

## 1

# Introduction to Computers, the Internet and Visual C#



## 1.10 C#

- C# was designed specifically for the .NET platform as a language that would enable programmers to migrate easily to .NET.
- C# is object oriented and has access to a powerful **class library** of prebuilt components.
- It has roots in C, C++ and Java, adapting the best features of each.



## 1.10 Visual C#

- Visual C# is an event-driven, visual programming language.
- You'll write programs that respond to **events** such as mouse clicks and keystrokes.
- You'll also use Visual Studio's graphical user interface to drag and drop predefined objects like buttons and textboxes into place.



## 1.10 Visual C# (Cont.)

- Microsoft introduced C# along with its .NET strategy in 2000.
- The **.NET platform** allows applications to be distributed to a variety of devices.
- The original C# programming language was standardized by Ecma International in December 2002.
- Since that time, Microsoft has proposed several language extensions that have been adopted as part of the revised Ecma C# standard.



## 1.13 Key Software Trend: Object Technology

- **Object technology** is a packaging scheme for creating meaningful software units.
- Almost any noun can be reasonably represented as a software object.
- Objects
  - have **properties** (also called **attributes**)
  - perform **actions** (also called **behaviors** or **methods**)



## 1.13 Key Software Trend: Object Technology (Cont.)

- **Classes** are types of related objects.
  - A class specifies the general format of its objects, and the properties and actions available to an object.
  - An object is related to its class in much the same way as a building is related to its blueprint.
- **Procedural programming languages** focus on actions rather than things.



## 1.13 Key Software Trend: Object Technology (Cont.)

- Properly designed classes can be reused on future projects.
- Using libraries of classes reduces the amount of effort required to implement new systems.
- Instead of worrying about minute details, you can focus on the behaviors and interactions of objects.



## 1.16 Introduction to Microsoft .NET

- Microsoft's **.NET initiative** uses the Internet and the web in the development, engineering, distribution and use of software.
- Applications in any .NET-compatible language can interact with each other.
- Microsoft's **ASP.NET** technology allows you to create web applications.
- The .NET strategy allows programmers to concentrate on their specialties without having to implement every component of every application.





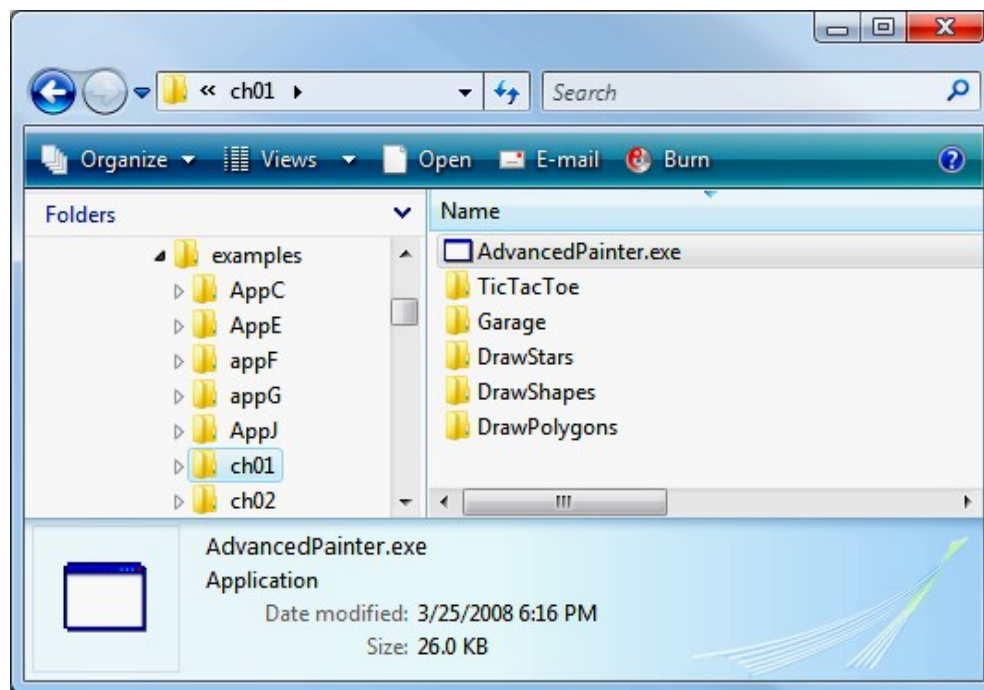
# 1.17 The .NET Framework and the Common Language Runtime

- The Microsoft **.NET Framework**:
  - manages and executes applications and web services
  - contains a class library (called the .NET Framework Class Library)
  - provides security and other programming capabilities.
- The **Common Language Runtime (CLR)**:
  - Programs are compiled first into **Microsoft Intermediate Language (MSIL)**.
  - When the application executes, the **just-in-time compiler** translates the MSIL in the executable file into machine-language code.



## 1.18 Test-Driving a C# Advanced Painter Application

- Open Windows Explorer and navigate to the directory with the chapter's examples (Fig. 1.2).
- Double click `AdvancedPainter.exe` to run the application

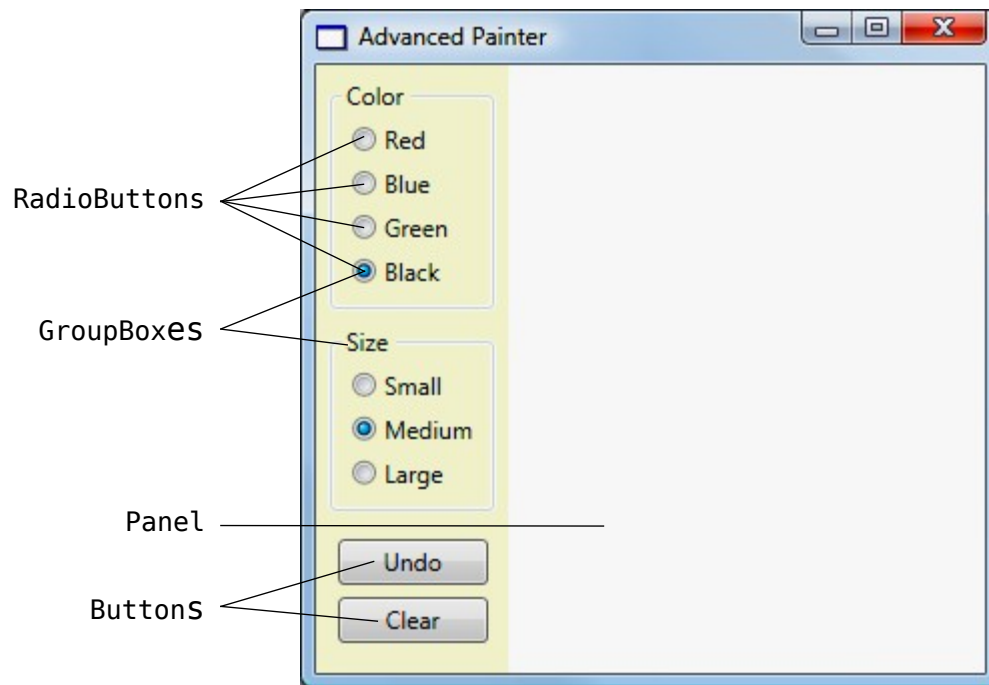


**Fig. 1.2** | Contents of C:\Examples\ch01.



## 1.18 Test-Driving a C# Advanced Painter Application (Cont.)

- In Fig. 1.3, the application's graphical elements—called **controls**—are called out.



**Fig. 1.3** | Advanced Painter application.

## 1.18 Test-Driving a C# Advanced Painter Application (Cont.)

- By using preexisting controls, you can create powerful applications in Visual C#.
- Click the **RadioButton**s labeled **Red** and **Small** to change the brush's color and size (Fig. 1.4).

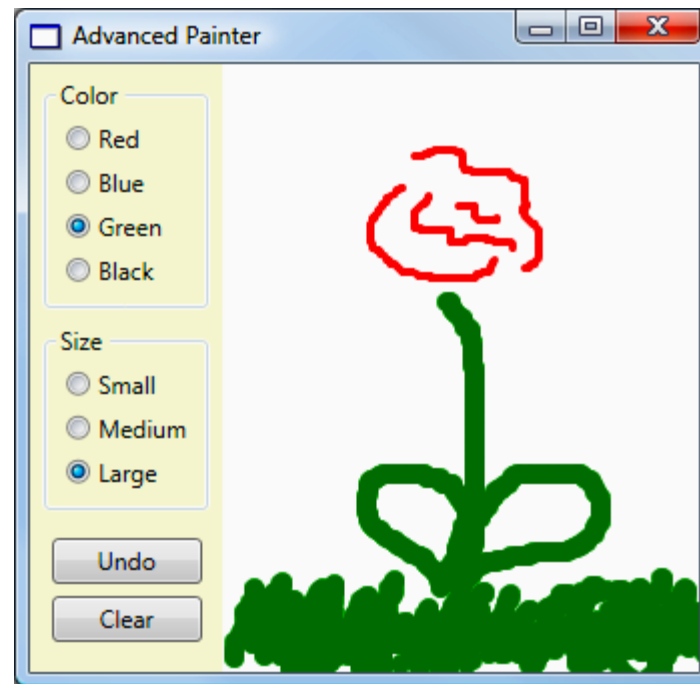


**Fig. 1.4** | Drawing with a new brush color.



## 1.18 Test-Driving a C# Advanced Painter Application (Cont.)

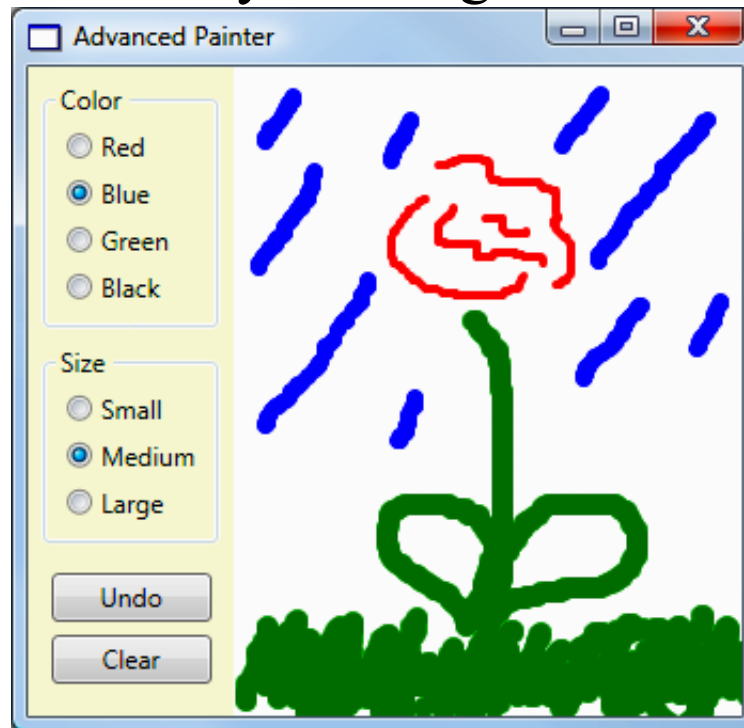
- Click the **RadioButtons** labeled **Green** and **Large** to draw grass and a flower stem (Fig. 1.5).



**Fig. 1.5** | Drawing with a new brush size.

## 1.18 Test-Driving a C# Advanced Painter Application (Cont.)

- Click the **Blue** and **Medium** RadioButtons to draw raindrops (Fig. 1.6).
- Close the application by clicking its **close box**.



**Fig. 1.6** | Drawing with a new brush size.



## 1.18 Test-Driving a C# Advanced Painter Application (Cont.)

- Figure 1.7 lists a few applications in the examples and exercises in this text.
- You are encouraged to test-drive each of them.

Application name	File to execute
Parking Fees	Garage.exe
Tic Tac Toe	TicTacToe.exe
Drawing Stars	DrawStars.exe
Drawing Shapes	DrawShapes.exe
Drawing Polygons	DrawPolygons.exe

**Fig. 1.7** | Examples of C# applications found in this book



## 2

# Dive Into<sup>®</sup> Visual C# 2008 Express





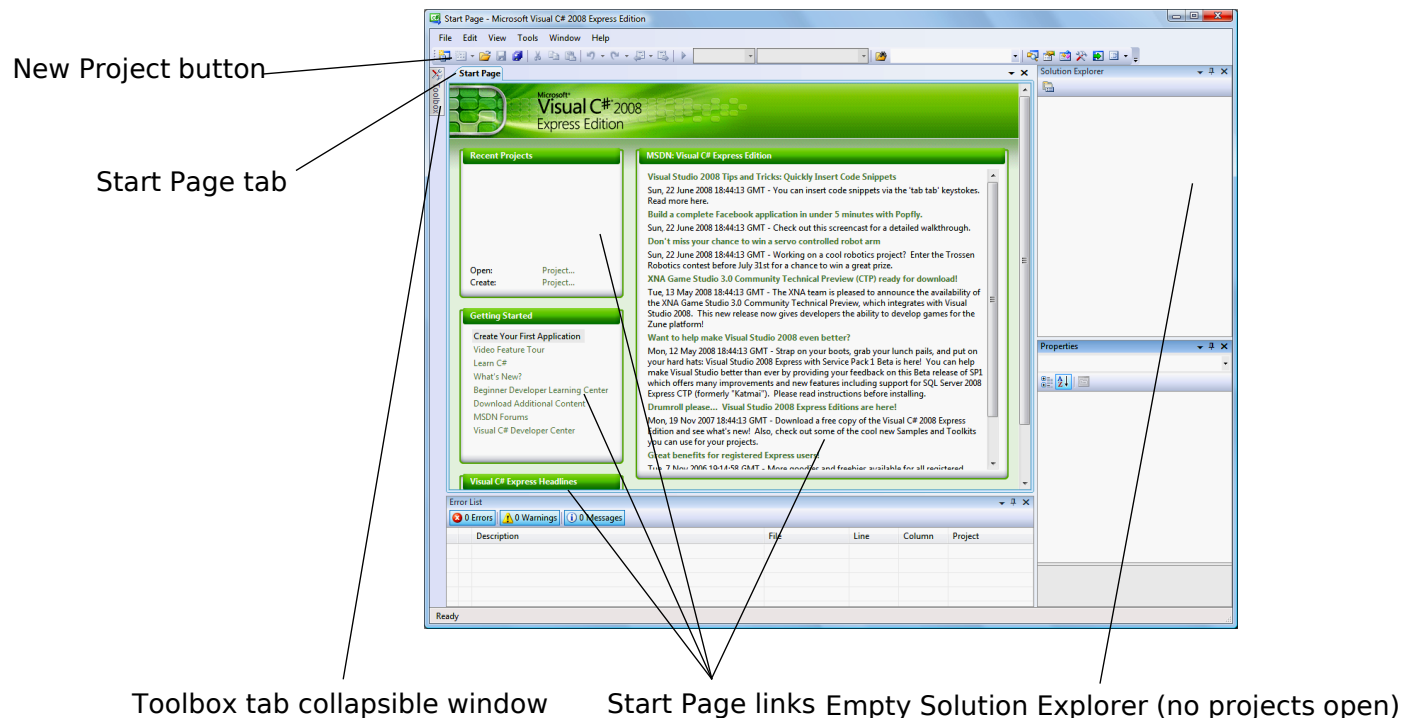
## Note:

- Our IDE is the complete version of Visual Studio .NET 2008
- The disk in the back of the book is the Express Edition of Visual Studio .NET 2008
  - Full versions of Visual Studio 2008 include support for other languages.
- Hopefully there will be vast similarities, few differences of note
  - We shall see



## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

- Select **Start > All Programs > Microsoft Visual C# 2008 Express Edition**.
- The **Start Page** displays (Fig. 2.1).



