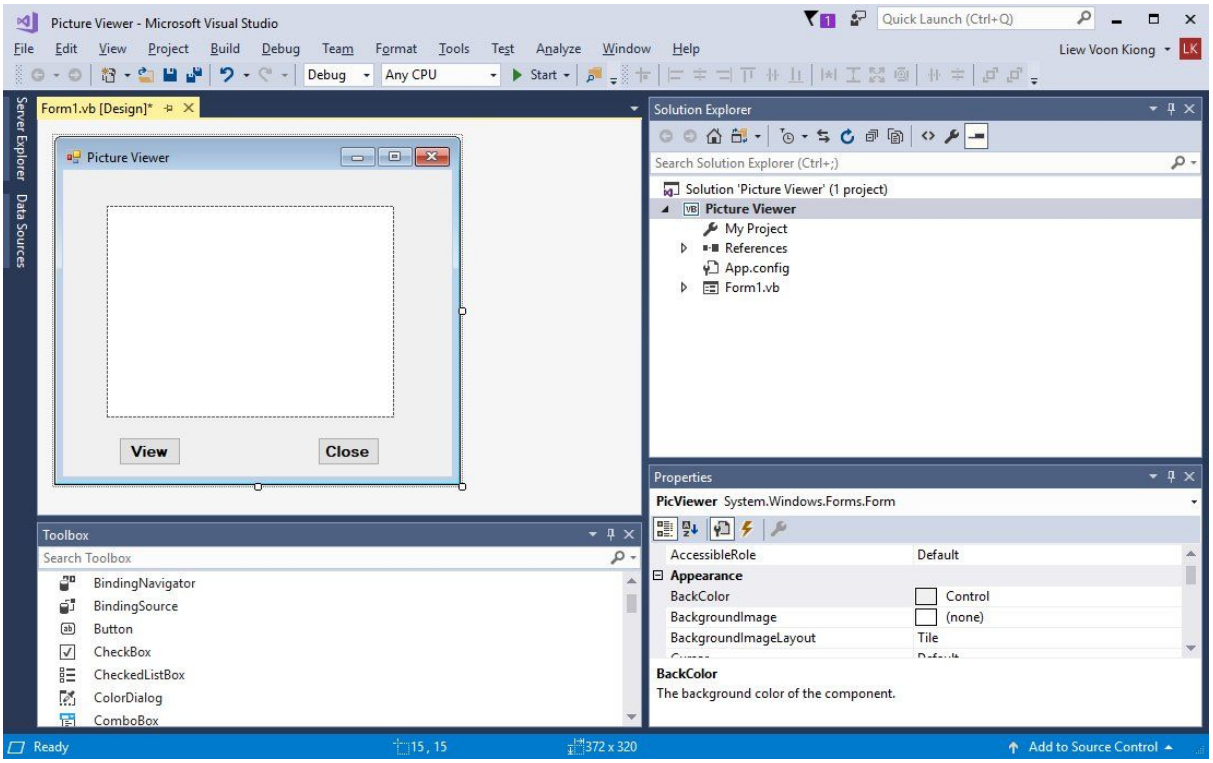

Figure 2.9

Summary

- In section 2.1, you have learned how to customize the form by changing the values of its properties.
- In section 2.2, you have learned how to add controls to the form and change their properties at design phase and at runtime.

Chapter 3
Writing the Code

- ❖ Learn the basics of writing code in Visual Basic 2015

In the previous chapter, we have learned how to design the user interface by adding controls to the form and by changing their properties. However, the user interface alone will not work without adding code to them. In this chapter, we shall learn how to write code for all the controls so that they can interact with the events triggered by the users. Before learning how to write Visual Basic 2015 code, let us dwell into the concept of event-driven programming

3.1 The Concept of Event-Driven Programming

Visual Basic 2015 is an event-driven programming language which means that the code is executed in response to events triggered by the user actions like clicking the mouse, pressing a key on the keyboard, selecting an item from from a drop-down list, typing some words into textbox and more. It may also be an event that runs in response to some other program . Some of the common events in Visual Basic 2015 are load, click, double-click, drag and drop, pressing the keys and more.

