

Chapter 2

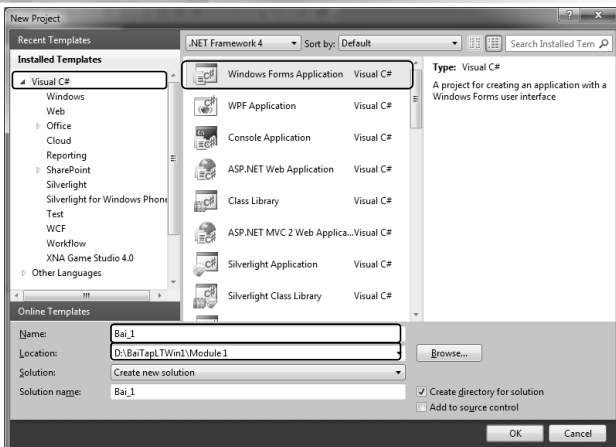
Windows Programming

Contents

- Introduction to Windows Form Application
- Introduction to Form
- Introduction to Control
- Events
- Some common Controls
- Some advanced Controls

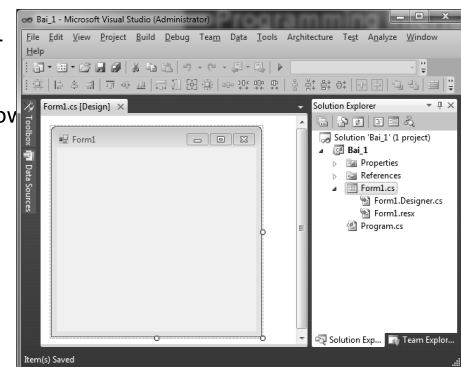
Slide 2

Create a Windows Form Application



Programming interface

- Review
 - Solution Explorer
 - Toolbox
 - Properties Window



Slide 4

Contents

- Introduction to Windows Form Application
- Introduction to Form
- Introduction to Control
- Events
- Some common Controls
- Some advanced Controls

Slide 5

Introduction to Form

- Form (also called Windows Form)
 - is a container for controls and components
 - represented by:
 - Window
 - MDI window
 - Dialog
- Form belongs to System.Windows.Forms namespace

Slide 6

Introduction to Form (cont.)

- Some actions with Form
 - To add a new Form to existing project: right-click to the project shown in Solution Explorer, select Add\Windows Form
 - To add a existing Form to existing project: right-click to the project shown in Solution Explorer, select Add\Existing Item (choose file .cs)
 - To exclude a Form: right-click to the form shown in Solution Explorer, select Exclude From Project
- In the Windows Forms Designer, a Windows form can be seen in two views: the Design View and the Code View
- Exploring the Generated Code

Slide 7

Common Form properties and methods

Form	Description
Common Properties	
AcceptButton	Which button will be clicked when <i>Enter</i> is pressed.
AutoScroll	Whether scrollbars appear when needed (if data fills more than one screen).
CancelButton	Button that is clicked when the <i>Escape</i> key is pressed.
FormBorderStyle	Border of the form (e.g., none , single , 3D , sizable).
Font	Font of text displayed on the form, as well as the default font of controls added to the form.
Text	Text in the form's title bar.
Common Methods	
Close	Closes form and releases all resources. A closed form cannot be reopened.
Hide	Hides form (does not release resources).
Show	Displays a hidden form.

Slide 8

Contents

- Introduction to Windows Form Application
- Introduction to Form
- Introduction to Control
- Events
- Some common Controls
- Some advanced Controls

Slide 9

Introduction to Control

- Control
 - is a component with graphical part, such as button, label...
 - is visible
- Controls belong to System.Windows.Forms namespace
- Most controls derive from the System.Windows.Forms.Control class → many properties and events in the controls are identical

Slide 10

Common properties of Controls

PROPERTY	DESCRIPTION
Anchor	Specifies how the control behaves when its container is resized
BackColor	The background color of a control
Dock	Docks a control to the edges of its container
Enabled	Specifies whether the control receive input from the user
ForeColor	The foreground color of the control
Name	The name of the control
TabIndex	The number the control has in the tab order of its container
TabStop	If true, user can use the Tab key to select the control
Text	Holds the text that is associated with this control
Visible	Specifies whether the control is visible at runtime
METHOD	DESCRIPTION
Hide	Hides the control
Focus	Transfers the focus to a control
Show	Shows the control

11

Anchor property

Design

Darkened bar indicates to which wall control is anchored

Click down-arrow in Anchor property to display anchoring window

Result

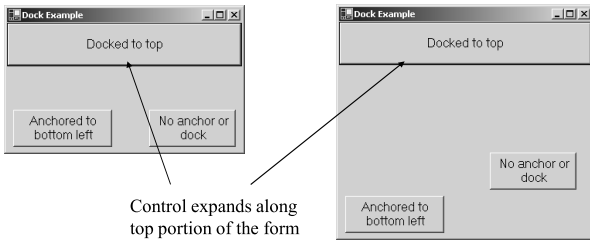
Constant distance to left and top sides

Before resize

After resize

Slide 12

Dock property



Slide 13

Contents

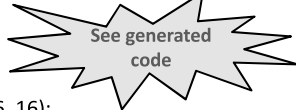
- Introduction to Windows Form Application
- Introduction to Form
- Introduction to Control
- Events
- Some common Controls
- Some advanced Controls