**Fig. 2.1 | Start Page** in Visual C# 2008 Express Edition.

## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

- The **Start Page** contains a list of IDE resources and web-based resources.
  - **Recent Projects** contains recently modified projects.
  - **Getting Started** focuses on using the IDE for creating programs, learning Visual C# and connecting to the developer community.
  - **Visual C# Express Headlines** and **MSDN: Visual C# Express Edition** provide information about programming in Visual C#.

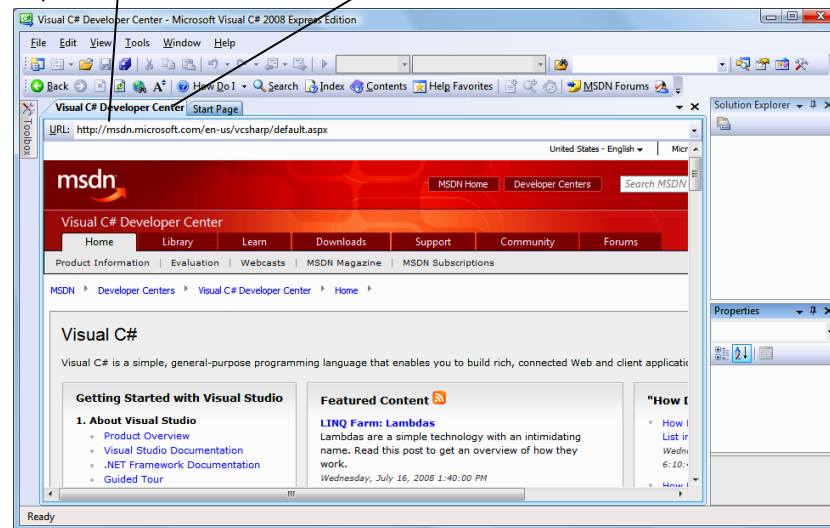


## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

- You can browse the web using the IDE's **internal web browser**.
- Select **View > Other Windows > Web Browser**.
- Request a web page by entering its URL into the location bar (Fig. 2.2).

Requested web page  
(URL in  
location bar drop-  
down)

Selected tab for requested  
web page



**Fig. 2.2** | Displaying a web page in Visual Studio.

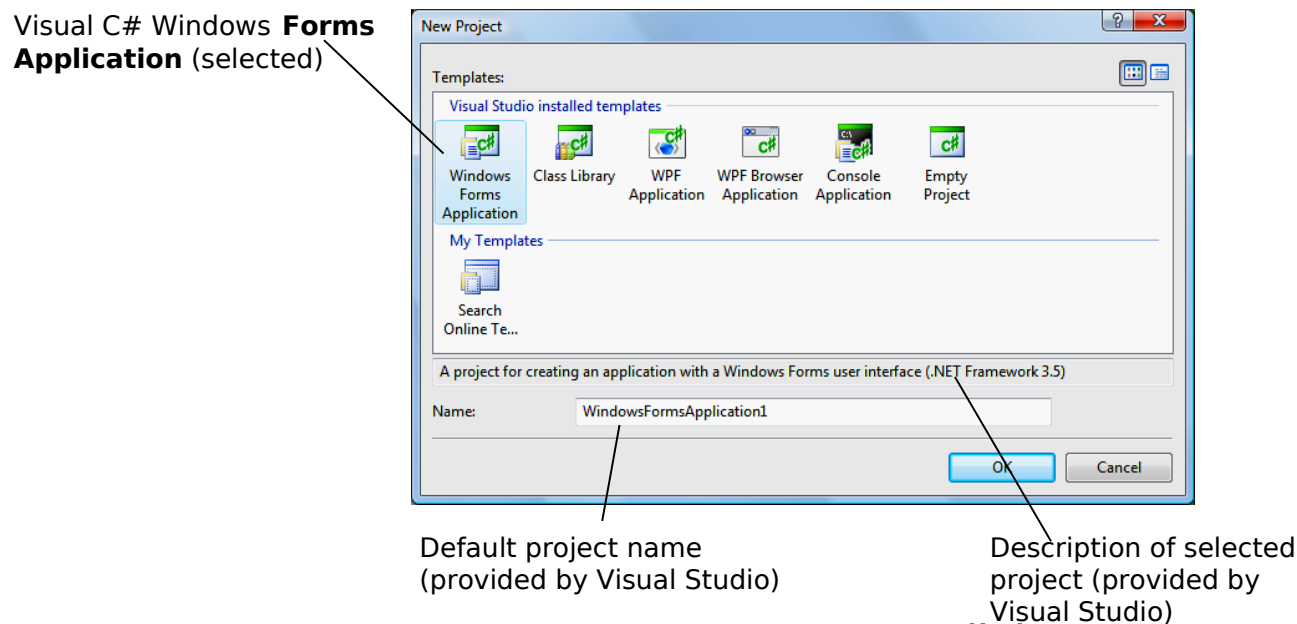
## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

- A **project** is a group of related files, such as the code files and any images that make up a program.
- **Solutions** contain one or more projects.
- To begin programming in Visual C#, select **File > New Project**



## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

- The **New Project dialog** (Fig. 2.3) displays.
- **Templates** are project types users can create in Visual C#.
  - A **Windows Forms application** executes within a Windows operating system and has a **graphical user interface (GUI)**.



**Fig. 2.3 | New Project dialog.**

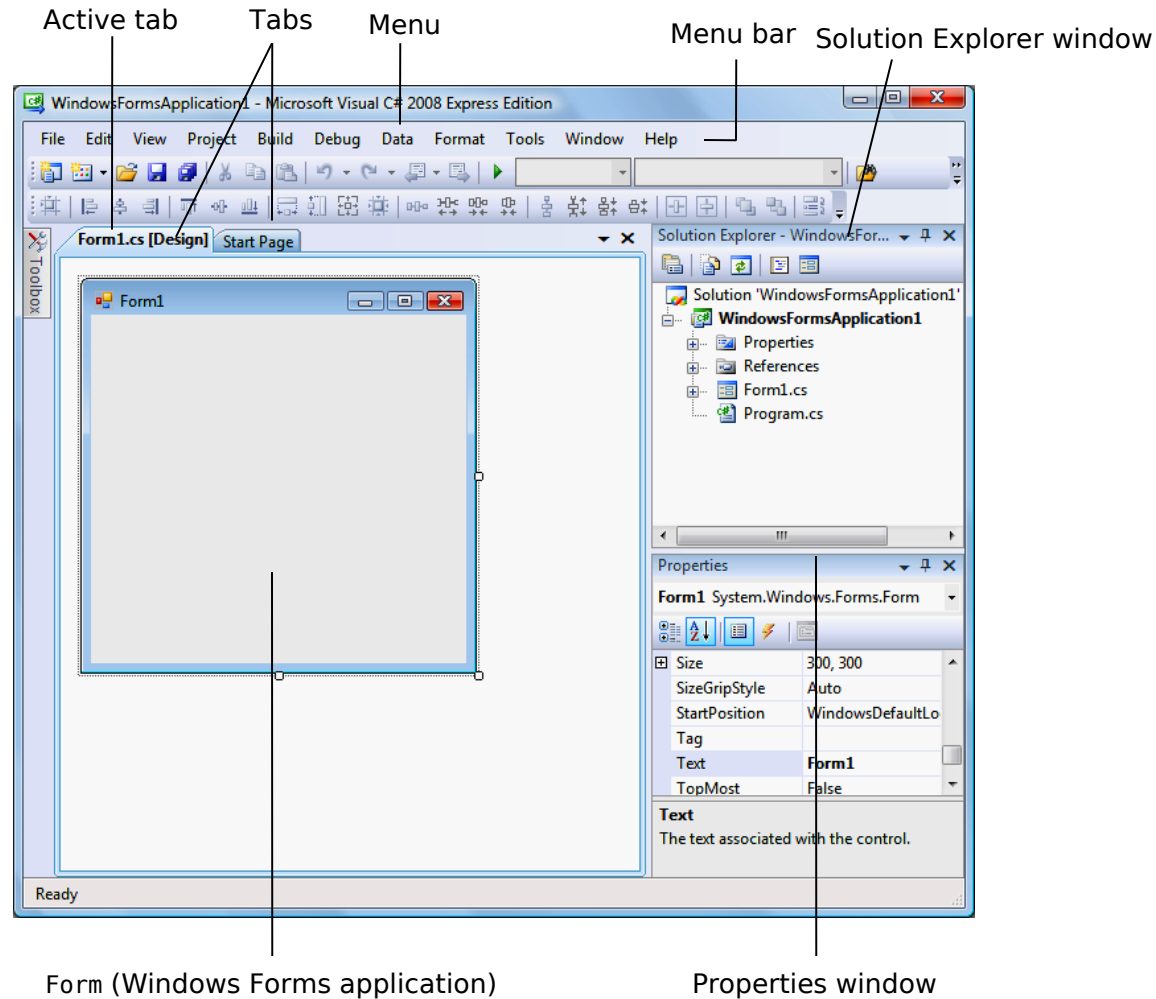


## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

- Click **OK** to display the IDE in **Design view** (Fig. 2.4), which contains features to create programs.
- The gray rectangle (called a **Form**) represents the main window of the application.



## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)



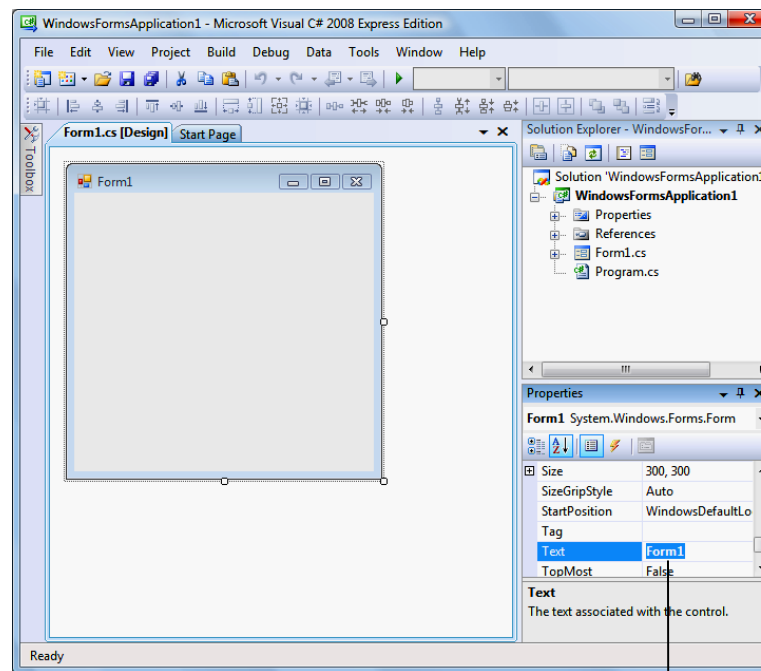
**Fig. 2.4 | Design view of the IDE.**





## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

- Figure 2.5 shows where the Form's name can be modified in the **Properties** window.



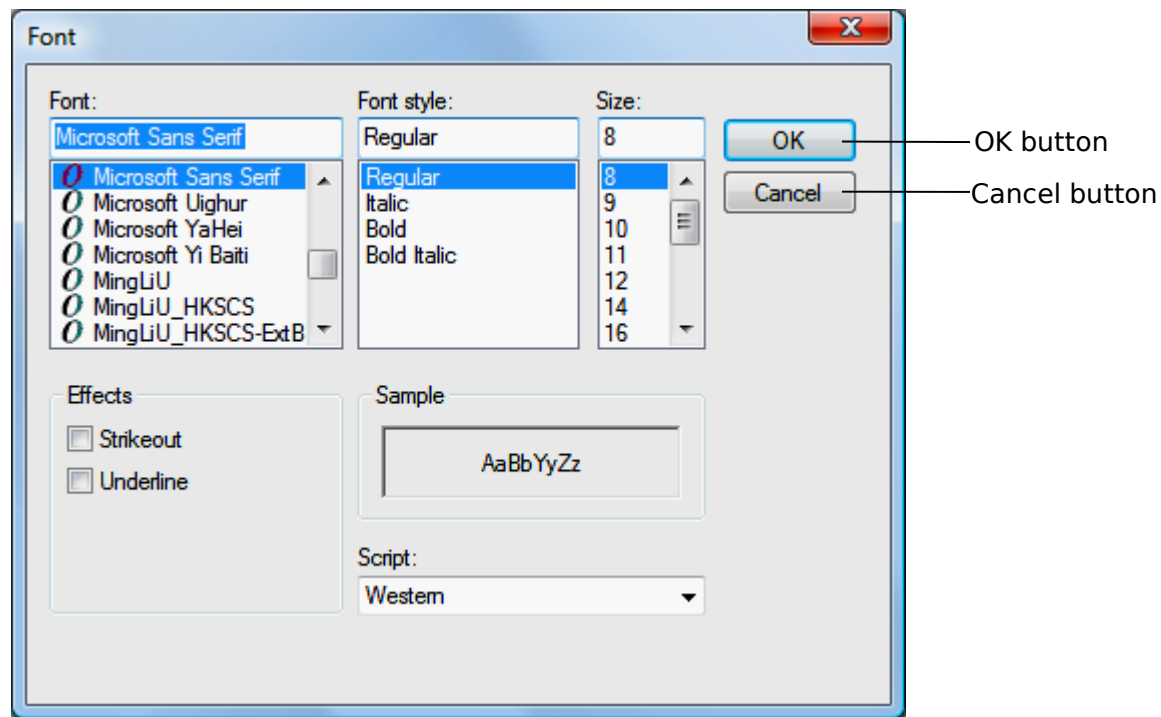
Text box (displaying the Form's name, Form1) which can be modified

**Fig. 2.5** | Textbox control for modifying a property in the Visual Studio IDE.



## 2.2 Overview of the Visual Studio 2008 IDE (Cont.)

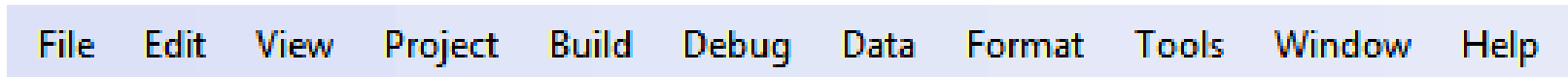
- Figure 2.6 shows a dialog in which a control's font properties can be modified.



