

Chapter 1

How to get started with Visual Studio

Objectives

Applied

1. Use Visual Studio 2015 to do any of these operations:
 - Open and close an existing C# project or solution
 - Display the Form Designer for each of the forms in a project
 - Display the Code Editor for each of the forms in a project
 - Use the Solution Explorer to review the files in a project
 - Open, hide, and adjust the windows for a project
 - Build and run a project

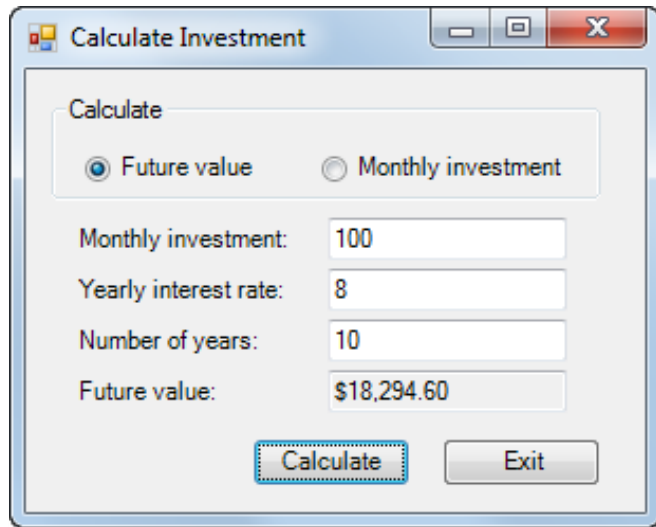
Knowledge

1. Describe the main difference between a Windows Forms application and a Web Forms application.
2. List two additional types of .NET applications that you can develop with Visual Studio 2015.

Objectives (cont.)

3. Name the two languages that you can use for rapid application development with Visual Studio 2015.
4. Describe the two main components of the .NET Framework.
5. In general terms, describe the C# compiler, Microsoft Intermediate Language, the assembly, and the Common Language Runtime.
6. Describe the use of each of these windows in the Visual Studio IDE: Form Designer, Code Editor, and Solution Explorer.
7. In general terms, explain what makes it possible to select a target framework with Visual Studio 2015.

A Windows Forms application running on the desktop



Common types of .NET applications

Type	Description
Windows Forms	Runs in its own window on the user's PC and consists of one or more Windows.
ASP.NET Web Forms	Runs on a web server and consists of one or more web forms that define pages that are displayed in a browser on the client.
ASP.NET MVC	Runs on a web server and consists of pages that are displayed in a browser on the client machine. Unlike ASP.NET Web Forms, ASP.NET MVC uses a Model-View-Controller design pattern to create its pages.
WPF	Runs in windows on the user's PC and provides an enhanced user experience.

