

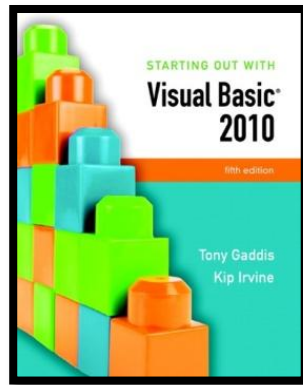


STARTING OUT WITH

Visual Basic® 2010

fifth edition

Tony Gaddis
Kip Irvine



Chapter 11

Developing Web Applications

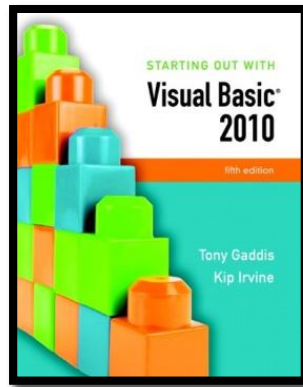
Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

Introduction

- In this chapter you will learn about:
 - Programming for the World Wide Web
 - Creating ASP.NET applications
 - Web server controls and web forms
 - Using databases in ASP.NET



Section 11.1

PROGRAMMING FOR THE WEB

A Web application runs on a Web server and presents its content to the user across a network, in a Web browser.

Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

HyperText Markup Language

- HTML stands for **HyperText Markup Language**
 - Describes appearance of web pages
 - A standardized formatting language
 - It is *not* a programming language
- Formatting instructions appear as commands called tags embedded in the web page text
 - Text following the bold tag (****) is shown in bold until an end bold tag (****) appears
 - This text is bold.This text is normal.**
- Web design editors create HTML for you

ASP.NET

- The acronym ASP originally stood for **Active Server Pages**
 - **ASP.NET**, the next generation, is a server-side Web programming **platform**
 - Provides development tools and visual controls for web browser based applications
 - Contains Web forms and controls, HTML, and program logic in compiled VB code
 - VB knowledge transfers directly to ASP.NET
- VB code runs on the server, not the client
 - Server runs code that creates an HTML page
 - Client web browser receives the HTML

Web Clients and Web Servers

- The *client-server* model
 - A server is a computer that produces data
 - A client is a computer that uses the data
- Web applications use the **client-server model**
 - Web browsers run on clients and request data from web servers
 - Web sites are hosted on Web servers that produce data as requested by Web browsers

Uniform Resource Locator (URL)

- A **URL (Uniform Resource Locator)** references a particular web page
- For example:
 - http://pearsonhighered.com**
 - Begins with the **protocol**
 - **http://**
 - Then the **domain name**
 - **pearsonhighered.com**
 - May end with a specific folder path and/or filename
- The URL is used as an address that uniquely identifies the web page to be retrieved

Displaying a Web Page

- What happens when a Web page is displayed by a Web browser?
 - A computer must be running a **Web server**, which waits for browser connection requests, this occurs in two steps:
 1. Browser connects to server requesting a URL
 2. Server translates URL into a physical file located within the server's file system and sends the requested file, called a **Web page**, back to the browser
 - Server breaks connection after sending Web page
 - Web Browser interprets HTML and renders a Web page
 - **PostBack** occurs if client requests Web page again
 - By clicking a **button control** or pressing the Enter key

Web Forms

- ASP.NET web pages are called Web forms
 - A **web form**
 - Uses a file name extension of **.aspx**
 - Contains text, **HTML tags**, and **HTML controls**
 - Also contains **Web server controls** such as text boxes, list boxes, and buttons
 - Also known as **ASP.NET Server controls**
 - Similar to Windows Forms controls
 - The **Program logic** for a Web form
 - is stored in a related **code-behind file** with extension **aspx.vb**
 - A **Cascading style sheet (CSS)** file customizes the appearance of a Web form