**Fig. 2.6** | Dialog for modifying a control's font properties in the Visual Studio IDE.

## 2.3 Menu Bar and Toolbar (Cont.)

- Commands for managing the IDE are contained in **menus** on the **menu bar** of the IDE (Fig. 2.7).
- The set of menus displayed depends on what you are currently doing in the IDE.
- Menus contain groups of related **menu items** that cause the IDE to perform specific actions.



**Fig. 2.7** | Visual Studio menu bar.

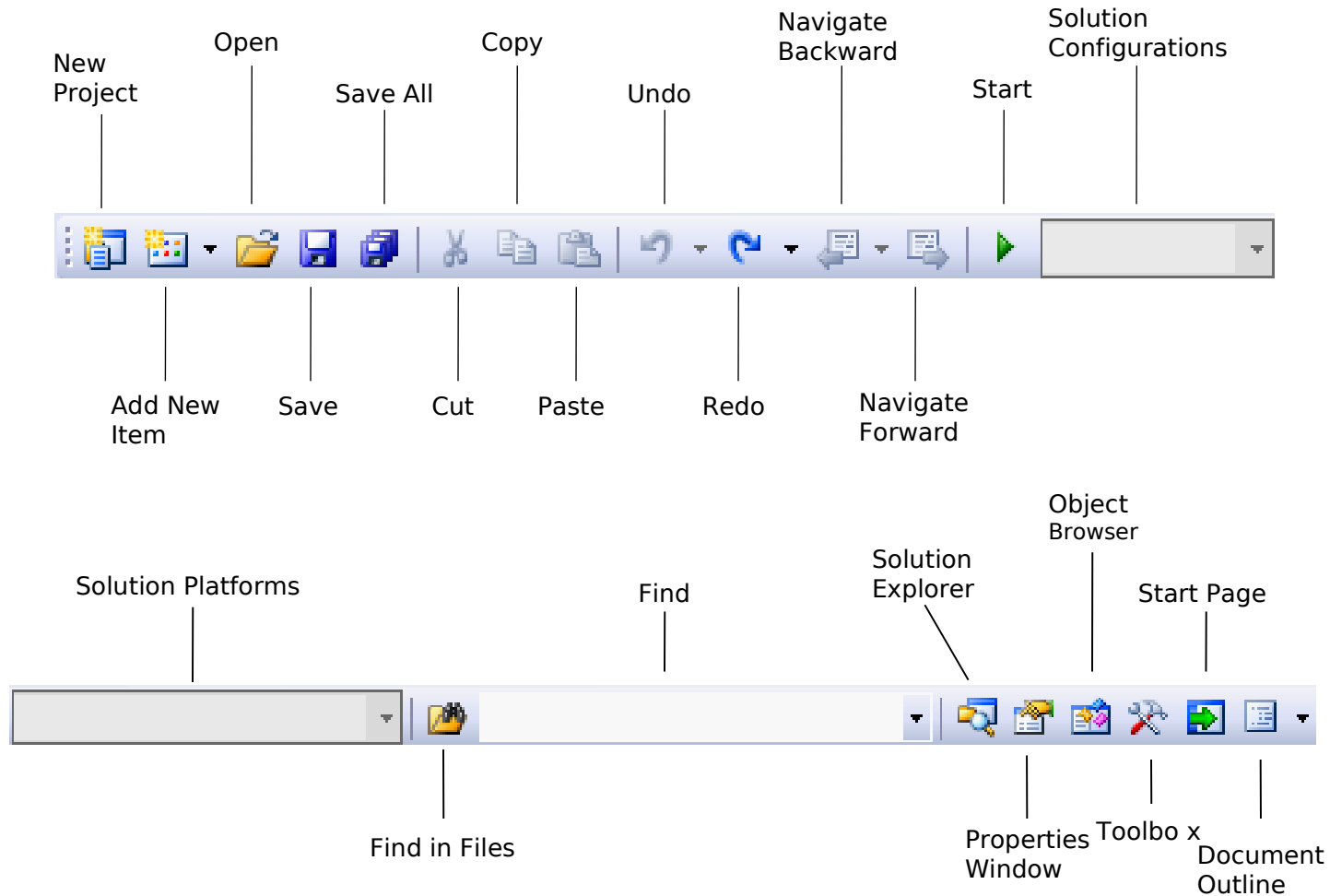
## 2.3 Menu Bar and Toolbar (Cont.)

Menu	Description
<b>File</b>	Commands for opening, closing, adding and saving projects.
<b>Edit</b>	Commands for editing programs, such as cut, copy, paste, undo, redo, delete, find and select.
<b>View</b>	Commands for displaying IDE windows and adding toolbars.
<b>Project</b>	Commands for managing projects and files.
<b>Build</b>	Commands for compiling programs.
<b>Debug</b>	Commands for debugging and running programs.
<b>Data</b>	Contains commands for interacting with databases.
<b>Format</b>	Commands for arranging and modifying a Form's controls.
<b>Tools</b>	Commands for customization of the IDE.
<b>Window</b>	Commands for hiding, opening, closing and displaying IDE windows.
<b>Help</b>	Commands for accessing the IDE's help features.

**Fig. 2.8** | Summary of Visual Studio 2008 IDE menus.



## 2.3 Menu Bar and Toolbar (Cont.)

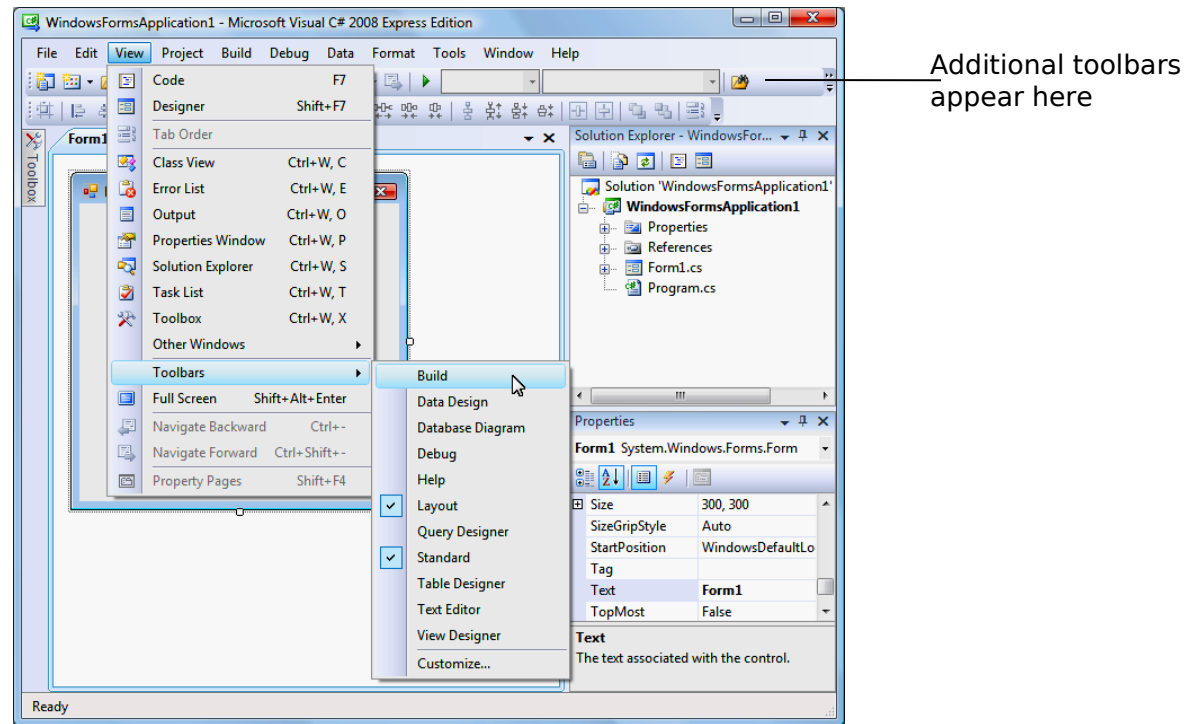


**Fig. 2.9** | Standard Visual Studio toolbar.



## 2.3 Menu Bar and Toolbar (Cont.)

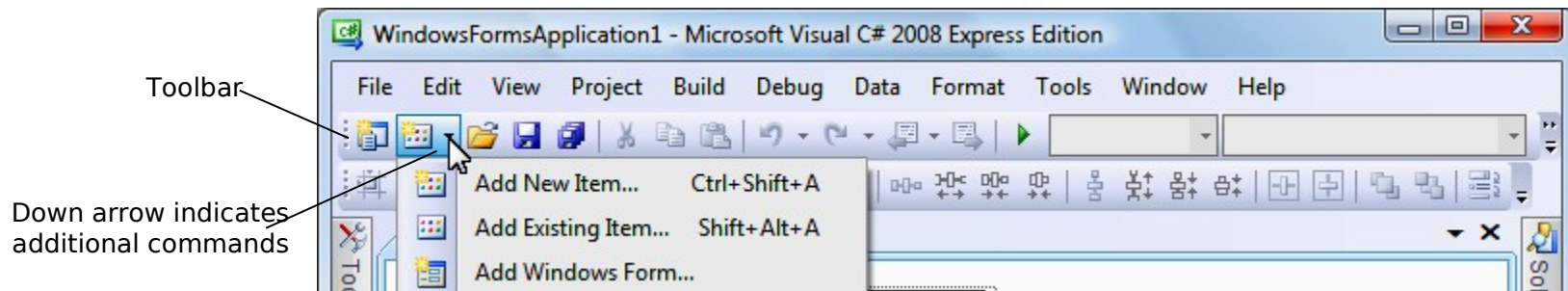
- You can add more toolbars by selecting **View > Toolbars** (Fig. 2.10).



**Fig. 2.10** | Adding the **Build** toolbar to the IDE.

## 2.3 Menu Bar and Toolbar (Cont.)

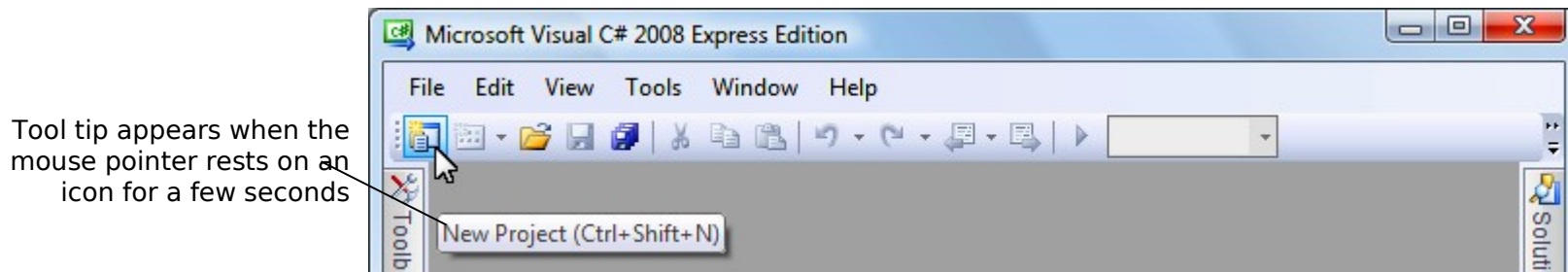
- Some icons contain a down arrow that displays related commands (Fig. 2.11).



**Fig. 2.11** | IDE toolbar icon showing additional commands.

## 2.3 Menu Bar and Toolbar (Cont.)

- Positioning the mouse pointer over an icon highlights it and, after a brief pause, displays a description of the icon called a tool tip (Fig. 2.12).

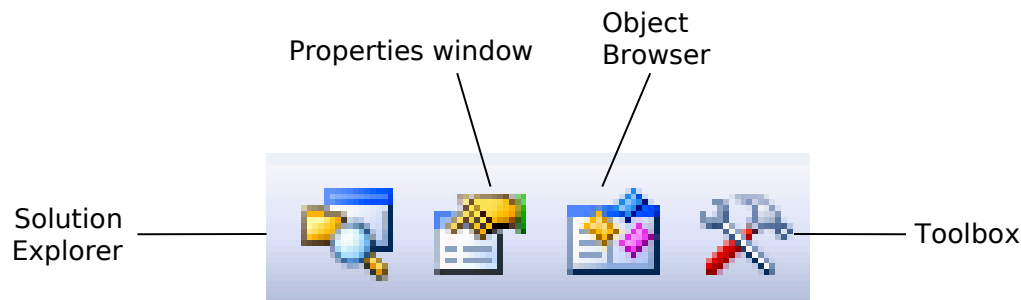


**Fig. 2.12** | Tool tip demonstration.



## 2.4 Navigating the Visual Studio IDE

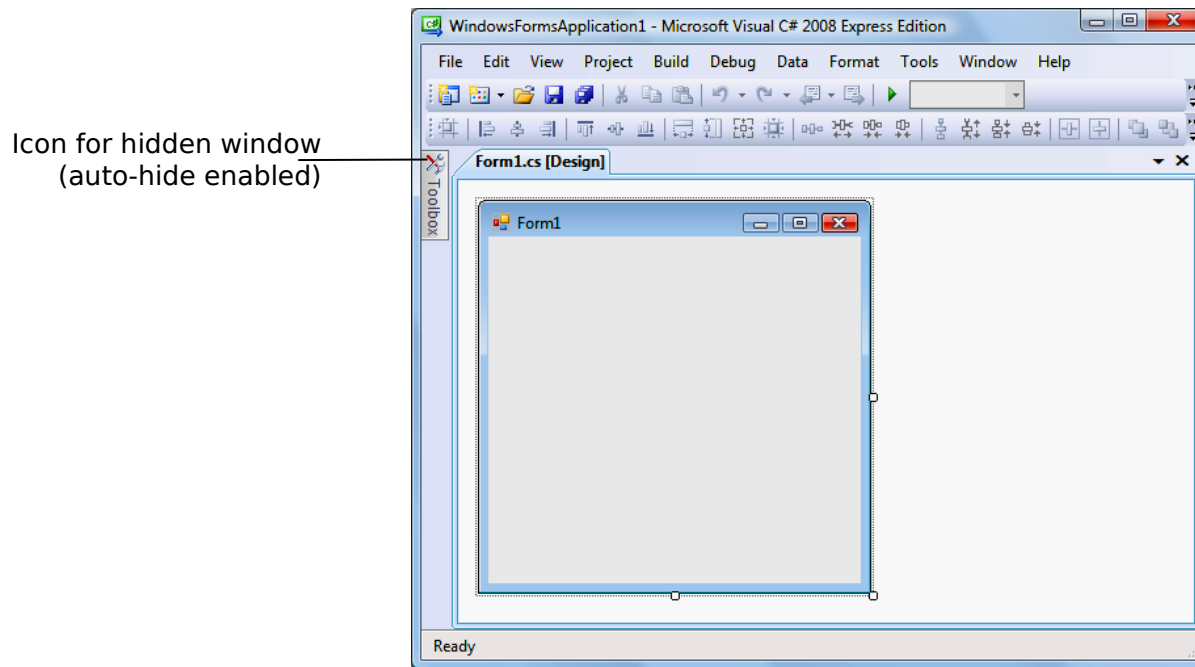
- The IDE provides windows for accessing project files and customizing controls.
- These windows can be accessed via the toolbar icons (Fig. 2.13) or the **View** menu.