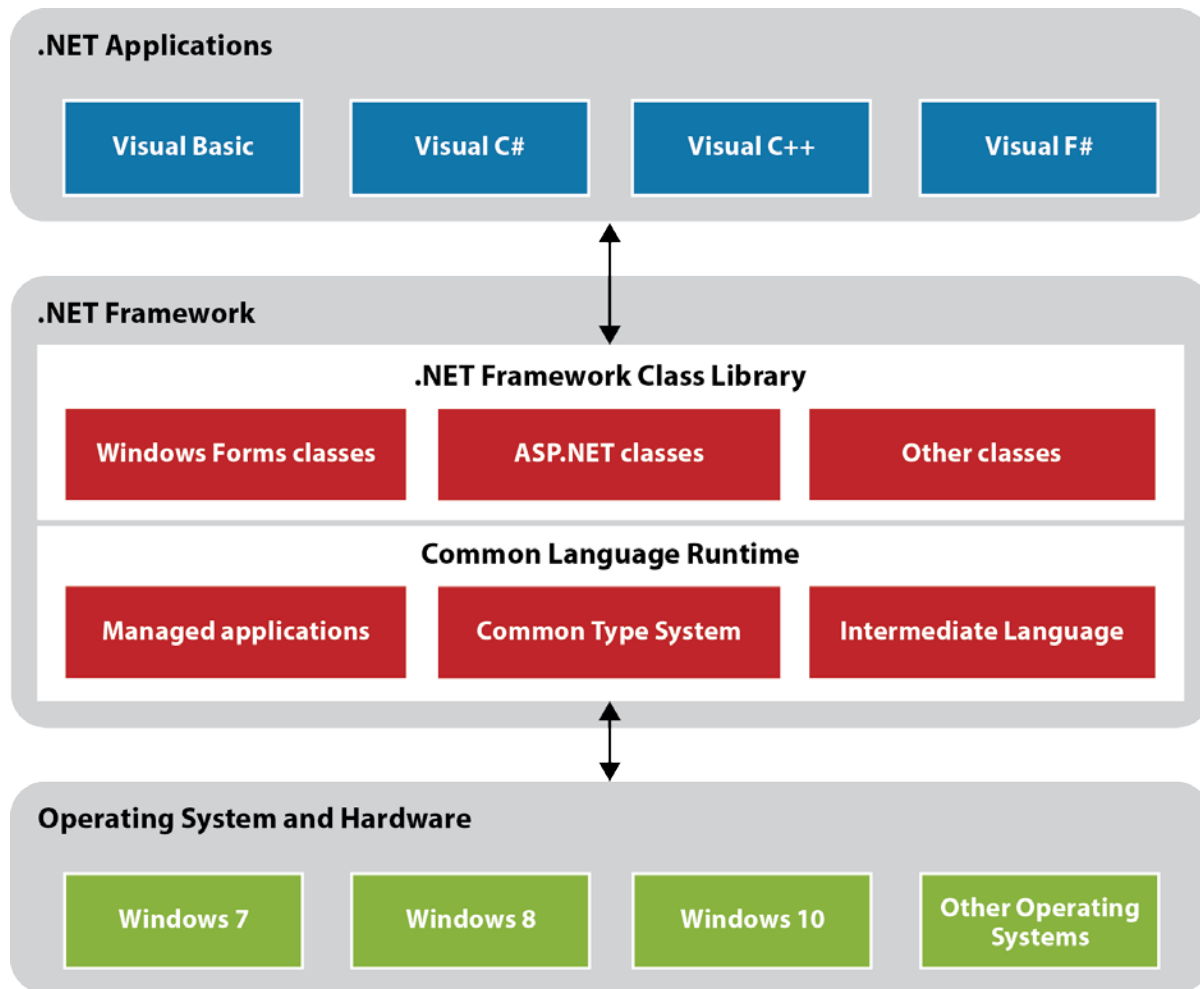
Visual Studio 2015 Editions

- Community Edition
- Professional Edition
- Enterprise Edition

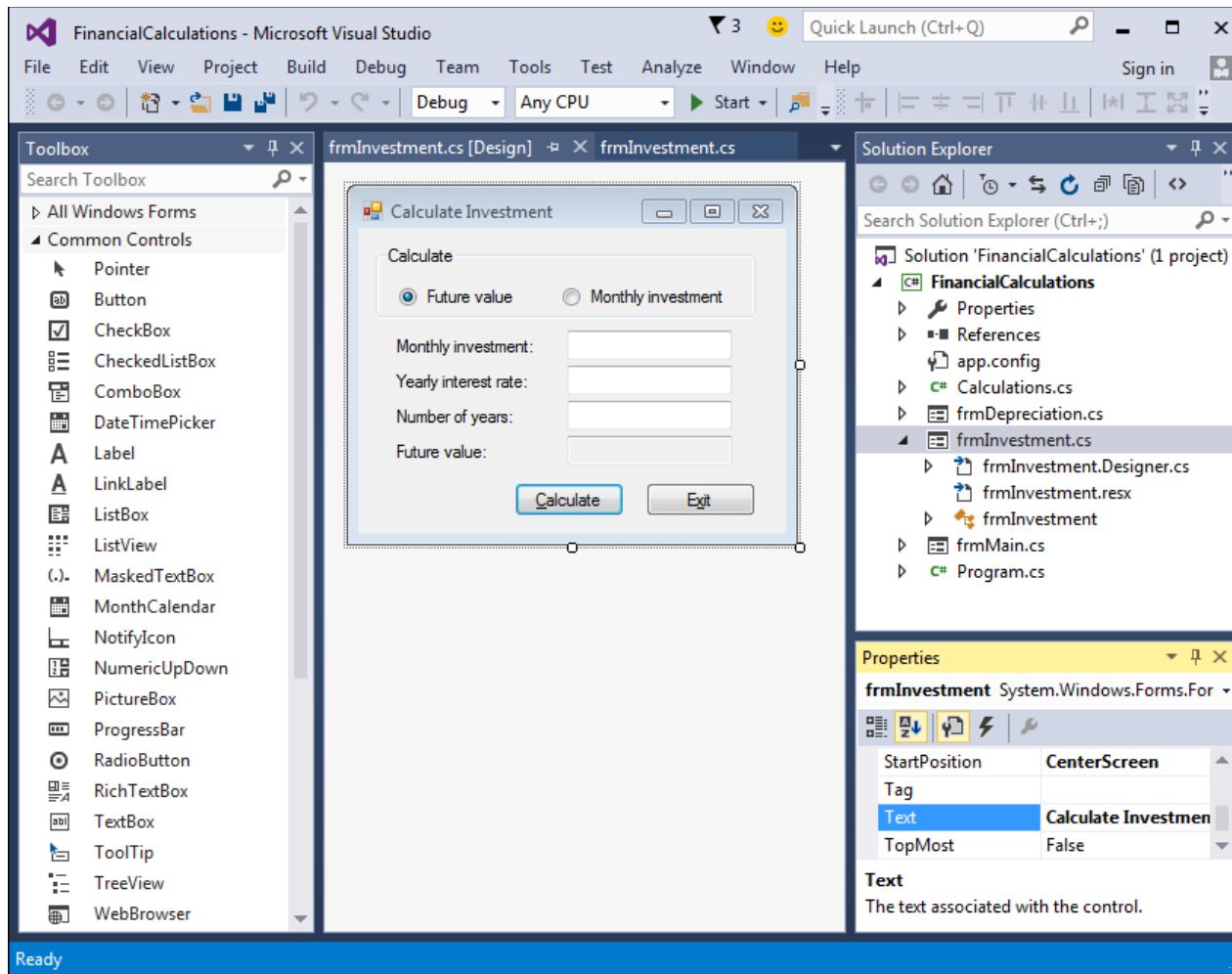
Programming languages supported by Visual Studio 2015