Web Servers

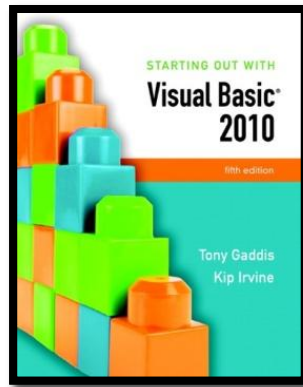
- Web applications must be run using a Web server
 - Three choices are available:
 1. **The ASP.NET Development Server**
 - Installed automatically with Visual Studio
 2. **Internet Information Services (ISS)**
 - An option with certain versions of Microsoft Windows
 - Requires careful security configuration
 3. A remote Web server
 - Typically available through an Internet Service Provider (ISP) or a corporate Web server
 - Must always have an account with a username and password

HTML Designer and Web Browser Support

- **HTML Designer** is a tool in Visual Studio that simplifies the design of Web pages and Web forms
 - Generates HTML source code and ASP.NET Web controls
 - Offers the following views of a Web page:
 - A **Design view** similar to Visual Studio's forms editor
 - A **Source view** for direct editing of HTML source code
 - A **Split view** that displays both views in separate panels
- There are many different Web browser versions in use
 - ASP.NET detects and generates version specific HTML
 - Always test Web applications on other browsers
 - Chrome, Safari , and Firefox are good choices

Types of Controls

Group	Description
Standard	Commonly used controls on Web forms
Data	Controls for connecting to and displaying data
Validation	Controls for validating user input
Navigation	Advanced controls for navigating between Web pages
Login	Controls related to authentication of usernames and passwords
WebParts	Controls modify content, appearance, and behavior of Web pages
AJAX Extensions	Controls that provide rich interface experiences
Reporting	Contains Microsoft Report Viewer for displaying Web-based reports
HTML	Standard HTML controls that do not generate user events



Section 11.2

CREATING ASP.NET APPLICATIONS

You can use Visual Studio or Visual Web Developer Express Edition to create Web applications in Visual Basic.

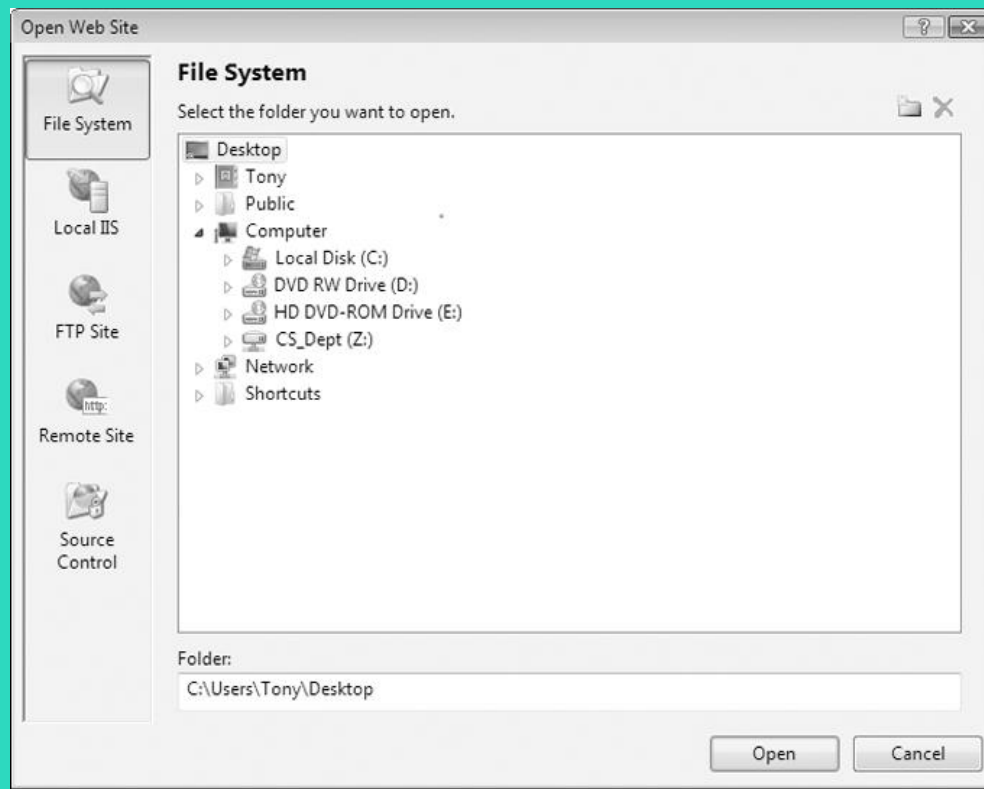
Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