**Fig. 2.13** | Toolbar icons for Visual Studio windows.

## 2.4 Navigating the Visual Studio IDE (Cont.)

- When **auto-hide** is enabled, a tab appears along the edge of the IDE window (Fig. 2.14).



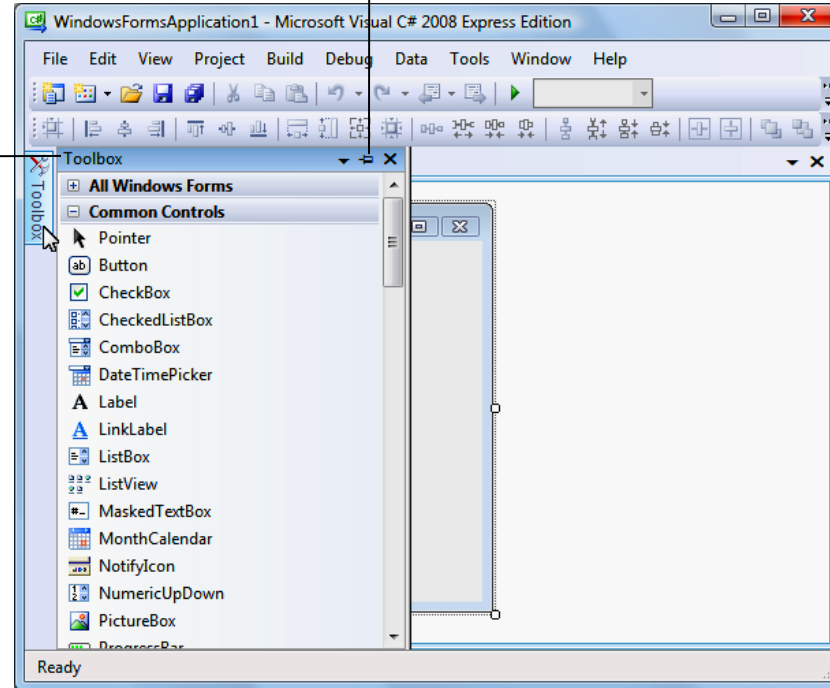
**Fig. 2.14** | Auto-hide feature demonstration.

## 2.4 Navigating the Visual Studio IDE (Cont.)

- Placing the mouse pointer over one of these icons displays that window (Fig. 2.15).

Horizontal orientation for pin icon when auto-hide is enabled

Toolbox title bar



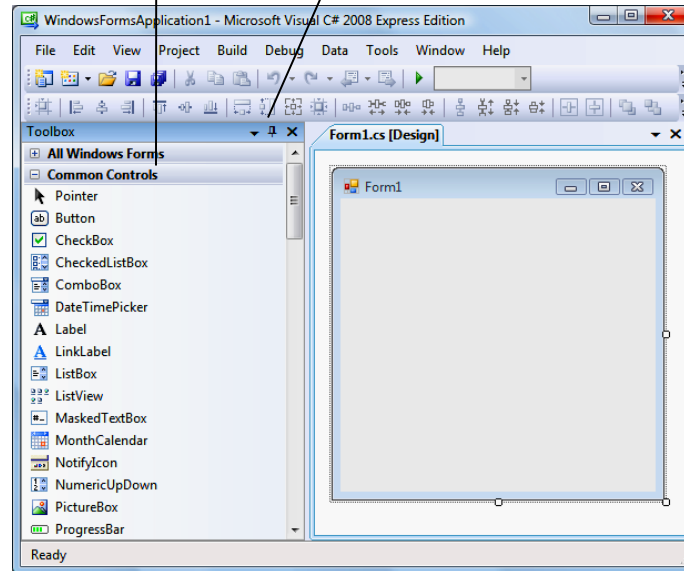
**Fig. 2.15** | Displaying a hidden window when auto-hide is enabled.



## 2.4 Navigating the Visual Studio IDE (Cont.)

- To disable auto-hide and keep the window open, click the pin icon in the window's upper-right corner.
- When a window is “pinned down,” the pin icon is vertical (Fig. 2.16).

Toolbox “pinned down” Vertical orientation for pin icon when window is pinned down



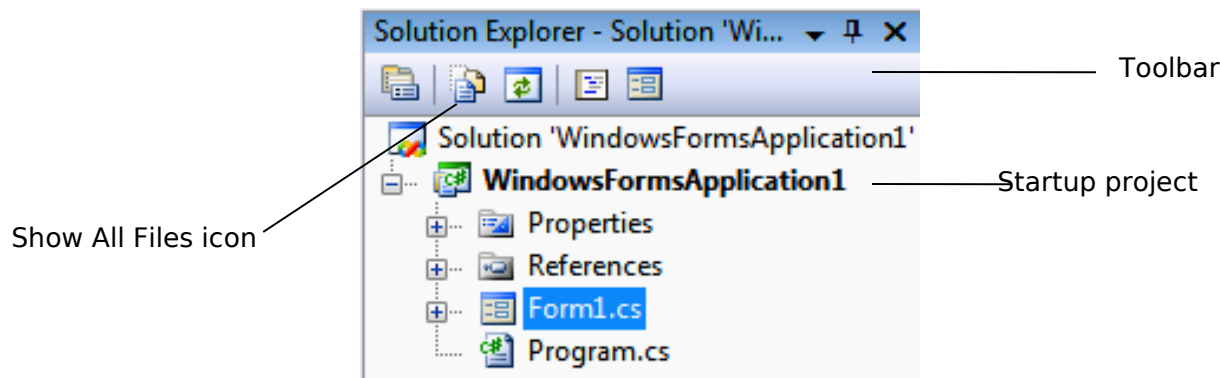
**Fig. 2.16** | Disabling auto-hide (“pinning down” a window).



## 2.4 Navigating the Visual Studio IDE (Cont.)

### 2.4.1 Solution Explorer

- The **Solution Explorer** window (Fig. 2.17) provides access to all of a solution's files.
- The solution's **startup project** runs when you select **Debug > Start Debugging**.
- The file that corresponds to the Form is named `Form1.cs`. Visual C# files use the `.cs` file-name extension.

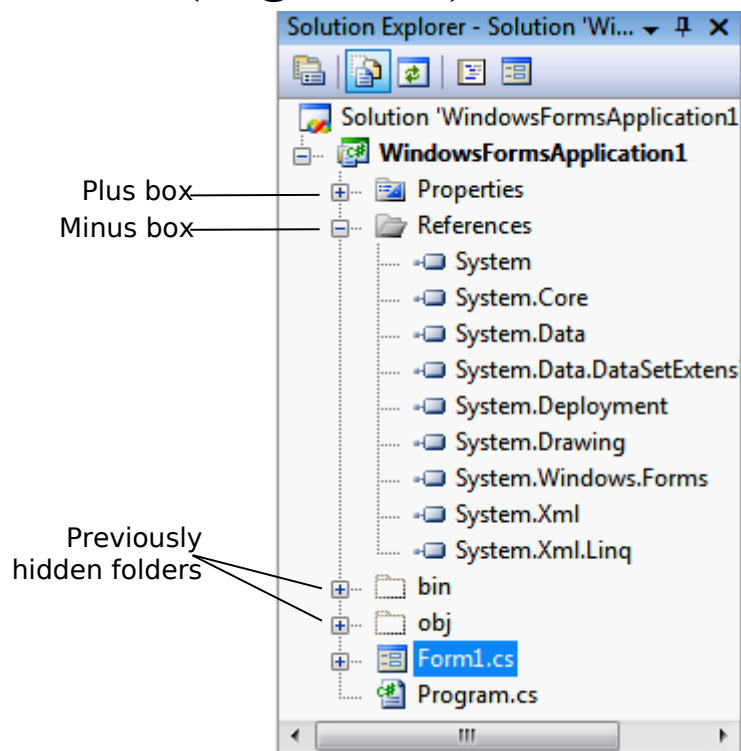


**Fig. 2.17 | Solution Explorer** with an open project.



## 2.4 Navigating the Visual Studio IDE (Cont.)

- Clicking the **Show All Files icon** displays all the files in the solution (Fig. 2.18).

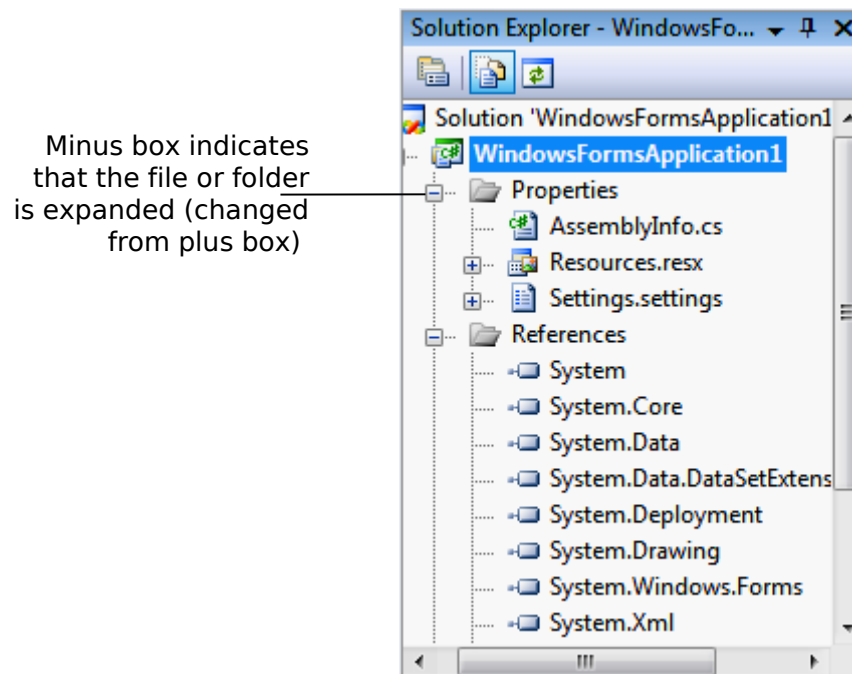


**Fig. 2.18 | Solution Explorer** showing plus boxes and minus boxes for expanding and collapsing the tree to reveal or hide project files, respectively.



## 2.4 Navigating the Visual Studio IDE (Cont.)

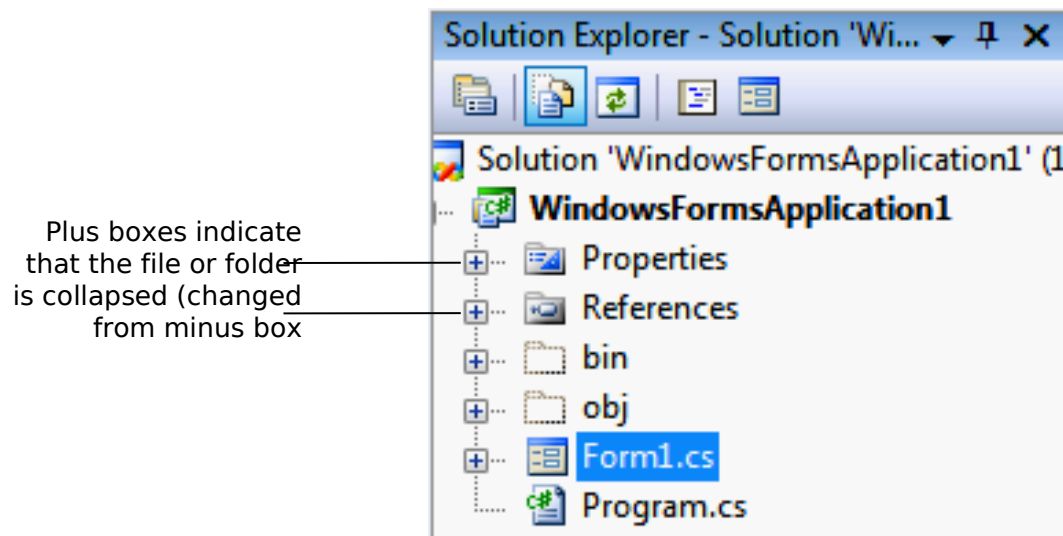
- Click a plus box to display grouped items (Fig. 2.19)



**Fig. 2.19** | **Solution Explorer** expanding the **Properties** file after you click its plus box.

## 2.4 Navigating the Visual Studio IDE (Cont.)

- Click the minus box to collapse the tree from its expanded state (Fig. 2.20).



