Microsoft Visual Basic 2015: Reloaded

Sixth Edition

Chapter Two
Creating a User Interface
Objectives

After studying this chapter, you should be able to:
• Plan an application using a TOE chart
• Use a text box
• Follow the Windows standards regarding the layout and labeling of controls
• Follow the Windows standards regarding the use of graphics, fonts, and color
• Assign access keys to controls
Objectives (cont'd.)

- Set the tab order
- Designate a default button
- Print the interface from code
- Play an audio file
Planning an Application

**HOW TO** Plan an Application

1. Identify the tasks the application needs to perform.
2. Identify the objects to which you will assign those tasks.
3. Identify the events required to trigger an object to perform its assigned tasks.
4. Design the user interface.

Figure 2-1: How to plan an application
Say Cheese! Company

• Orders for cheesecakes taken by phone
• Cheesecakes priced at $25 each – available in two flavors:
  • Vanilla bean and strawberry
• Salespeople record each order on a form that contains the customer’s name and address, and the number of each flavor of cheesecake ordered
• Salespeople then calculate the total number of cheesecakes ordered and the total price of the order, including a 3% sales tax
• Sales manager feels that having the salespeople manually perform the necessary calculations is much too time-consuming and prone to errors
Say Cheese! Company (cont.)

• **Identifying the Application’s Tasks**
  • What information will the application need to display on the screen and/or print on the printer?
  • What information will the user need to enter into the user interface to display and/or print the desired information?
  • What information will the application need to calculate to display and/or print the desired information?
  • How will the user end the application?
  • Will previous information need to be cleared from the screen before new information is entered?
Say Cheese! Company (cont’d)

---

**ORDER FORM**

Customer name: ________________________________________________

Address: _______________________________________________________

City: ___________________________  State: ____  ZIP: ________________

---

<table>
<thead>
<tr>
<th>VANILLA BEAN</th>
<th>STRAWBERRY</th>
<th>TOTAL ORDERED</th>
<th>TOTAL PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 2-2: Current Order Form
<table>
<thead>
<tr>
<th>Task</th>
<th>Object</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get the following order information from the user:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Customer name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Street address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- City</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ZIP code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of vanilla bean ordered</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of strawberry ordered</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculate total ordered and total price</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display the following information:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Customer name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Street address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- City</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ZIP code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of vanilla bean ordered</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of strawberry ordered</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Total ordered</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Total price</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print the order form</td>
<td></td>
<td></td>
</tr>
<tr>
<td>End the application</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clear the screen for the next order</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2-3: Tasks entered in a TOE chart
Say Cheese! Company (cont’d)

• Identifying the Objects
  • A label control displays information that you do not want the user to change while the application is running
  • A button control performs an action immediately after the user clicks it
  • A **text box** gives the user an area in which to enter data
### Say Cheese! Company (cont'd)

<table>
<thead>
<tr>
<th>Task</th>
<th>Object</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get the following order information from the user:</td>
<td>nameTextBox</td>
<td></td>
</tr>
<tr>
<td>Customer name</td>
<td>addressTextBox</td>
<td></td>
</tr>
<tr>
<td>Street address</td>
<td>cityTextBox</td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>stateTextBox</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>zipTextBox</td>
<td></td>
</tr>
<tr>
<td>ZIP code</td>
<td>vanillaTextBox</td>
<td></td>
</tr>
<tr>
<td>Number of vanilla bean ordered</td>
<td>strawberryTextBox</td>
<td></td>
</tr>
<tr>
<td>Number of strawberry ordered</td>
<td>calcButton</td>
<td></td>
</tr>
<tr>
<td>Calculate total ordered and total price</td>
<td>calcButton</td>
<td></td>
</tr>
</tbody>
</table>

Display the following information:

<table>
<thead>
<tr>
<th>Task</th>
<th>Object</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer name</td>
<td>nameTextBox</td>
<td></td>
</tr>
<tr>
<td>Street address</td>
<td>addressTextBox</td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>cityTextBox</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>stateTextBox</td>
<td></td>
</tr>
<tr>
<td>ZIP code</td>
<td>zipTextBox</td>
<td></td>
</tr>
<tr>
<td>Number of vanilla bean ordered</td>
<td>vanillaTextBox</td>
<td></td>
</tr>
<tr>
<td>Number of strawberry ordered</td>
<td>strawberryTextBox</td>
<td></td>
</tr>
<tr>
<td>Total ordered (from calcButton)</td>
<td>totalOrderedLabel</td>
<td></td>
</tr>
<tr>
<td>Total price (from calcButton)</td>
<td>totalPriceLabel</td>
<td></td>
</tr>
<tr>
<td>Print the order form</td>
<td>printButton</td>
<td></td>
</tr>
<tr>
<td>End the application</td>
<td>exitButton</td>
<td></td>
</tr>
<tr>
<td>Clear the screen for the next order</td>
<td>clearButton</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2-4: Tasks and objects entered in a TOE chart**
Say Cheese! Company (cont’d)