- Visual Basic
- Visual C#
- Visual C++
- Visual F#

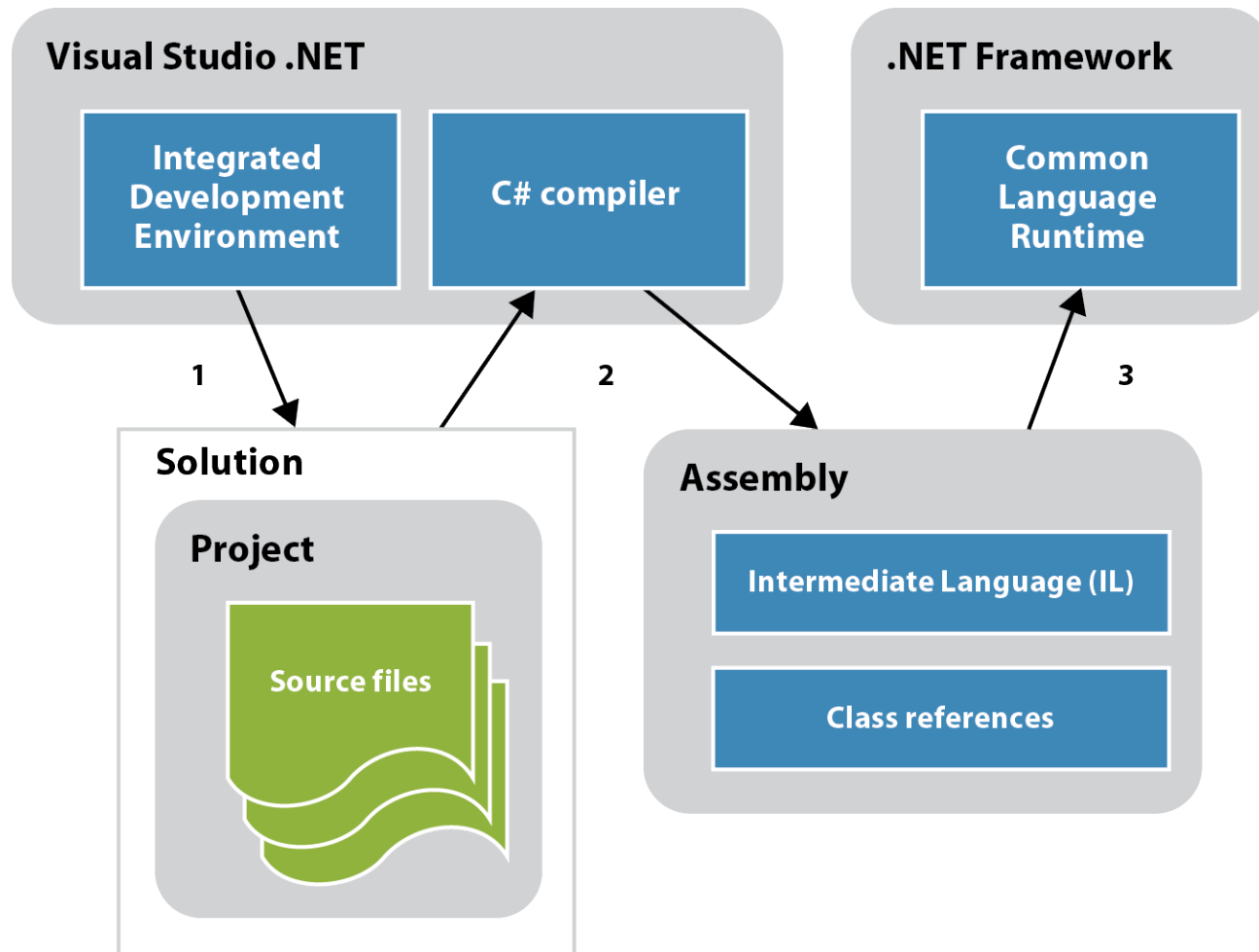
The .NET Framework