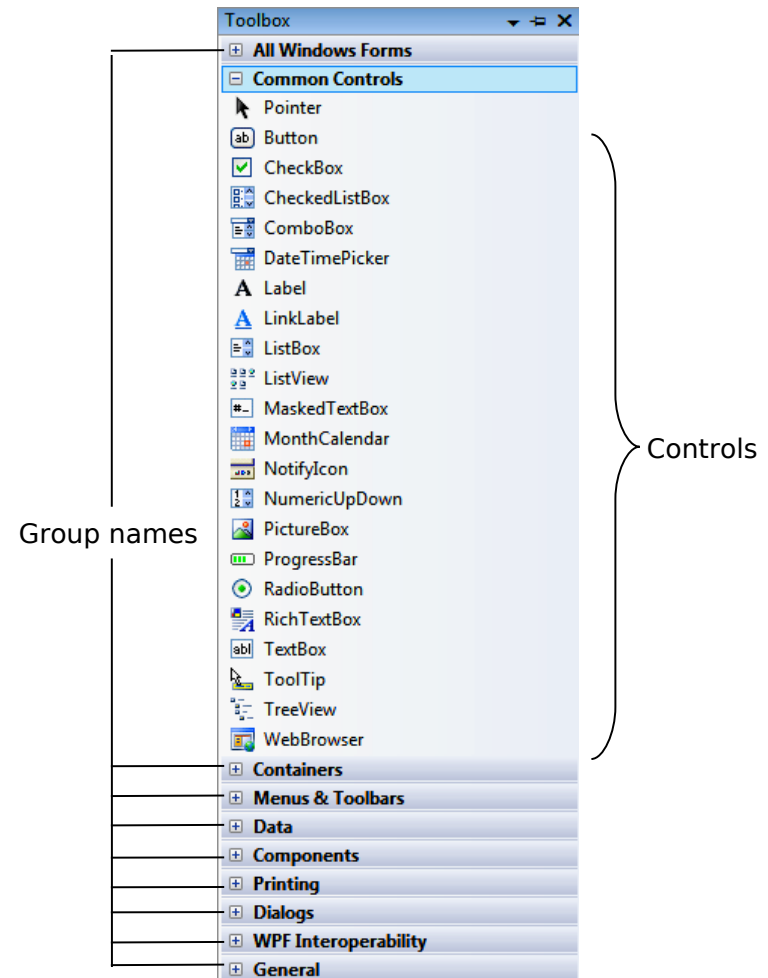
**Fig. 2.20** | **Solution Explorer** collapsing all files after you click any minus boxes.



## 2.4 Navigating the Visual Studio IDE (Cont.)

### 2.4.2 Toolbox

- The **Toolbox** contains icons representing controls used to customize Forms (Fig. 2.21).
- The **Toolbox** groups the prebuilt controls into categories.



**Fig. 2.21** | **Toolbox** window displaying controls for the **Common Controls** group.

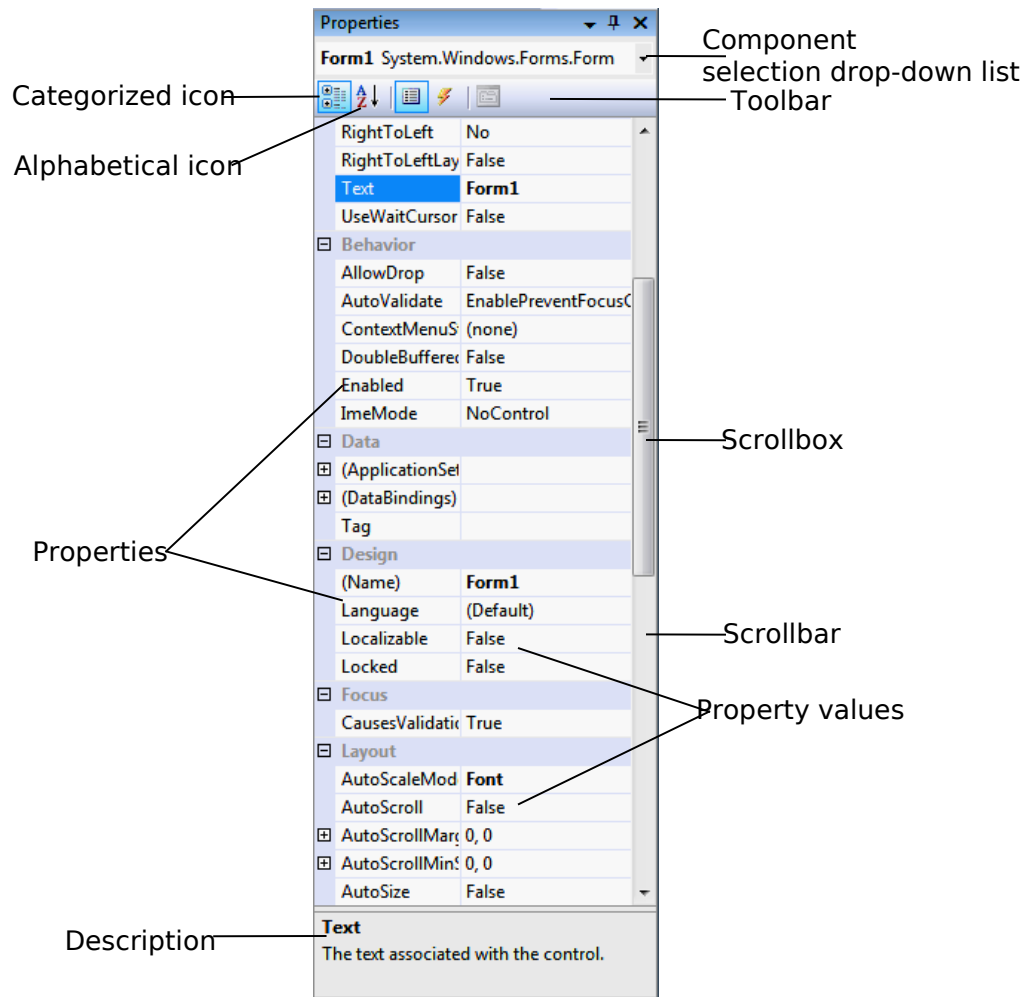
## 2.4 Navigating the Visual Studio IDE (Cont.)

### 2.4.2 Toolbox

- To display the **Properties** window, select **View > Properties Window**.
- The **Properties window** allows you to modify a control's properties visually, without writing code (Fig. 2.22).



## 2.4 Navigating the Visual Studio IDE (Cont.)



**Fig. 2.22 | Properties** window showing the description of the selected property.



## 2.5 Using Help

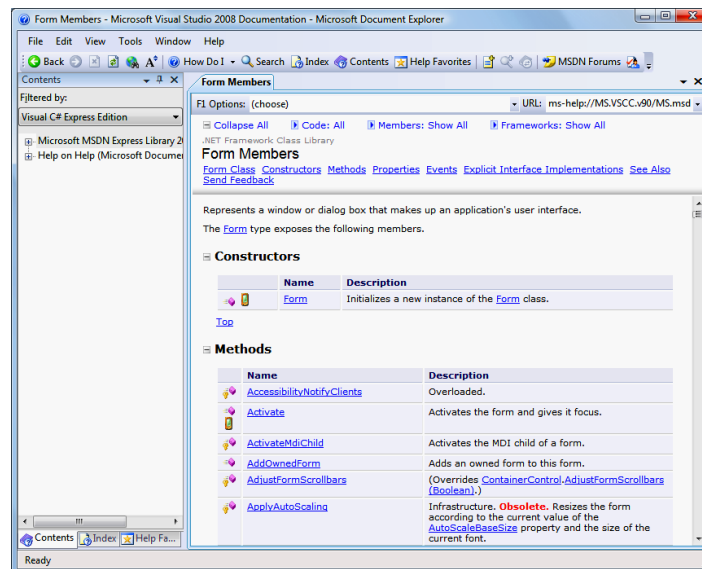
- The **Help menu** commands are summarized in Fig. 2.23.

Command	Description
<b>How Do I?</b>	Contains links to relevant topics, including how to upgrade programs and learn more about web services, architecture and design, files and I/O, data, debugging and more.
<b>Search</b>	Finds help articles based on search keywords.
<b>Index</b>	Displays an alphabetized list of topics.
<b>Contents</b>	Displays a categorized table of contents in which help articles are organized by topic.

**Fig. 2.23 | Help menu commands.**

## 2.5 Using Help (Cont.)

- **Context-sensitive help** displays relevant help articles rather than a generalized list (Fig. 2.24).
- To use context-sensitive help, click an item, then press the *F1* key.



**Fig. 2.24** | Using context-sensitive help to show help articles related to a Form.

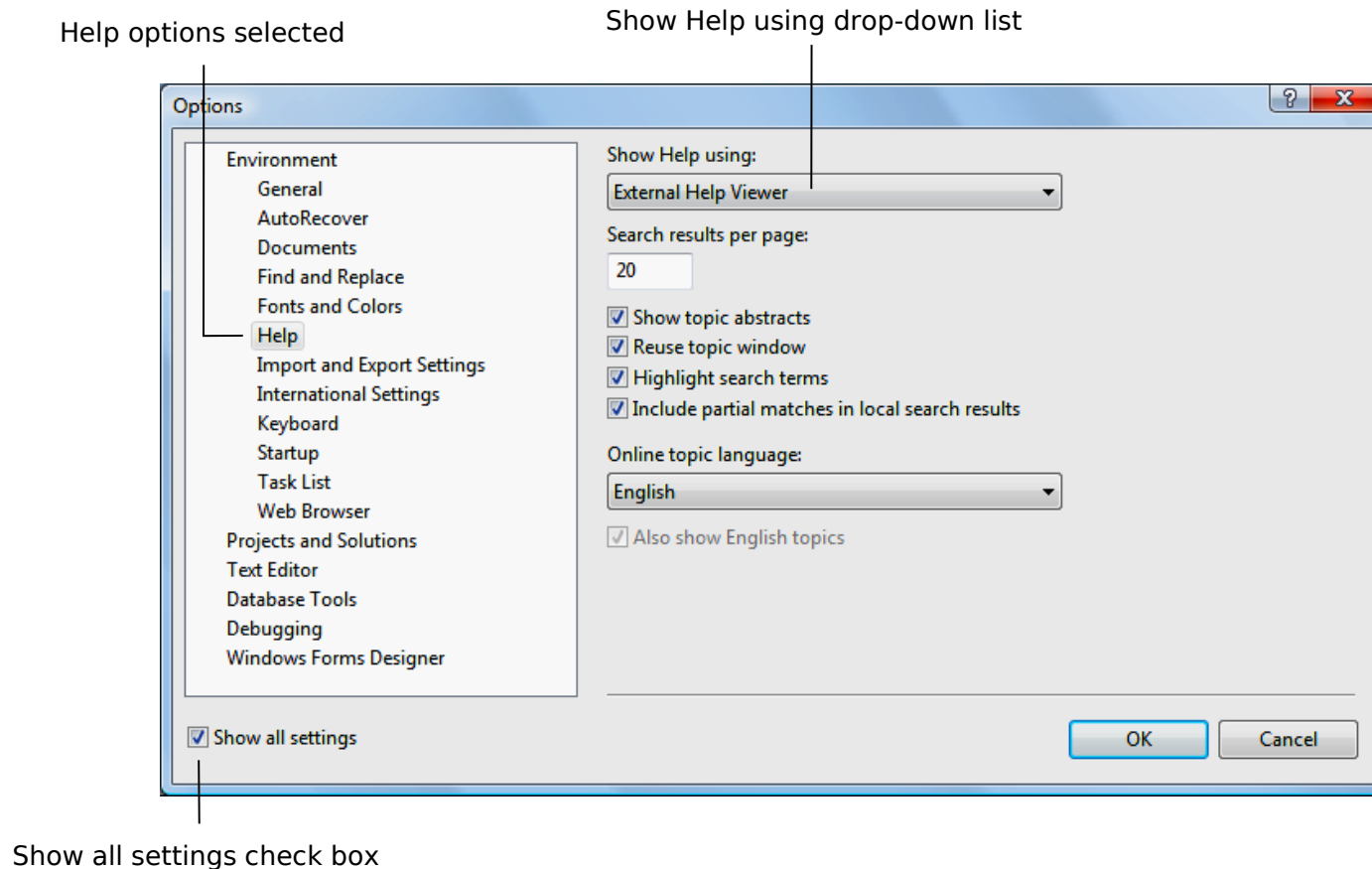


## 2.5 Using Help (Cont.)

- Select **Tools > Options...**
- Make sure that the **Show all settings** checkbox is checked (Fig. 2.25).
- Select **Help** on the left, then locate the **Show Help using:** drop-down list.
  - **External Help Viewer** displays articles in a separate window
  - **Integrated Help Viewer** displays a help article inside the IDE.



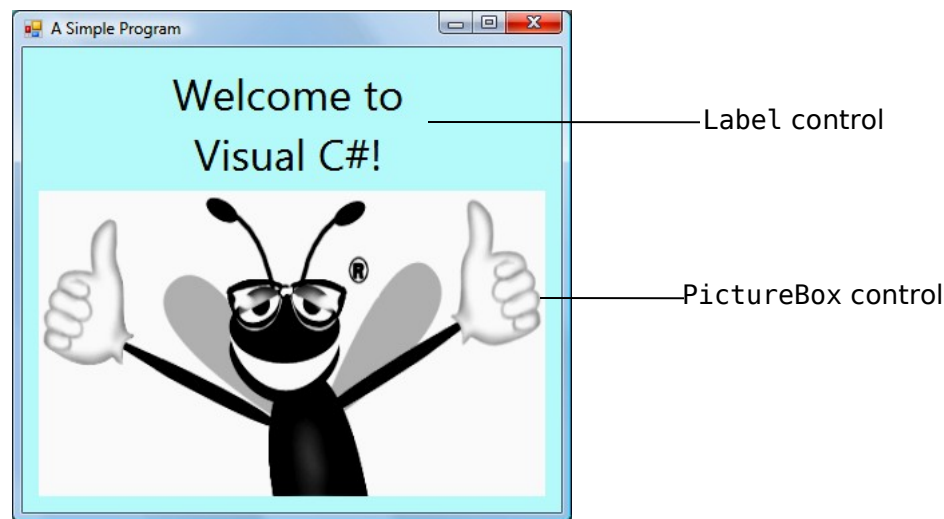
## 2.5 Using Help (Cont.)



**Fig. 2.25** | **Options** dialog displaying **Help** settings.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image

- Visual C# has preexisting controls used to build and customize programs (Fig. 2.26).
- A `Label` contains descriptive text.
- A `PictureBox` displays an image, such as the Deitel bug mascot.