Slide 14

Add a control to a Form

- By using the Windows Forms Designer
 - Select a control and draw it on the container surface
 - Drag a control onto the form at the desired location
 - Add a control to a form by double clicking it
 - Copy/Paste a control to a form
- By programmatically
 - Create a control object and add it to the container
 - Example:


```
Label lblName = new Label();
lblName.Text = "Hello";
lblName.Location = new Point(16, 16);
this.Controls.Add(lblName);
```



Slide 15

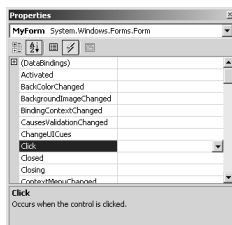
Events

- **Event**
 - When you perform an action with an *object*, the object in turn raises *events* in the application
 - Some common events: clicking a button, typing in a textbox, selecting an item from a menu, closing a window, moving the mouse,...
- **Event handler**
 - Event handler is a *method*, that is executed as a response to an event
- See table 15-2, p.452 for common control event

Slide 16

Handle an Event

- Three basic ways to handle an event
 - Double-click a control, which takes you to the event handler for the control's *default event*
 - Use the Events list in the Properties window
 - Double-click that event in the Events list
 - Type a name for the method to handle the event next to that event in the Events list, and press the Enter key
 - Write the code to subscribe to the event yourself



Slide 17

Event handler

- Should have the name as corresponding delegate:

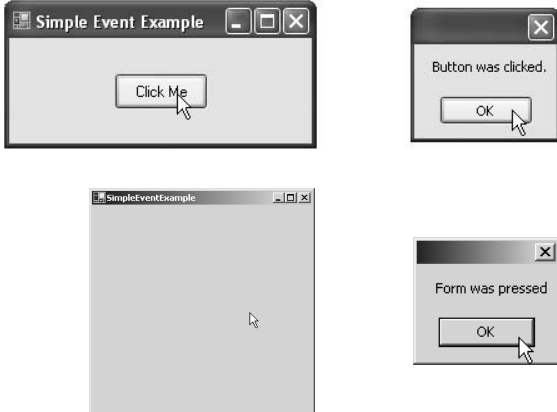
ControlName_EventName
- Must have two object reference are passed in: object, EventArgs
 - Example:


```
private void btnTinh_Click (object sender, EventArgs e)
```
- Must be registered with delegate object
 - Add event handlers to the delegate's invocation list
 - Example:


```
btnTinh.Click += new System.EventHandler(btnTinh_Click);
```

Slide 18

Example



Slide 19

Contents

- Introduction to Windows Form Application
- Introduction to Form
- Introduction to Control
- Events
- Some common Controls
- Some advanced Controls

Slide 20

Naming Rules

- Each control has an identifier (Name property)
 - Can contain letters, digits, and underscores (_)
 - Cannot start with digits
 - Can start with the at symbol (@)
 - No spaces or punctuation marks
- If multiple words capitalize 1st letter of each word
- 3 letter lowercase prefix identifies control type
 - Button - btn
 - Label - lbl
 - Form - frm

Slide 21

Recommended Naming

Object Class	Prefix	Example
Form	frm	frmDataEntry
Button	btn	btnExit
TextBox	txt	txtPaymentAmount
Label	lbl	lblTotal
RadioButton	rad	radBold
CheckBox	chk	chkPrintSummary
PictureBox	pic	picLandscape
ComboBox	cbo	cboBookList
ListBox	lst	lstIngredients
GroupBox	grb	grbColor

Slide 22

Some common Controls

1. Button
2. Label, LinkLabel
3. TextBox, RichTextBox
4. GroupBox, Panel
5. CheckBox, RadioButton
6. PictureBox
7. ListBox, CheckedListBox
8. ComboBox

Slide 23

Button

- Button (p.453): allows the user to click it to perform an action
 - Some properties:
 - FlatStyle
 - Image
 - ImageAlign
 - Text
 - TextAlign
 - Default event:
 - Click event: happens whenever a user clicks the button
- Try it out – Working with Button (p.454)

Slide 24

Label, LinkLabel

- Label (p.456): to display text or images that cannot be edited by the user
 - Some properties:
 - AutoSize
 - BorderStyle
 - FlatStyle
 - Font
 - Image
 - ImageAlign
 - Text
 - TextAlign

Slide 25

Label, LinkLabel (cont.)

- LinkLabel (p.456): to add Web-style links to applications
 - Some properties:
 - ActiveLinkColor
 - LinkArea
 - LinkColor
 - LinkVisited
 - VisitedLinkColor

Slide 26

TextBox, RichTextBox

- TextBox (p.457): to enter text or display information
 - Some properties:
 - CausesValidation
 - CharacterCasing
 - MaxLength
 - MultiLine
 - PasswordChar
 - ReadOnly
 - ScrollBars
 - ScrollBars
 - WordWrap

Slide 27

TextBox, RichTextBox (cont.)

- Some event (TextBox):
 - TextChanged
 - Enter
 - Leave
 - Validating
 - Validated
 - KeyDown
 - KeyPress
 - KeyUp

These events occur in this order

KeyDown and KeyUp receive the keycode corresponding to the key that was pressed.
KeyPress receives the character corresponding to a keyboard key.
- Try it out - Working with a TextBox Control (p.459)

Slide 28