• Identifying the Events
  • After defining the application’s tasks and assigning the tasks to objects in the interface, you then determine which event (if any) must occur for an object to carry out its assigned task
  • The seven text boxes are assigned the task of getting and displaying the order information
  • The two label controls are assigned the task of displaying the total number of cheesecakes ordered and the total price of the order
  • The remaining objects are four buttons
    • You will have each button perform its assigned task(s) when the user clicks it
### Task

Get the following order information from the user:

- Customer name
- Street address
- City
- State
- ZIP code
- Number of vanilla bean ordered
- Number of strawberry ordered

Calculate total ordered and total price

Display the following information:

- Customer name
- Street address
- City
- State
- ZIP code
- Number of vanilla bean ordered
- Number of strawberry ordered
- Total ordered (from calcButton)
- Total price (from calcButton)

Print the order form

End the application

Clear the screen for the next order

### Object

- nameTextBox
- addressTextBox
- cityTextBox
- stateTextBox
- zipTextBox
- vanillaTextBox
- strawberryTextBox
- calcButton
- nameTextBox
- addressTextBox
- cityTextBox
- stateTextBox
- zipTextBox
- vanillaTextBox
- strawberryTextBox
- totalOrderedLabel
- totalPriceLabel
- printButton
- exitButton
- clearButton

### Event

- None
- None
- None
- None
- None
- None
- None
- None
- None
- Click
- Click
- Click

---

**Figure 2-5: Completed TOE chart ordered by task**
### Say Cheese! Company (cont’d)

<table>
<thead>
<tr>
<th>Task</th>
<th>Object</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Calculate total ordered and total price</td>
<td>calcButton</td>
<td>Click</td>
</tr>
<tr>
<td>2. Display total ordered and total price in totalOrderedLabel and totalPriceLabel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print the order form</td>
<td>printButton</td>
<td>Click</td>
</tr>
<tr>
<td>End the application</td>
<td>exitButton</td>
<td>Click</td>
</tr>
<tr>
<td>Clear the screen for the next order</td>
<td>clearButton</td>
<td>Click</td>
</tr>
<tr>
<td>Display total ordered (from calcButton)</td>
<td>totalOrderedLabel</td>
<td>None</td>
</tr>
<tr>
<td>Display total price (from calcButton)</td>
<td>totalPriceLabel</td>
<td>None</td>
</tr>
<tr>
<td>Get and display the order information</td>
<td>nameTextBox,</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>addressTextBox,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cityTextBox,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stateTextBox,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zipTextBox,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vanillaTextBox,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>strawberryTextBox</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2-6: Completed TOE chart ordered by object
Designing the User Interface

• Control Placement
  • The user interface should be organized so that:
    • Information flows either vertically or horizontally
    • The most important information is always located in the upper-left corner of the interface
  • In a vertical arrangement:
    • Information flows from top to bottom
    • Essential information is located in the first column
    • Secondary information is placed in subsequent columns
Designing the User Interface

• **Control Placement**
  • In a horizontal arrangement:
    • Information flows from left to right
    • Essential information is placed in the first row
    • Secondary information placed in subsequent rows
  • Related controls should be grouped together using:
    • Either white (empty) space
    • Or one of the GroupBox, Panel, or TableLayoutPanel tools located in the Containers section of the toolbox
Designing the User Interface (cont’d.)

Figure 2-7: Vertical arrangement of the interface

Figure 2-8: Horizontal arrangement of the interface
Designing the User Interface (cont’d.)

• **Sentence capitalization**
  • Capitalize only the first letter in the first word and in any words that are customarily capitalized

• **Label control guidelines:**
  • Use sentence capitalization for text contained in label controls
  • Text should meaningfully identify the label and be left-aligned
  • Text should consist of one to three words and appear on one line
  • Labels should be positioned either above or to the left of the control it identifies.
  • Labels should end with a colon (:)
Designing the User Interface (cont’d.)

• **Book title capitalization**
  • Capitalize the first letter in each word, except for articles, conjunctions, and prepositions that do not occur at either the beginning or end of the caption

• **Button guidelines:**
  • Use book title capitalization for buttons
  • Caption (text) should be meaningful
  • Caption should consist of one to three words and appear on one line
  • Size buttons relative to each other: same height
  • If stacked horizontally, same width also
  • Most commonly used button should be first
Designing the User Interface (cont’d.)