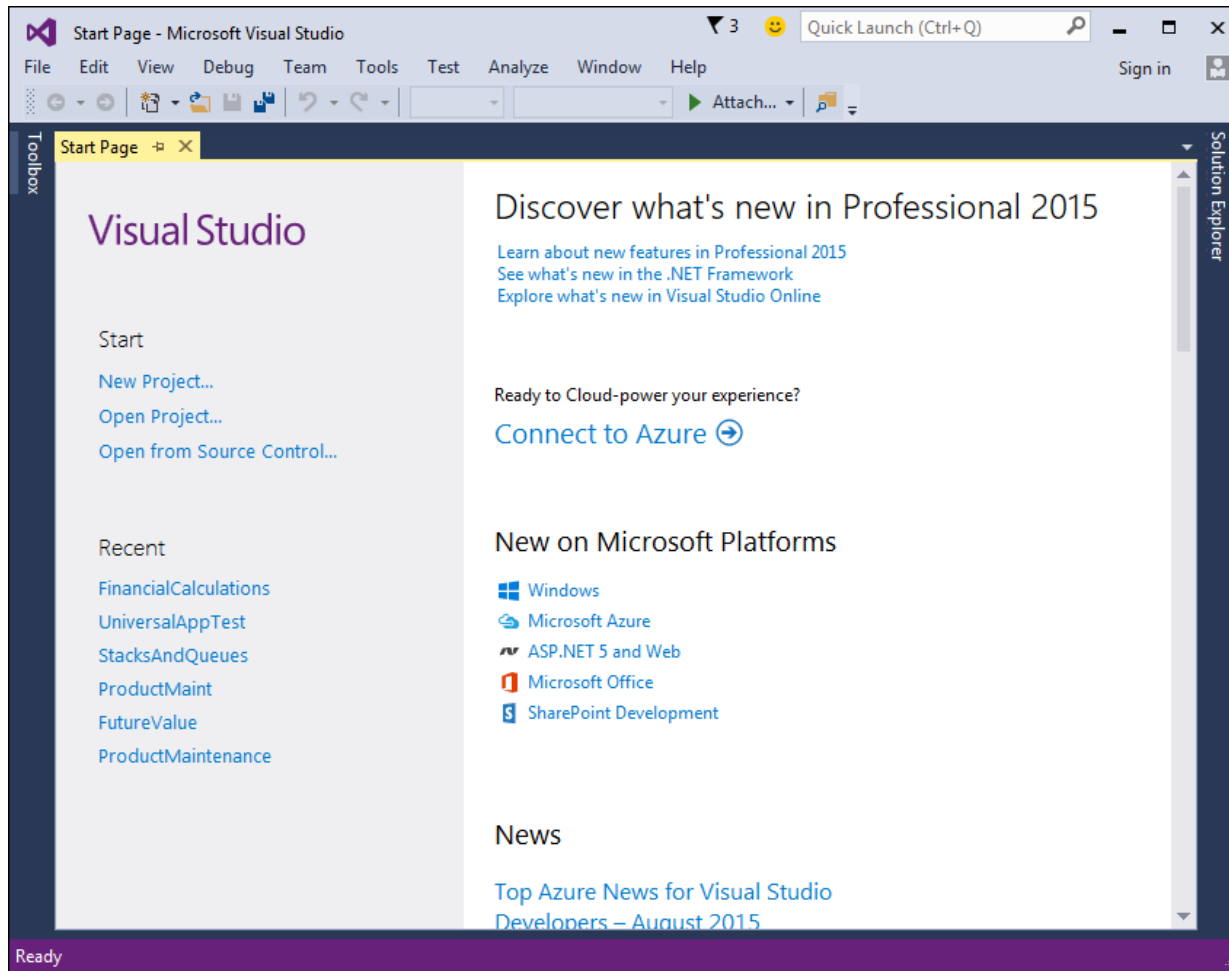
The Visual Studio IDE



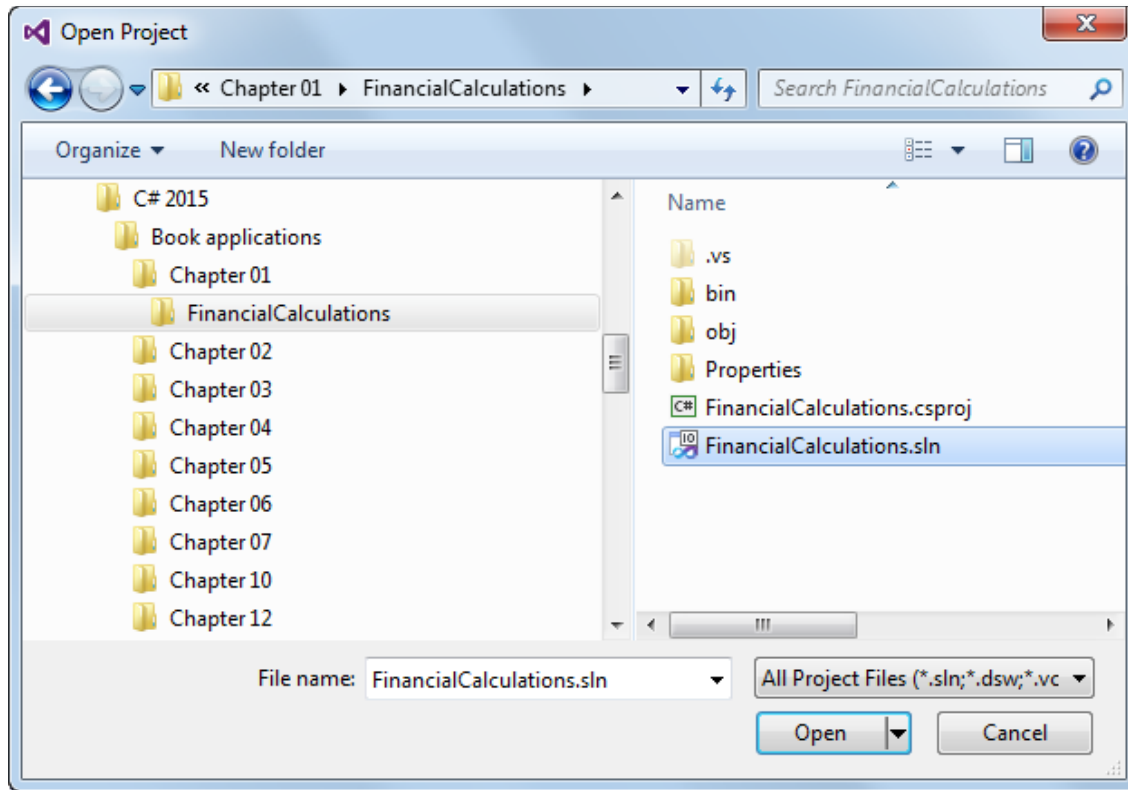
How a C# application is compiled and run