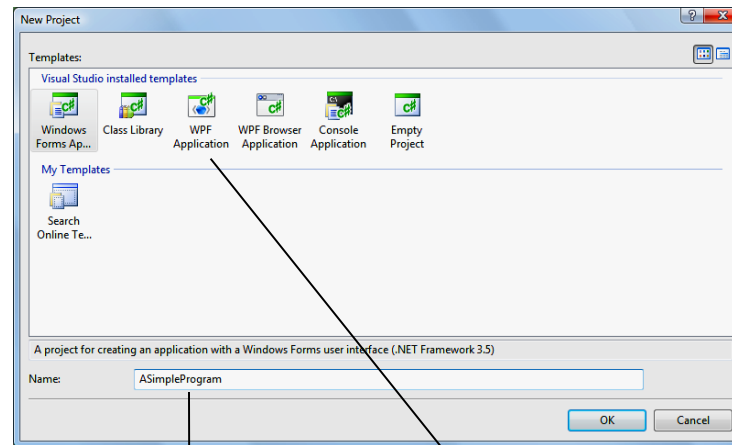
**Fig. 2.26** | Simple program executing.





## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Select **File > New Project...** and create a new **Windows Forms Application** (Fig. 2.27).
- Name the project **ASimpleProgram** and click **OK**.

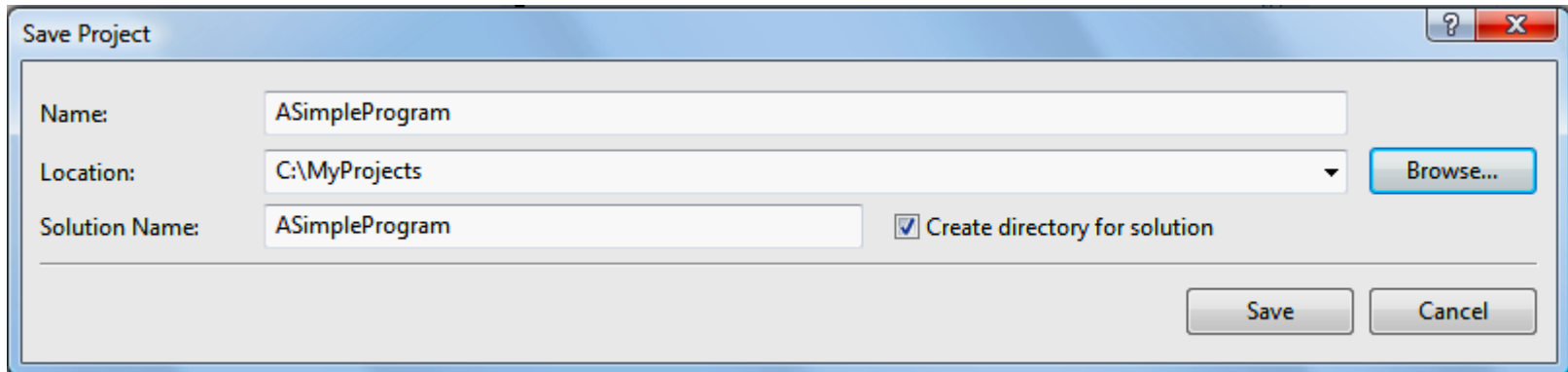


Type the project name    Template types

**Fig. 2.27 | New Project dialog.**

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Select **File > Save All** to display the **Save Project dialog** (Fig. 2.28).



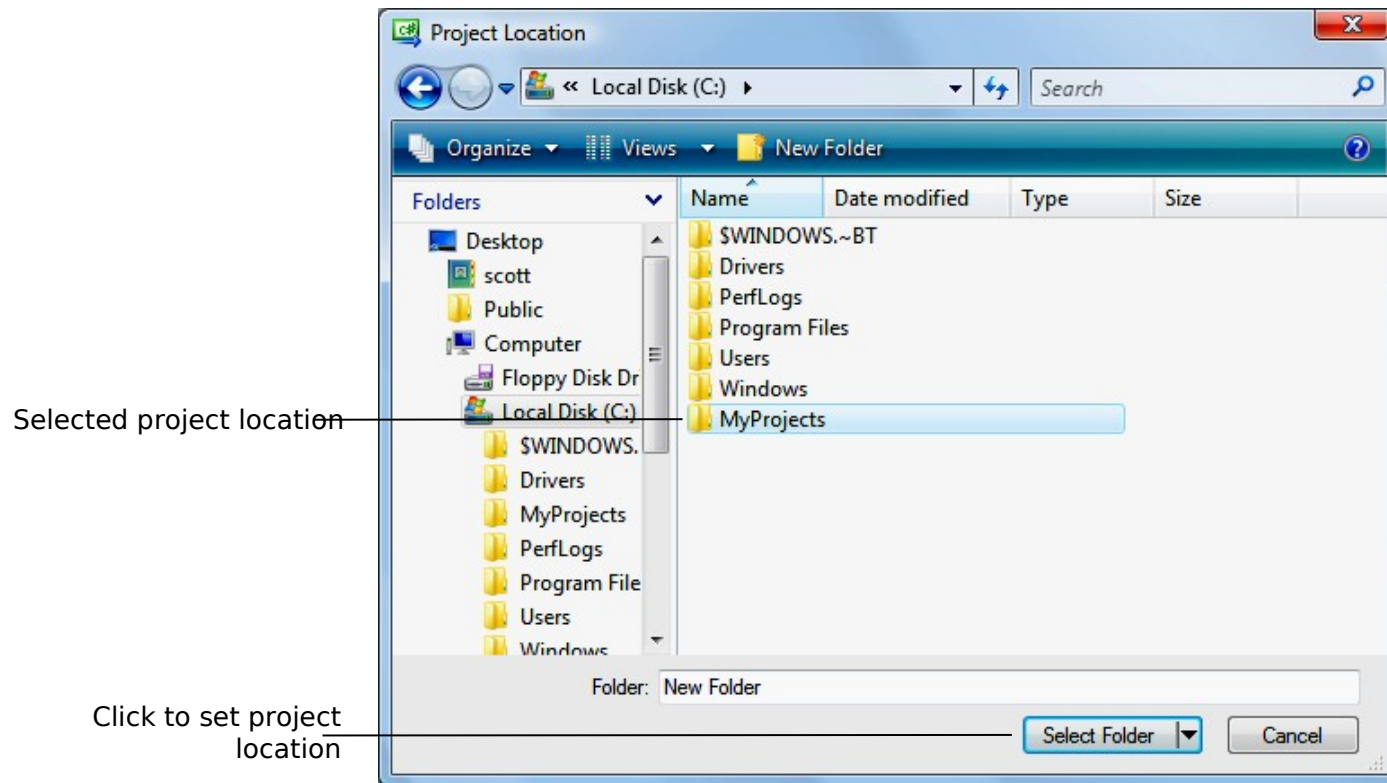
**Fig. 2.28 | Save Project** dialog.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Click the **Browse...** button, which opens the **Project Location dialog** (Fig. 2.29).
- Navigate through the directories and select one in which to place the project.



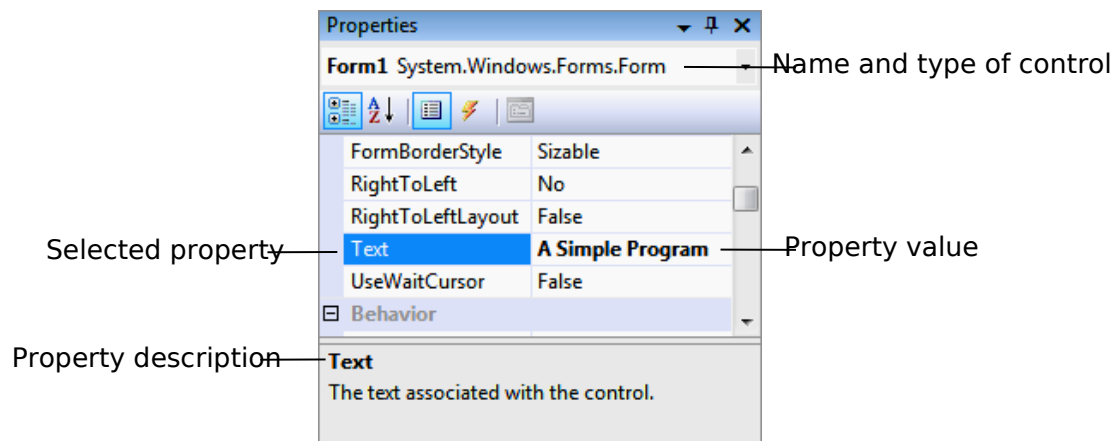
## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)



**Fig. 2.29** | Setting the project location in the **Project Location** dialog.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Click anywhere in the **Form** to display the Form's properties in the **Properties** window.
- Click in the textbox to the right of the **Text** property box and type "A Simple Program" (Fig. 2.30).

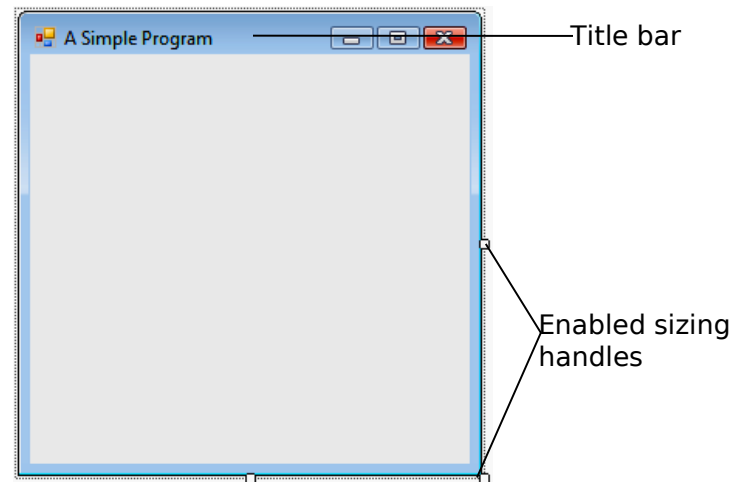


**Fig. 2.30** | Setting the Form's Text property in the **Properties** window.



## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

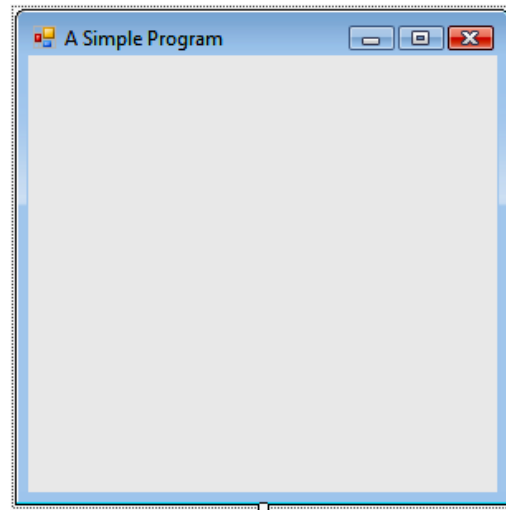
- Press *Enter*—the Form's title bar is updated immediately (Fig. 2.31).
- Resize the Form by clicking and dragging one of the **enabled sizing handles** (Fig. 2.31).



**Fig. 2.31** | Form with enabled sizing handles.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

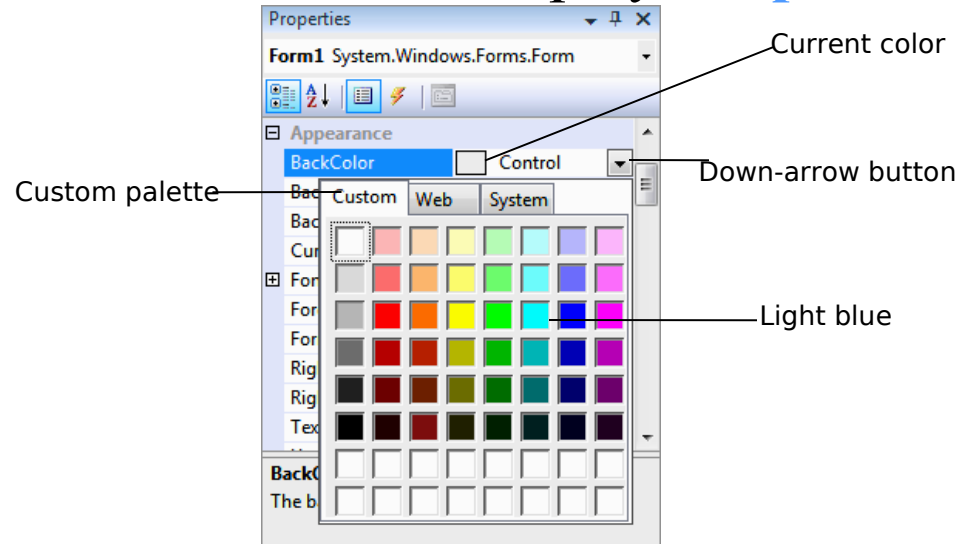
- Select the bottom-right sizing handle and drag it down and to the right to make the **Form** larger (Fig. 2.32).
- You can also resize a **Form** by setting its **Size** property.



**Fig. 2.32** | Resized Form.

## 2.6 Using Visual Programming to Create a Simple Program (Cont.)

- Clicking **BackColor** in the **Properties** window causes a down-arrow button to appear (Fig. 2.33).
- When clicked, the arrow displays tabs for **Custom**, **Web** and **System** colors.
- Click the **Custom** tab to display the **palette** and select light blue.



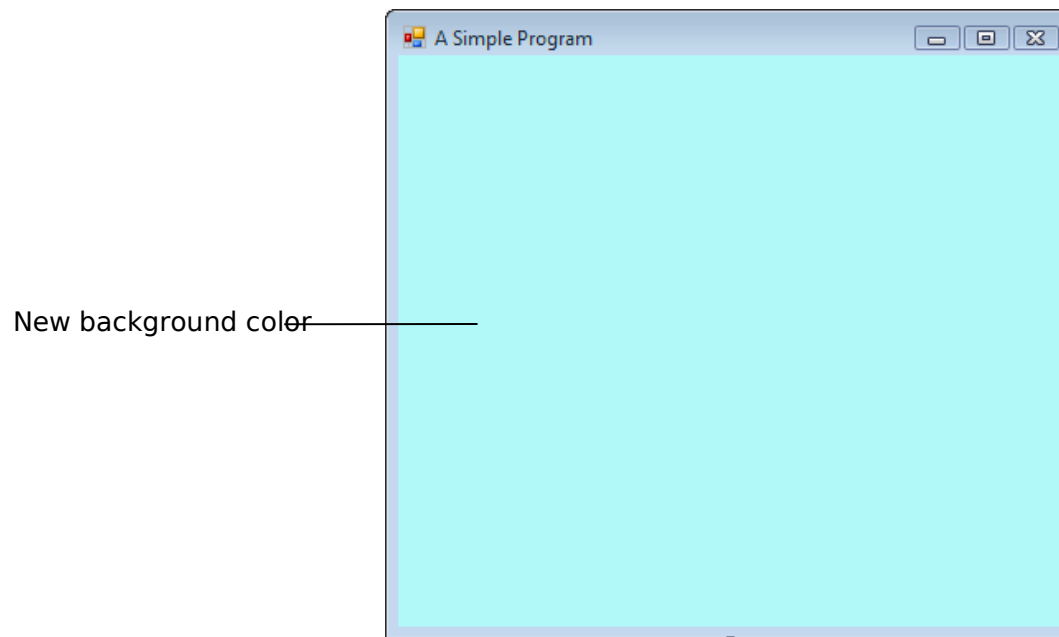
**Fig. 2.33** | Changing the Form's BackColor property.





## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

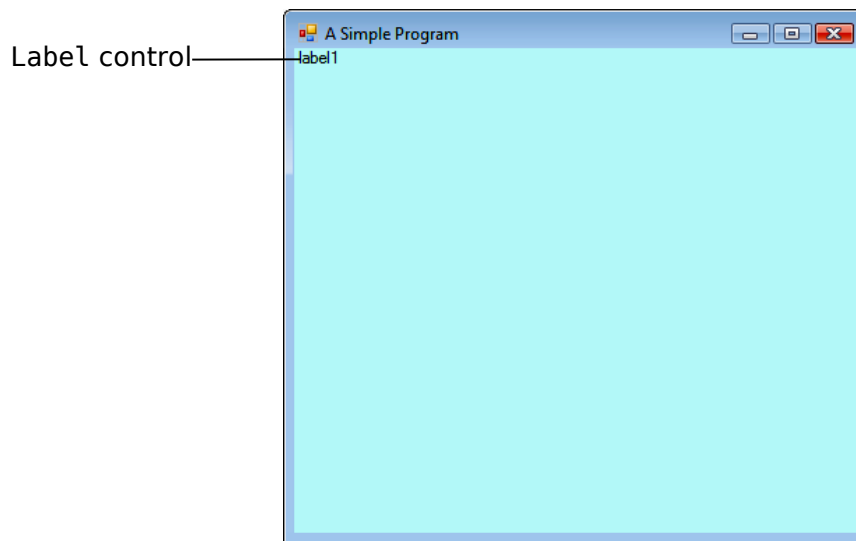
- Once you select the color, the Form's background changes to light blue (Fig. 2.34).



**Fig. 2.34** | Form with new BackColor property applied.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Double click the `Label` control in the **Toolbox** to add a `Label` (Fig. 2.35).
- You also can “drag” controls from the **Toolbox** to the Form.



**Fig. 2.35** | Adding a `Label` to the Form.

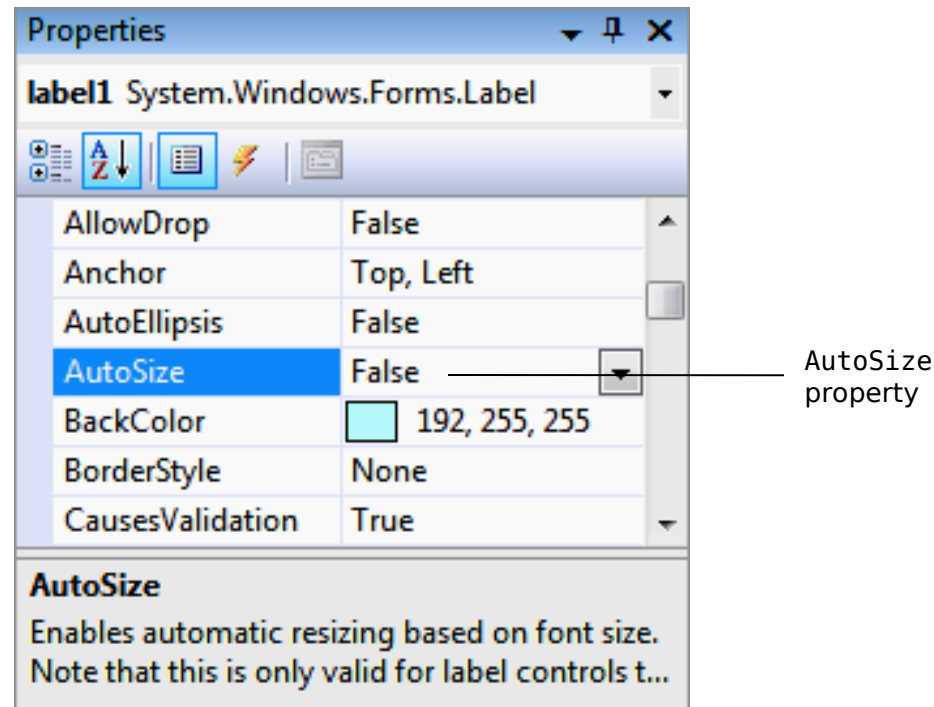


## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Select the `Label` to make its properties appear in the **Properties** window (Fig. 2.36).
- Set the `Label`'s `Text` property to **Welcome to Visual C#!**.
- The **AutoSize property** is set to `True`, which allows the `Label` to resize to fit its text.
- Set the `AutoSize` property to `False`.



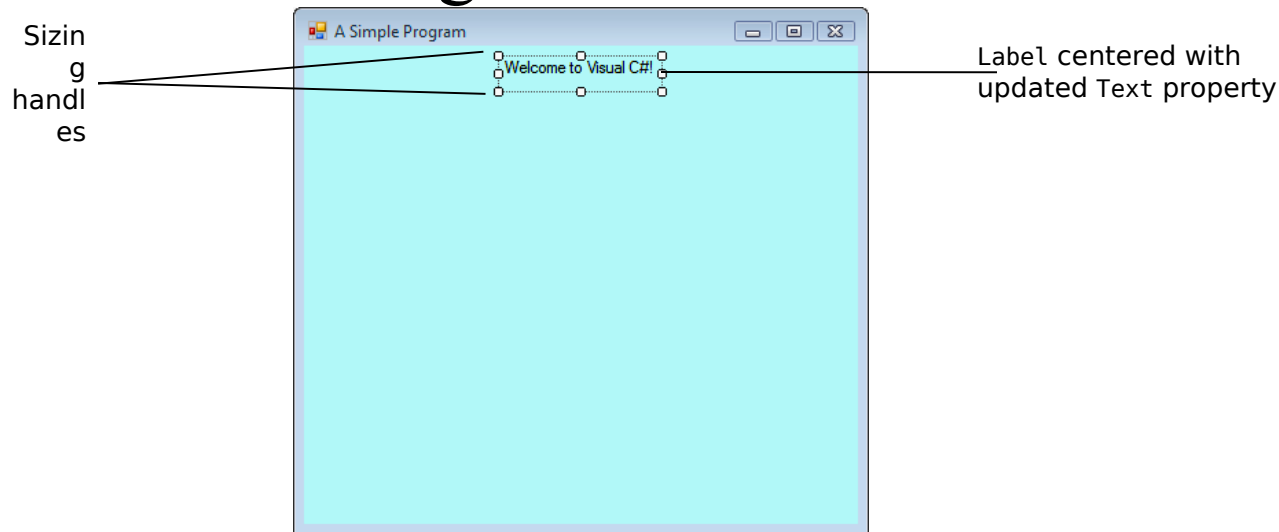
## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)



**Fig. 2.36** | Changing the Label's AutoSize property to False.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Move the `Label` by dragging it or by using the left and right arrow keys (Fig. 2.37).
- When the `Label` is selected, you can also center the `Label` using the **Format** menu.

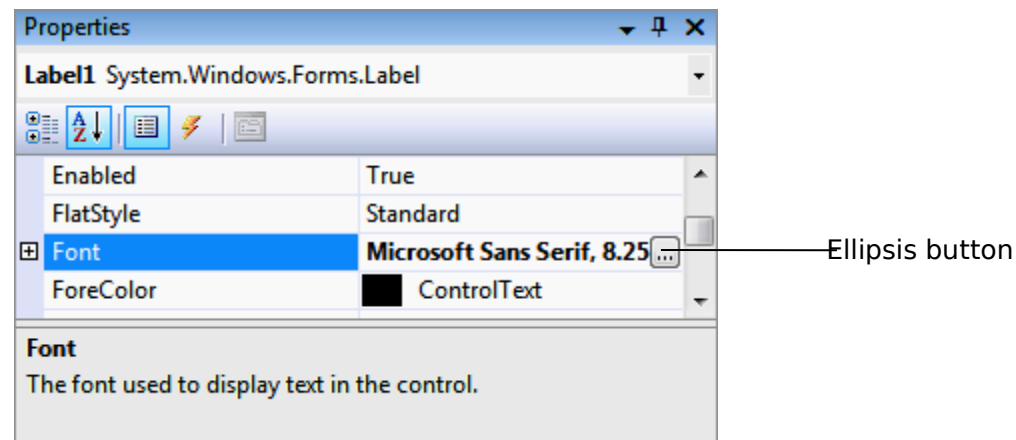


**Fig. 2.37** | GUI after the Form and Label have been customized.



## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

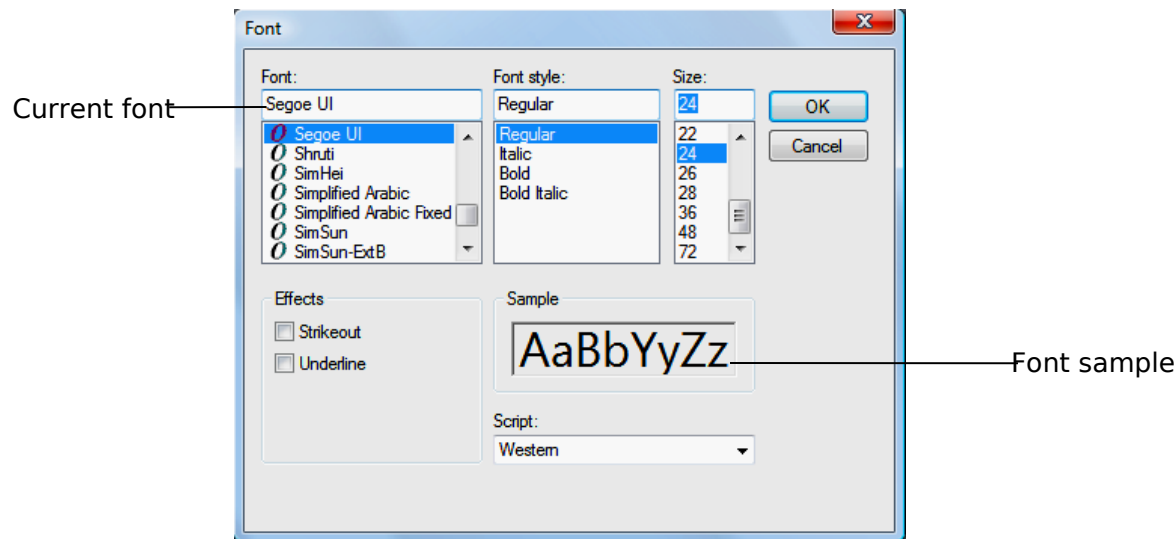
- To change the font of the Label's text, select the **Font property** (Fig. 2.38).
- When the ellipsis button is clicked, a dialog appears that provides additional values.



**Fig. 2.38** | **Properties** window displaying the Label's properties.

## 2.6 Using Visual Programming to Create a Simple Program (Cont.)

- The **Font dialog** (Fig. 2.39) allows you to select the font name, style and size.
- Under **Font**, select **Segoe UI**. Under **Size**, select **24** points and click **OK**.
- Resize the **Label** if it's not large enough to hold the text.

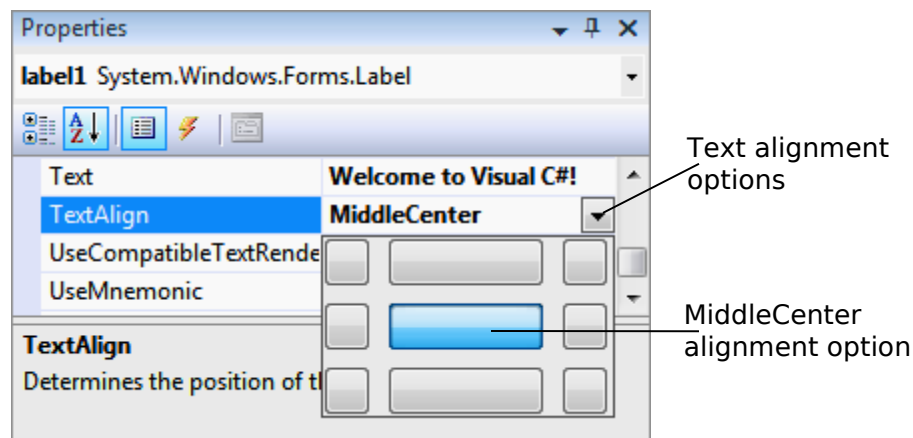


**Fig. 2.39** | **Font** dialog for selecting fonts, styles and sizes.



## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Select the Label's **TextAlign** property (Fig. 2.40).
- Set the Text -Align property to **MiddleCenter**.