• **Graphics, Fonts, and Colors**
  - Interfaces that contain a lot of different colors, fonts, and graphics become tiresome after a while.
  - The human eye is attracted to pictures before text, so use graphics to either emphasize or clarify a portion of the screen.
  - Keep the following three points in mind when deciding whether to include color in an interface:
    - People who have some form of either color blindness or color confusion will have trouble distinguishing colors.
    - Color is very subjective: A color that looks pretty to you may be hideous to someone else.
    - A color may have a different meaning in a different culture.
Designing the User Interface (cont’d.)

• **Graphics, Fonts, and Colors**
  • Best to use black text on a white, off-white, or light gray background
    • Dark text on a light background is the easiest to read
  • Keep never use a dark color for the background or a light color for the text
    • Dark background is hard on the eyes
    • Light-colored text can appear blurry
  • When including color in the interface, limit the number of colors to three, not including white, black, and gray
Assigning Access Keys

- An access key allows the user to select an object using the Alt key in combination with a letter or number.

Figure 2-9: Say Cheese! application’s interface
Assigning Access Keys (cont’d.)

- Assign access keys to each of the controls (in the interface) that can accept user input
  - Exceptions to this rule are the OK and Cancel buttons, which typically do not have access keys in Windows applications

- It is important to assign access keys to controls:
  - Users can work with the application even when the mouse becomes inoperative
  - Users who are fast typists can keep their hands on the keyboard
  - Users who cannot work with a mouse, such as people with disabilities, can use the application

- Assign access keys by including an ampersand (&) in the control’s caption or identifying label
Controlling the Tab Order

• Each control’s TabIndex property contains a number that represents the order in which the control was added to the form.

• The first control added to a form has a TabIndex value of 0; the second control has a TabIndex value of 1; and so on.

• The TabIndex values determine the tab order, which is the order that each control receives the focus when the user either presses the Tab key or employs an access key while an application is running.
Controlling the Tab Order (cont’d)

<table>
<thead>
<tr>
<th>Controls that accept user input, along with their identifying labels</th>
<th>TabIndex value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label1 (Name:)</td>
<td>0</td>
</tr>
<tr>
<td>nameTextBox</td>
<td>1</td>
</tr>
<tr>
<td>Label2 (Address:)</td>
<td>2</td>
</tr>
<tr>
<td>addressTextBox</td>
<td>3</td>
</tr>
<tr>
<td>Label3 (City:)</td>
<td>4</td>
</tr>
<tr>
<td>cityTextBox</td>
<td>5</td>
</tr>
<tr>
<td>Label4 (State:)</td>
<td>6</td>
</tr>
<tr>
<td>stateTextBox</td>
<td>7</td>
</tr>
<tr>
<td>Label5 (ZIP:)</td>
<td>8</td>
</tr>
<tr>
<td>zipTextBox</td>
<td>9</td>
</tr>
<tr>
<td>Label6 (Vanilla bean:)</td>
<td>10</td>
</tr>
<tr>
<td>vanillaTextBox</td>
<td>11</td>
</tr>
<tr>
<td>Label7 (Strawberry:)</td>
<td>12</td>
</tr>
<tr>
<td>strawberryTextBox</td>
<td>13</td>
</tr>
<tr>
<td>calcButton</td>
<td>14</td>
</tr>
<tr>
<td>printButton</td>
<td>15</td>
</tr>
<tr>
<td>clearButton</td>
<td>16</td>
</tr>
<tr>
<td>exitButton</td>
<td>17</td>
</tr>
</tbody>
</table>

Other controls

<table>
<thead>
<tr>
<th>Controls</th>
<th>TabIndex value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label10 (Order Form)</td>
<td>18</td>
</tr>
<tr>
<td>Label8 (Total ordered:)</td>
<td>19</td>
</tr>
<tr>
<td>totalOrderedLabel</td>
<td>20</td>
</tr>
<tr>
<td>Label9 (Total price:)</td>
<td>21</td>
</tr>
<tr>
<td>totalPriceLabel</td>
<td>22</td>
</tr>
<tr>
<td>PictureBox1</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Figure 2-10: List of controls and Tabindex values
Controlling the Tab Order (cont’d)

**HOW TO** Set the TabIndex Property Using the Tab Order Option

1. If necessary, make the designer window the active window.

2. Click View on the menu bar and then click Tab Order. The current TabIndex values appear in blue boxes on the form.

3. Click the first control you want in the tab order. The color of the box changes to white, and the number 0 appears in the box.