The Visual Studio Start Page



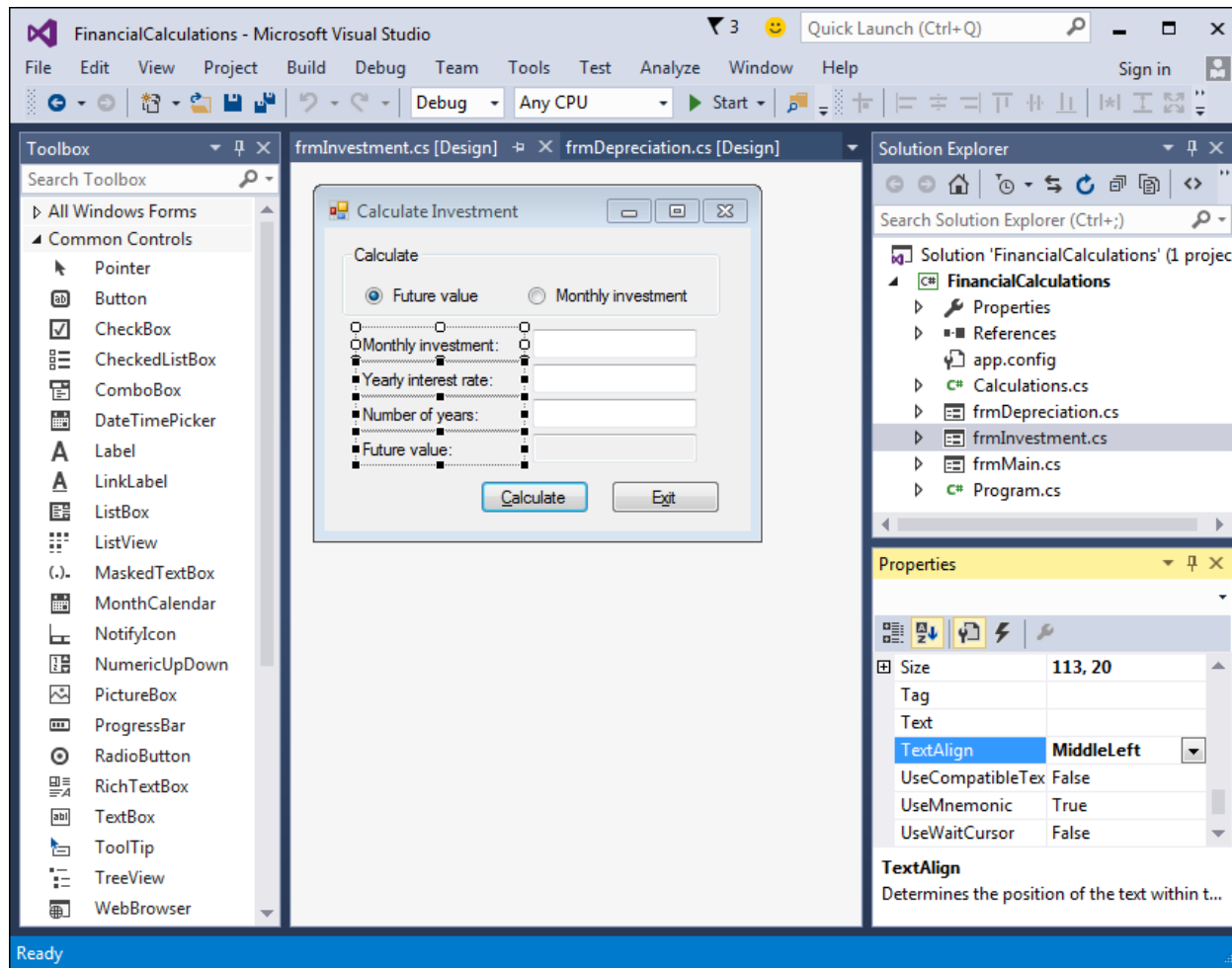
The Open Project dialog box



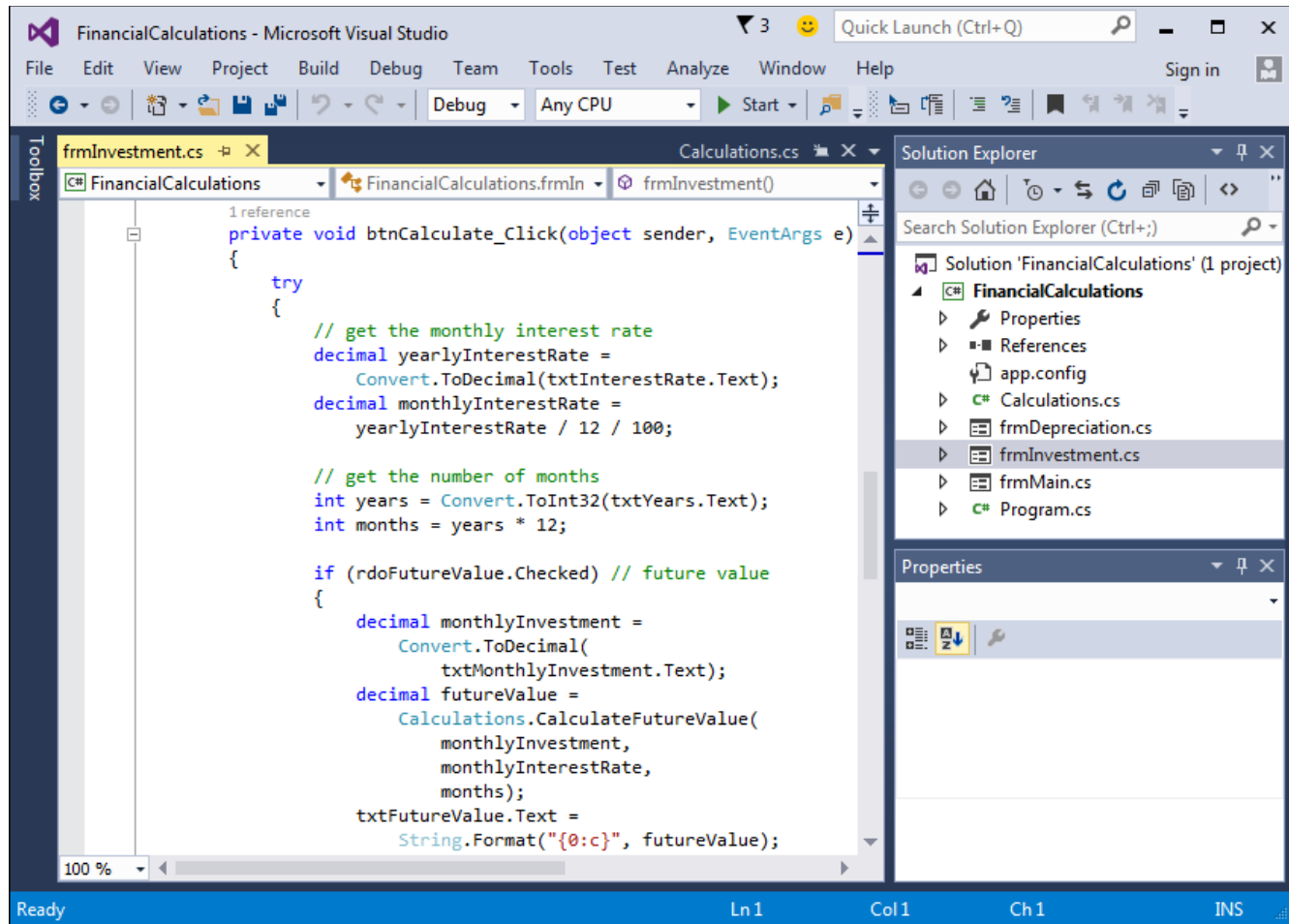
Project and solution concepts

- Every C# project has a *project file* with an extension of *csproj* that keeps track of the files that make up the project and records various settings for the project.
- Every solution has a *solution file* with an extension of *sln* that keeps track of the projects that make up the solution.
- When you open a project file, Visual Studio opens the solution that contains the project. And when you open a solution file, Visual Studio automatically opens all the projects contained in the solution.
- The project file can be stored in the same directory as the solution file or in one of its subdirectories.