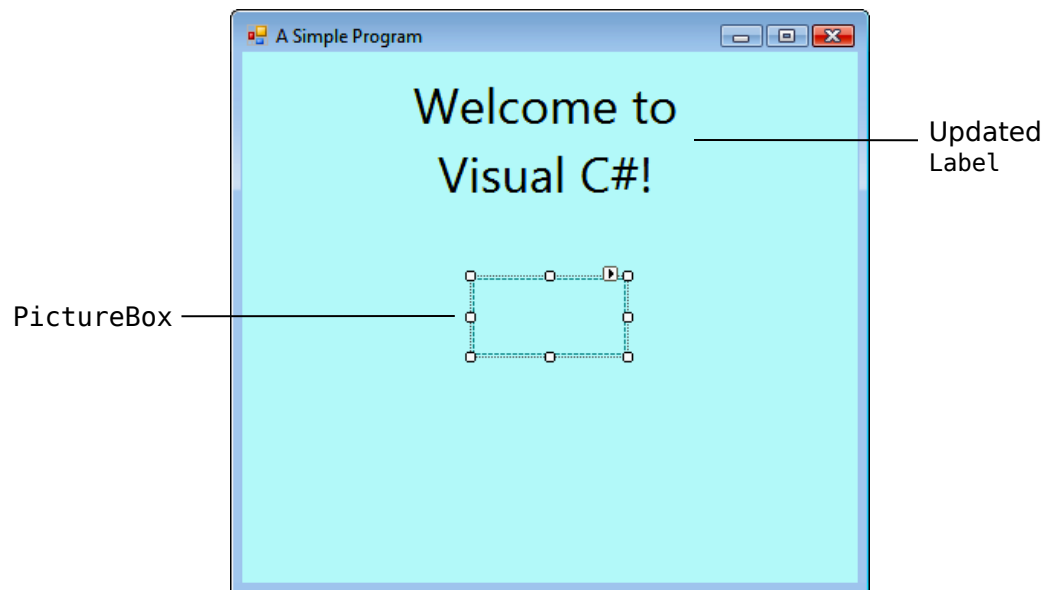


**Fig. 2.40** | Centering the Label's text.



## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

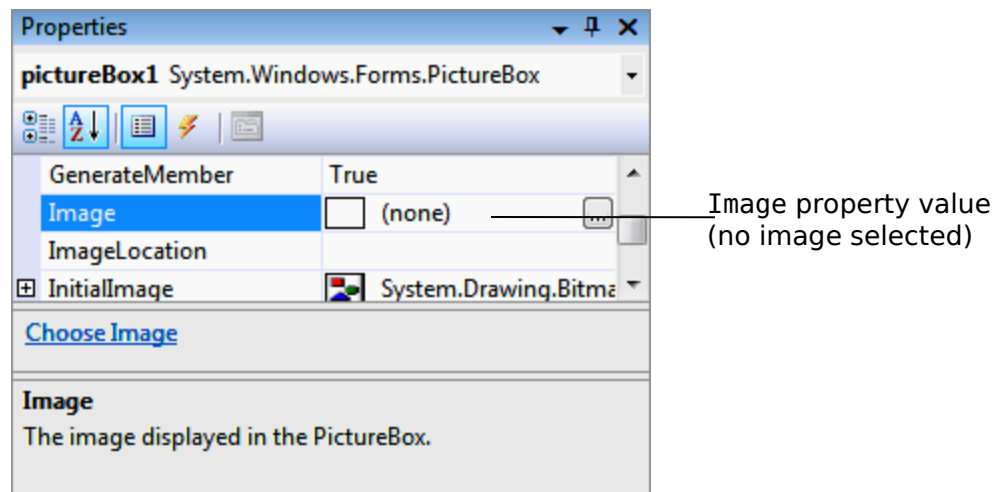
- Locate the **PictureBox** in the **Toolbox** and add it to the **Form** (Fig. 2.41).



**Fig. 2.41** | Inserting and aligning a PictureBox.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

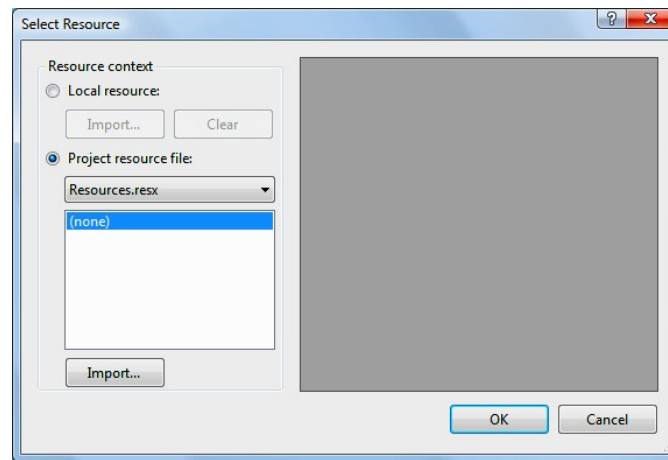
- Click the `PictureBox` to display its properties in the **Properties** window (Fig. 2.42).
- The **Image** property displays a preview of the image, if one exists.



**Fig. 2.42** | Image property of the PictureBox.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- Click the ellipsis button to display the **Select Resource dialog**- (Fig. 2.43).
- Click the **Import...** button to browse for the image to insert (bug . png)

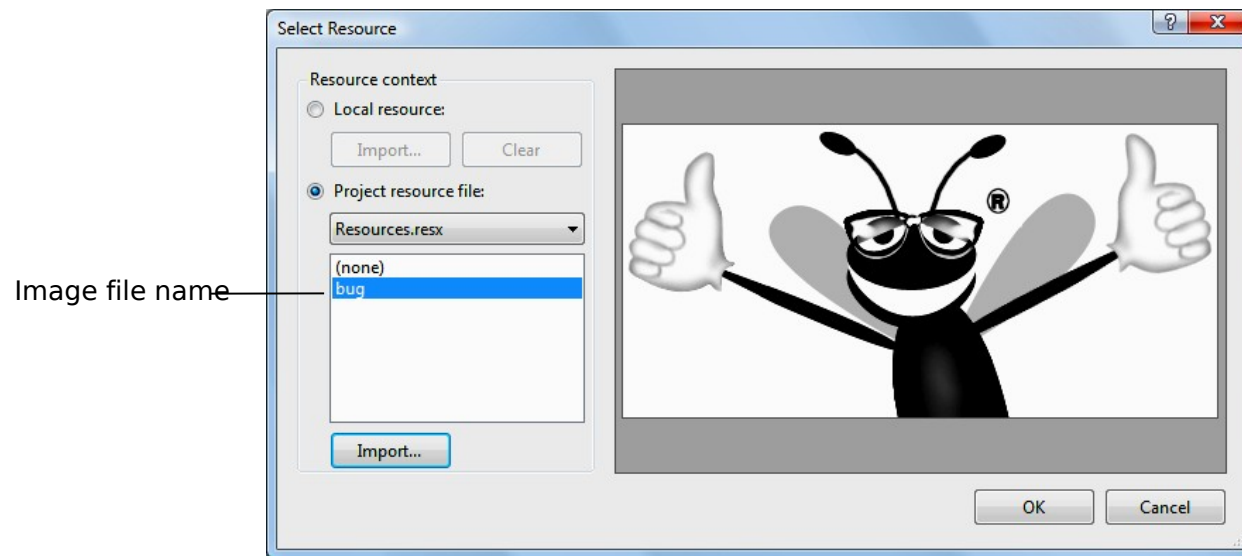


**Fig. 2.43** | **Select Resource** dialog to select an image for the PictureBox.



## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- In the dialog that appears, locate the image file, select it and click **OK** (Fig. 2.44).
- Click **OK** to place the image in your program.

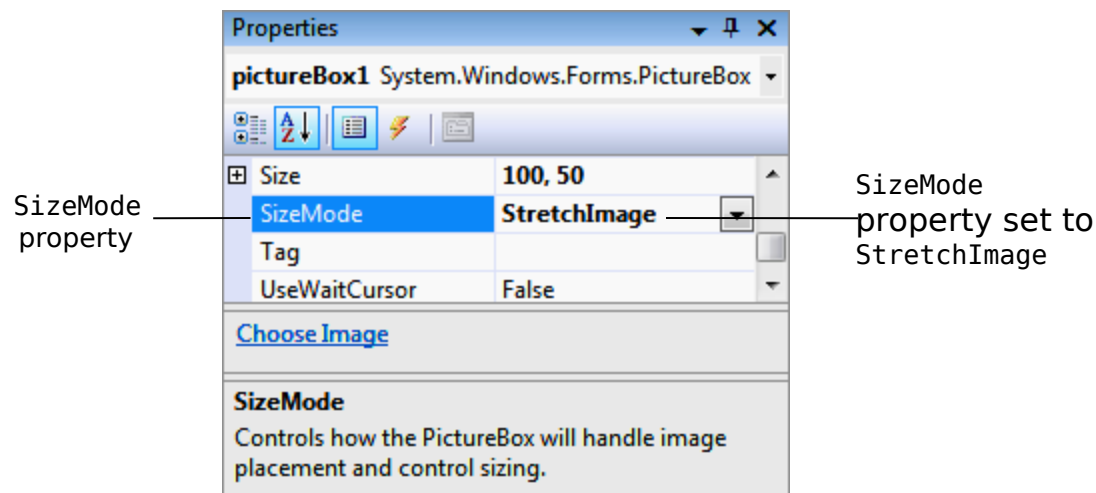


**Fig. 2.44** | **Select Resource** dialog displaying a preview of selected image.



## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

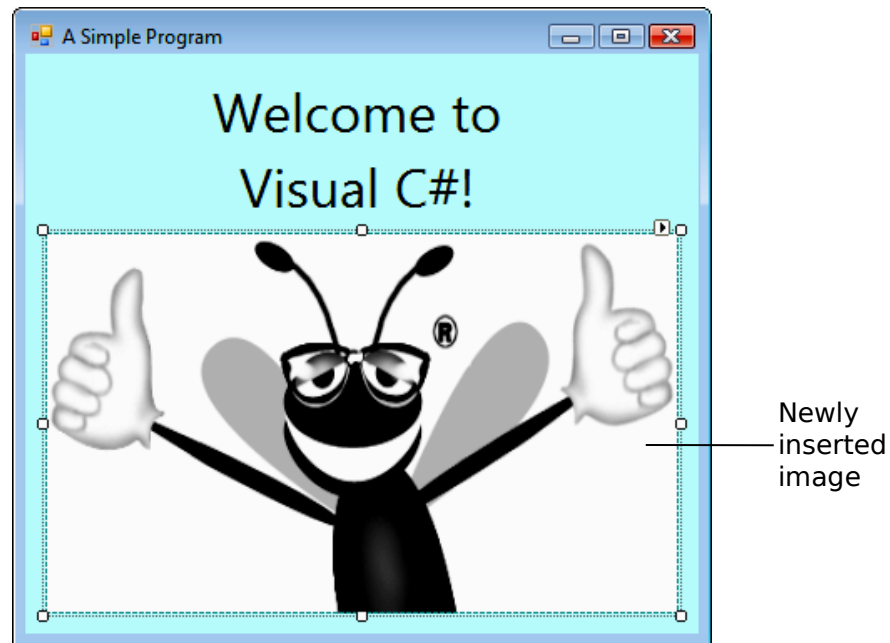
- To size the image to the PictureBox, change the **SizeMode** property to **StretchImage** (Fig. 2.45).



**Fig. 2.45** | Scaling an image to the size of the PictureBox.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

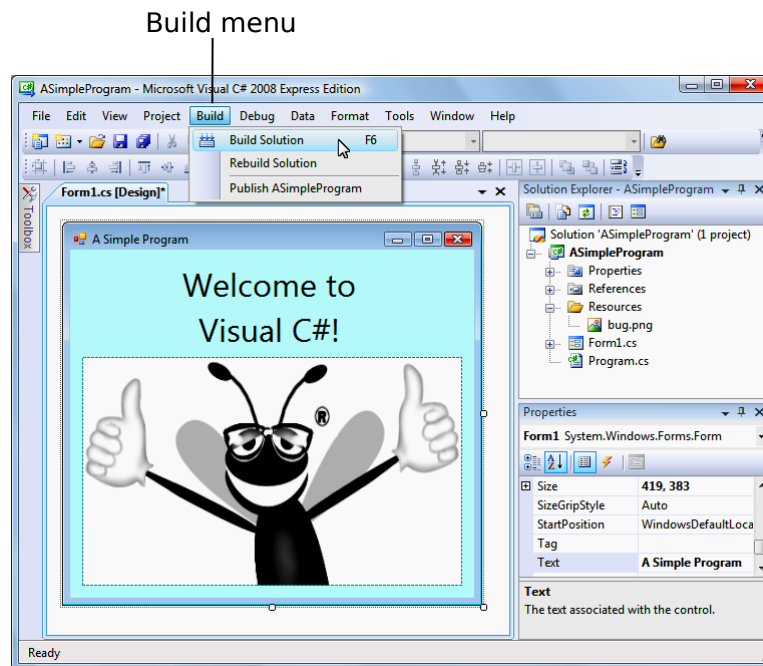
- Resize the PictureBox, making it larger (Fig. 2.46).
- Select **File > Save All** to save the entire solution.



**Fig. 2.46** | PictureBox displaying an image.

## 2.6 Using Visual Programming to Create a Simple Program that Displays Text and an Image (Cont.)

- In **run mode**, the program is executing, and some features are disabled (Fig. 2.47).



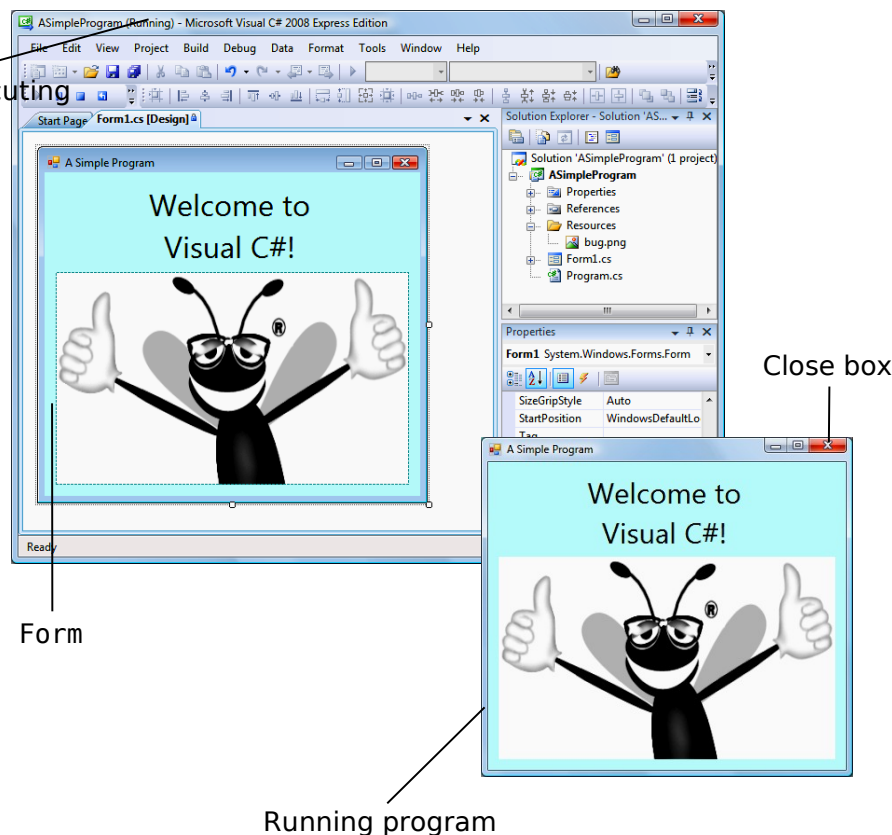
**Fig. 2.47** | Building a solution.



## 2.6 Using Visual Programming to Create a Simple Program (Cont.)

- Select **Debug > Start Debugging** to execute the program (Fig. 2.48).

IDE displays text Running, which signifies that the program is executing



**Fig. 2.48** | IDE in run mode, with the running program in the foreground.