Every form and every control you place on the form has a set of events related to them. To view the events, double-click the control (object) on the form to enter the code window. The default event will appear at the top part on the right side of the code window. You need to click on the default event to view other events associated with the control. The code appears on the left side is the event procedure associated with the load event. Figure 3.1 illustrates the event procedure load associated with the Form (its name has been changed to PicViewer therefore you can see the words PicViewer events) and Figure 3.2 shows the events associated with button.

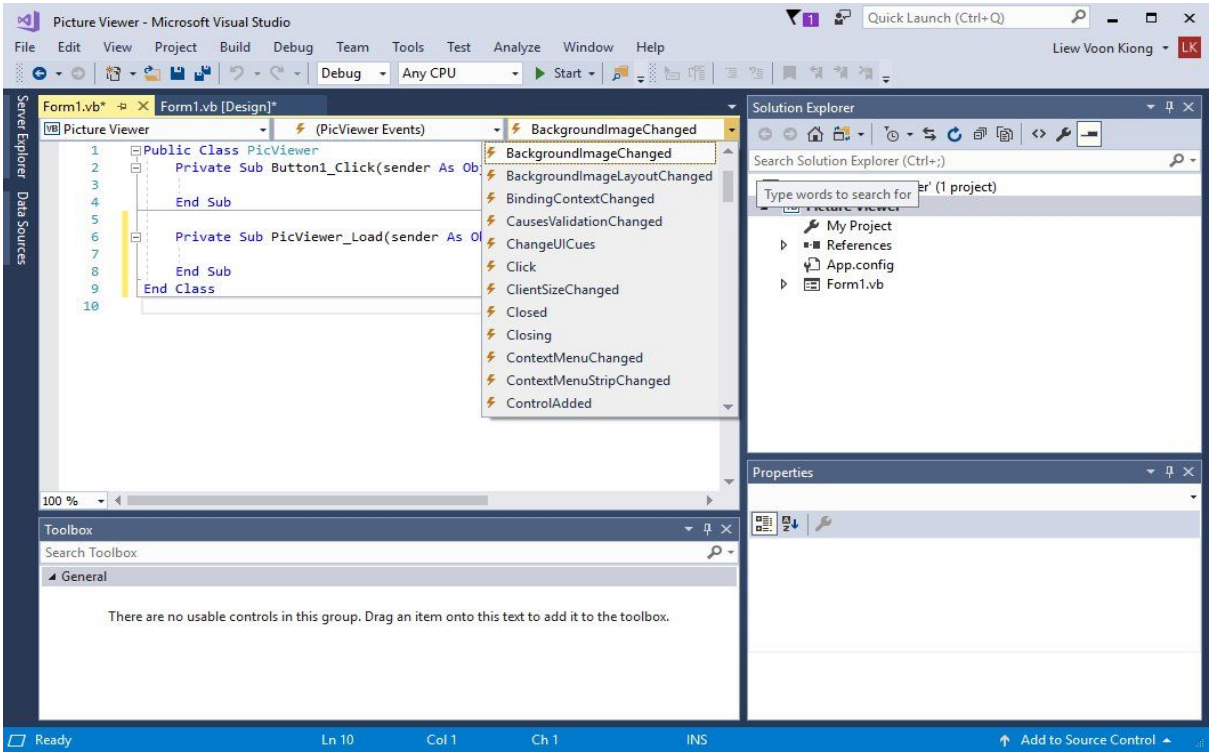

Figure 3.1: Events associated with Form

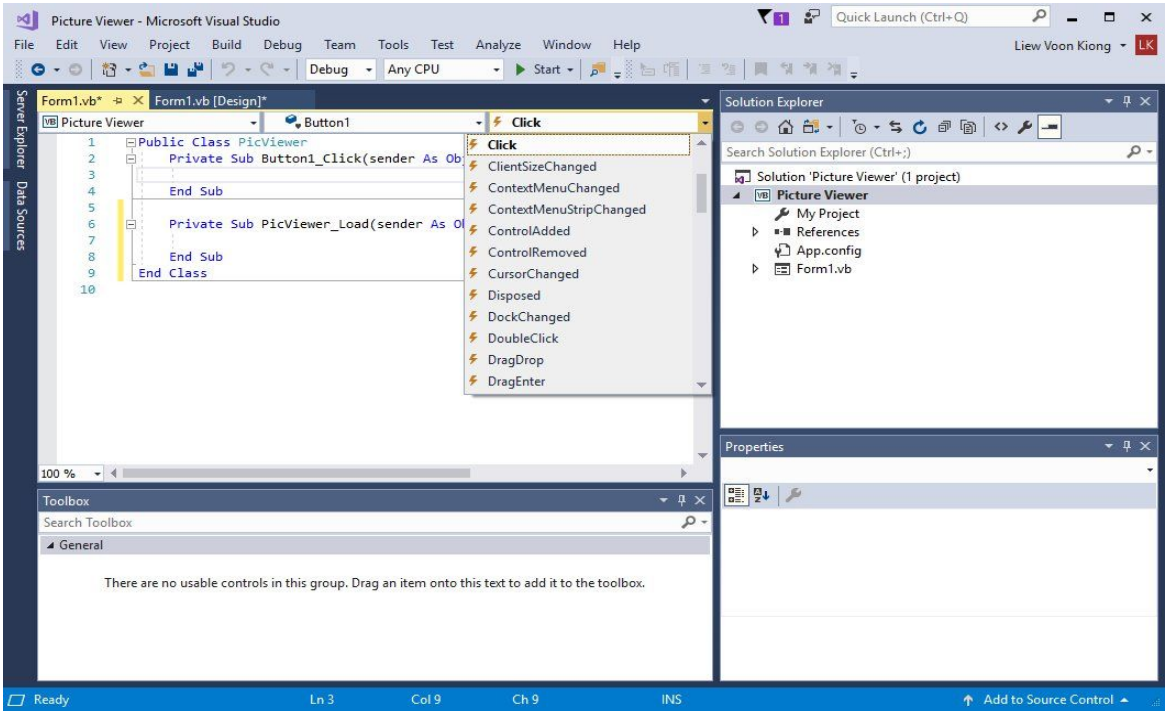

Figure 3.2: Events associated with the button

3.2 Writing the Code

To start writing code in Visual Basic 2015, click on any part of the form to go into the code window as shown in Figure 3.1. The event procedure is to load Form1 and it starts with the keywords **`Private Sub`** and ends with **`End Sub`**. This procedure includes the Form1 class and the event Load, and they are bind together with an underscore, i.e. **`Form_Load`**. It does nothing other than loading an empty form. To make the load event does something, insert the statement.

```
MsgBox ( "Welcome to Visual Basic 2015" )
```

The Code

```
Public Class Form1
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