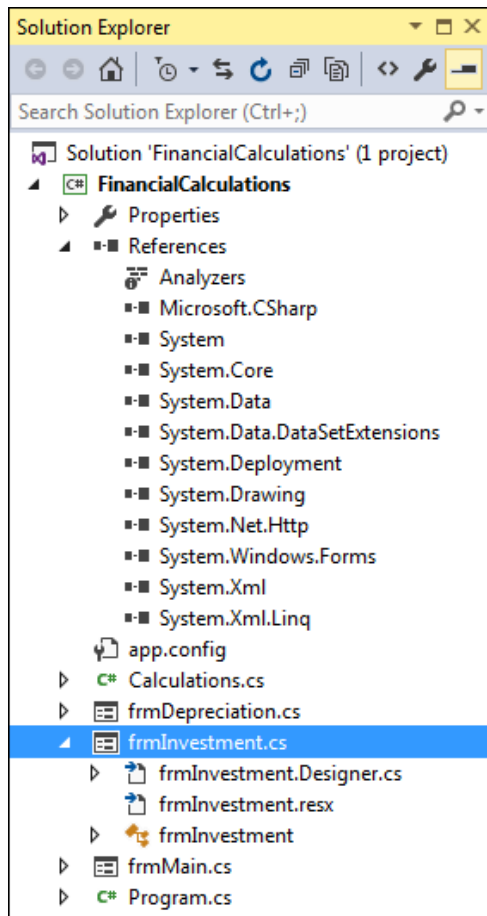
The Form Designer window of Visual



The Code Editor window of Visual



The Solution Explorer



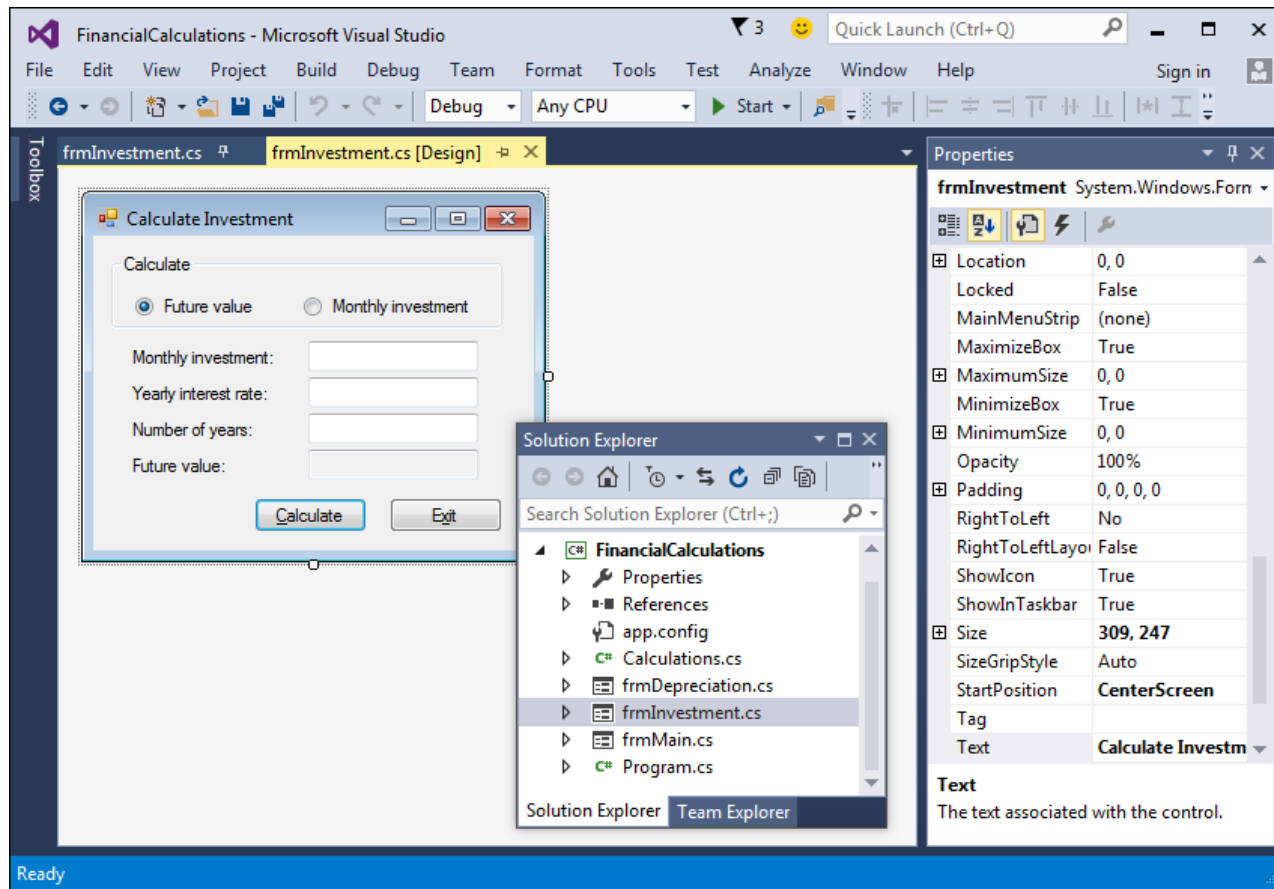
Project files

- The Solution Explorer uses different icons to distinguish between source code files that define forms and source code files that don't.
- Each form is defined by two C# source files where each file contains a *partial class*. The file with the cs extension contains the code that's entered by the programmer, and the file with the Designer.cs extension contains the code that's generated when the programmer works with the Form Designer. When the project is compiled, these partial classes are combined into a single class that determines the appearance and operation of the form.
- A source code file that doesn't define a form is usually stored in a single C# source file that contains a single class.

Project files (cont.)

- The Program.cs file contains the C# code that starts the application and runs the first form. The app.config file contains configuration information for the application.
- The References folder contains references to the assemblies for the namespaces that the application can use.