4. Click the second control you want in the tab order, and so on. If you make a mistake when specifying the tab order, press the Esc key to remove the boxes from the form, and then start over again.

5. When you have finished setting all of the TabIndex values, the color of the boxes will automatically change from white to blue. See Figure 2-12.

6. Press the Esc key to remove the blue boxes from the form. Or, click View on the menu bar and then click Tab Order.

Figure 2-11: How to set the Tabindex property using the Tab Order option
Controlling the Tab Order (cont'd.)

Figure 2-12: Correct Taxindex values
Designating the Default Button on a Form

- Specify the **default button** by setting the form’s AcceptButton property to the name of the button.
- A form can have only one default button, but it does not have to have a default button.

Figure 2-13: Default Button on the MainForm
Printing an Interface from Code

- Use Visual Basic’s PrintForm tool (contained in the Visual Basic PowerPacks section of the toolbox)

![PrintForm Example application’s interface](Figure 2-14)
Printing an Interface from Code (cont'd.)

Figure 2-15: Print Preview and Print buttons’ Click event procedures

```vbnet
Private Sub previewButton_Click(sender As Object, e As EventArgs)
    PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
    PrintForm1.Print()   \ starts the print operation
End Sub

guard

Private Sub printButton_Click(sender As Object, e As EventArgs) Han
    PrintForm1.PrintAction = Printing.PrintAction.PrintToPrinter
    PrintForm1.Print()   \ starts the print operation
End Sub
```

© 2016 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Playing Audio Files

• Some applications contain an audio component, such as music, sound effects, or spoken text
• The keyword My in the syntax refers to Visual Basic’s **My feature**—a feature that exposes a set of commonly-used objects to the programmer
• Following the Play method in the syntax is a set of parentheses containing the text fileName
  • Items within parentheses after a method’s name are called **arguments** and represent information that the method needs to perform its task
Playing Audio Files (cont’d)

**HOW TO** Play an Audio File

**Syntax**

```
```

**Example**

```
My.Computer.Audio.Play("GoodMorning.wav")
```

Figure 2-16: How to play an audio file
Programming Tutorial 1

Figure 2-18: MainForm for the Music Sampler application

Everything Begins  
Dan-O

Invisible Love  
Dan-O

Magic Ghost  
Dan-O

Three Drops  
Dan-O

Unclaimed Territory  
Dan-O

Print Preview  Exit
Programming Tutorial 2

Figure 2-27: MainForm for the Alligator Inc. application
Programming Example

Figure 2-31: User interface
Summary

• Plan an application jointly with the user to ensure that the application meets the user’s needs
• Planning an application requires that you identify the application’s tasks, objects, and events
• Build the interface, record the tasks, objects, and events in a TOE chart
• Not all objects will need an event to occur for them to perform their assigned task
• Use a text box control to give the user an area in which to enter data
Summary (cont'd.)

• Organize the user interface so that the information flows either vertically or horizontally, with the most important information always located in the upper-left corner of the screen.

• Group related controls together using either white (empty) space or one of the tools located in the Containers section of the toolbox.

• Text contained in identifying labels should be left-aligned within the label.
  • Identifying labels should be positioned either above or to the left of the control they identify.
Summary (cont'd.)

• Identifying labels and button captions should:
  • Contain only one to three words
  • Appear on one line
  • Have meaningful captions
  • Label captions end with a colon and be entered using sentence capitalization
  • Button captions should be entered using book title capitalization

• When positioning the controls, maintain a consistent margin from the edges of the form
Summary (cont'd)

- Related controls are typically placed close together in the interface
- Horizontally positioned buttons should be the same height; their widths, may vary
- Vertically positioned buttons should be the same width
- Align the borders of the controls if possible to minimize the number of different margins used in the interface
Summary (cont'd)

• Graphics and color should be used sparingly in an interface

• Avoid using italic and underlining in an interface, and limit the use of bold text to titles, headings, and key items that you want to emphasize

• Use only one font type and not more than two different font sizes for the text in an interface
  • Segoe UI (9pt) is the recommended font for Windows 8 and Windows 7 applications
Summary (cont’d)

• Assign access keys to each of the controls that can accept user input (such as text boxes and buttons)
  • assign an access key by including an ampersand (&) in the control’s caption or identifying label

• The TabIndex property determines the order in which a control receives the focus when the user either presses the Tab key or employs an access key during run time

• You use a form’s AcceptButton property to designate a default button
Summary (cont'd.)

• The Visual Basic PowerPacks section of the toolbox provides the PrintForm tool for instantiating a print form control, which you can use to print the interface during run time

• The instantiated control appears in the component tray in the IDE

• You can use the Play method of the My.Computer.Audio object to play a WAV file during run time