MsgBox ( "My First Visual Basic 2015 Program", ,"My Message")
End Sub
End Class
```

MsgBox is a built-in function that displays a message in a pop-up message box. The function comprises a few arguments, the first is the message that is displayed and the third is the title of the message box. Running the program produces the message "My First Visual Basic 2015 Program" , as shown in Figure 3.3.

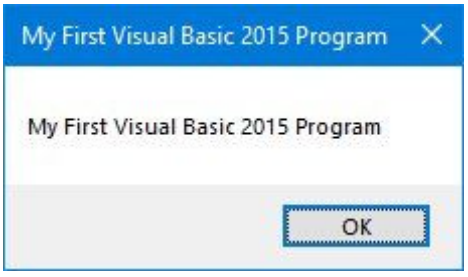

Figure 3.3

You will notice that above Private Sub structure there is a preceding keyword **`Public Class Form1`**. This is the concept of an object oriented programming language. When we start a windows application in Visual Basic 2015, we will see a default form with the name Form1 appears in the IDE, it is actually the Form1 Class that inherits from the Form class **`System.Windows.Forms.Form`**. A class has events as it creates an instant of a class or an object.

You can also write code to perform arithmetic calculation. For example, you can use the **`MsgBox`** and the arithmetic operator plus to perform addition of two numbers, as shown below:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
MsgBox("2" & "+" & "5" & "=" & 2 + 5)
End Sub
```

*The symbol & (ampersand) is to perform string concatenation. The output is as shown in Figure 3.4

Figure 3.4

Summary

- In section 3.1, you have learned the concepts of event driven programming
- In section 3.2, you have learned how to write code for the controls