Two floating windows, a hidden window, and a pinned window



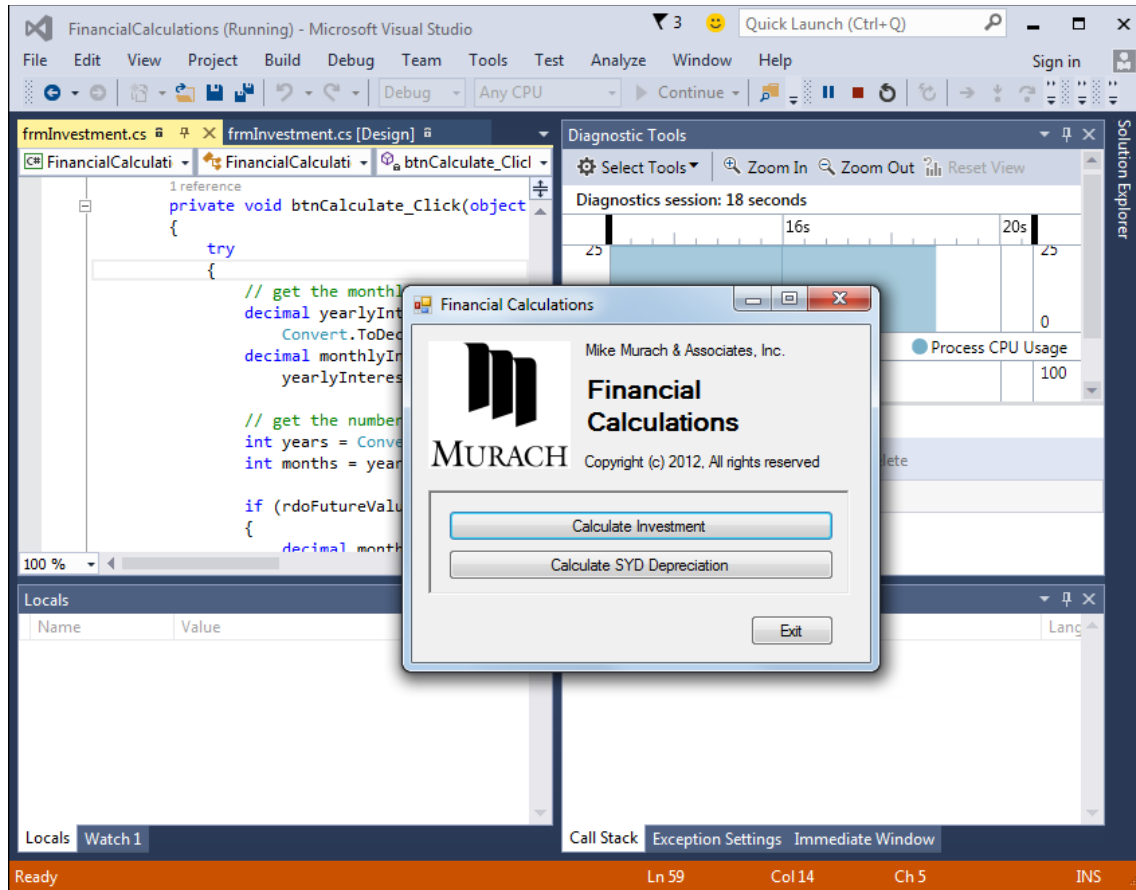
How to rearrange windows

- To undock a *docked window* so it floats on the screen, drag it by its title bar away from the edge of the application window or double-click its title bar.
- To dock a floating window, drag it by its title bar onto one of the positioning arrows that become available. Or, hold down the Ctrl key and then double-click its title bar to return it to its previous docked location.
- To hide a docked window, click its Auto Hide button. Then, the window is displayed as a tab at the edge of the screen, and you can display it by clicking on the tab.
- To display a window in a group of tabbed windows, click on its tab.
- If you dock, undock, hide, or unhide a tabbed window, all the windows in the group are docked, undocked, hidden, or unhidden.

How to rearrange windows (cont.)

- To pin a window to the left side of the Form Designer, click its Toggle Pin Status button.
- You can also use the commands in the Window menu to work with windows. For example, you can use the Reset Window Layout command to return the windows to their default layout. You can also save, apply, and manage custom layouts.

The form that's displayed when the Financial Calculations project is run



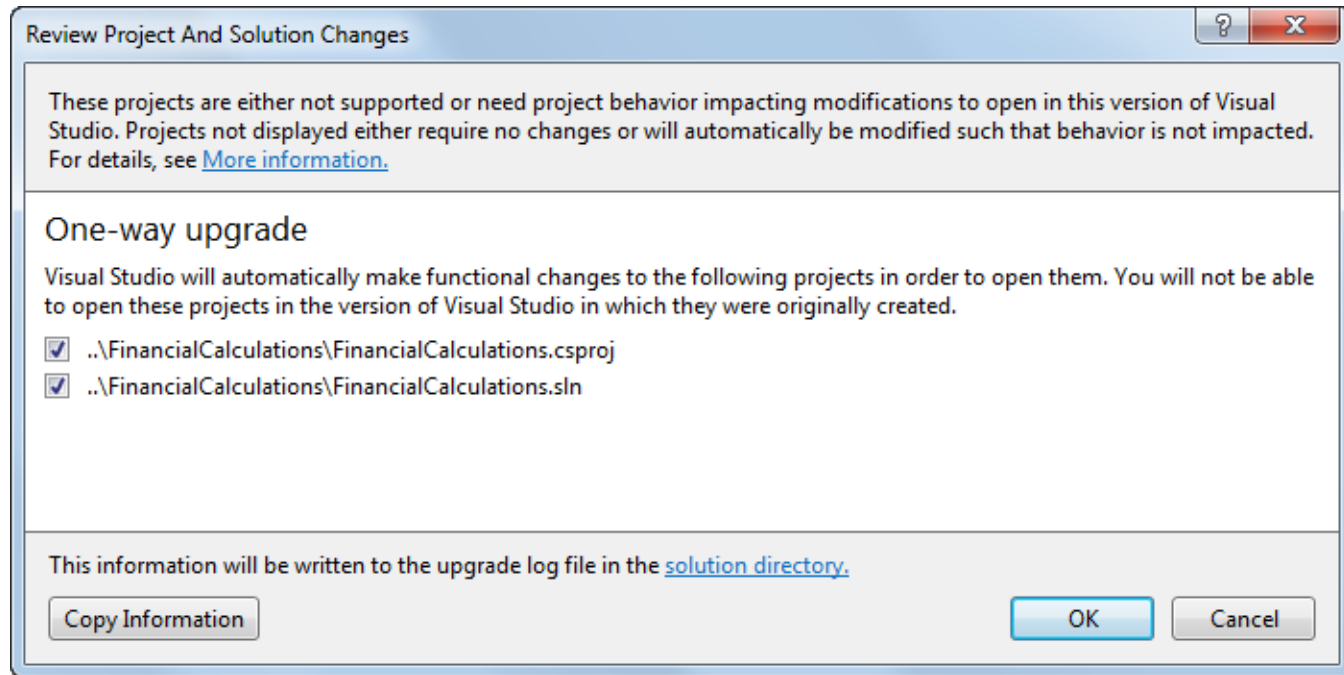
How to build a project without running it

- Use the Build→Build Solution command. Or, right-click the project in the Solution Explorer and select the Build command from the shortcut menu.

How to run a project

- You can *run* a project by clicking on the Start button in the Standard toolbar or by pressing F5. Then, the first form of the project is displayed on top of the Visual Studio window.
- If the project hasn't already been built, the project is first built and then run.

The dialog box for upgrading a project to Visual Studio 2015



The application properties for a project