Open Web Site Dialog Box

- Select *Open Web Site* from the *File* menu when you want to open an existing Web application

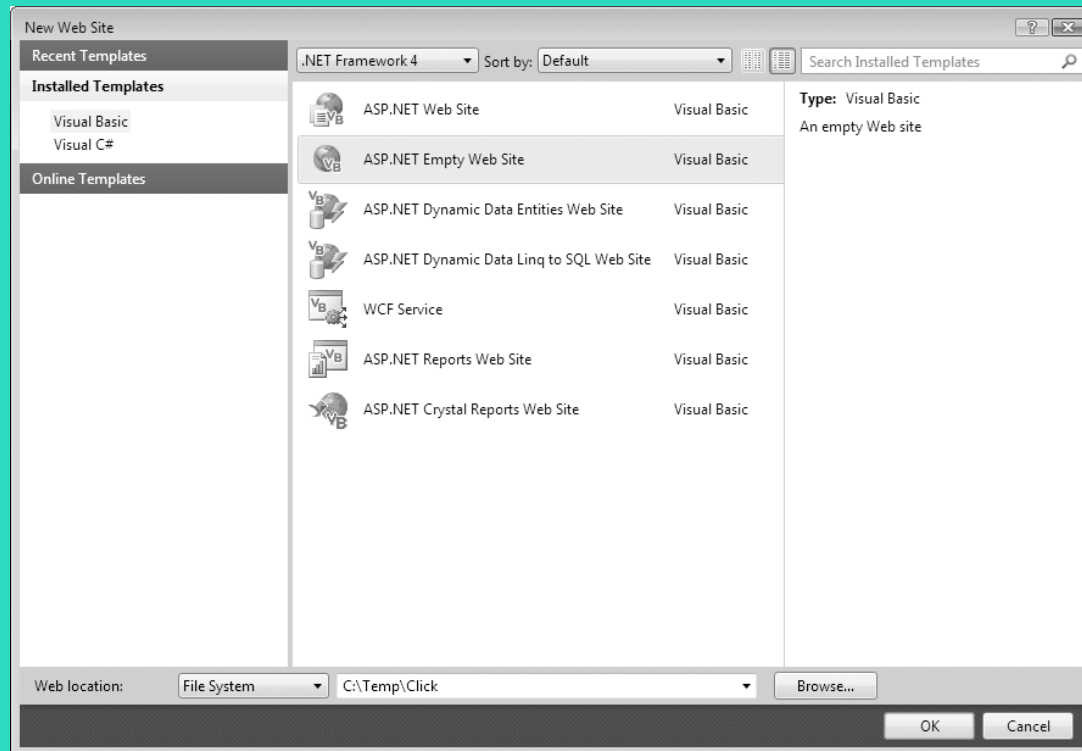


Types of Web Sites

- **A File System Web Site** is best suited to a network
 - Uses ASP.NET Development Server
 - Supplied with Visual Studio
 - Simple to use, not open to security attacks
- **Local ISS** for a local Web server
 - Uses Internet Information Services, or IIS
 - Professional level, extensive security features
 - Extensive set-up, must have admin rights
- **FTP Site** is located on a different machine
 - Stands for *File Transfer Protocol*
 - Usually on the Internet
 - Provides a way of copying files from one machine to another
- **Remote Site** if existing site on remote server
 - Need userID & password to upload application

Creating a Web Application

- Click *New Web Site* from *File* menu
 - The *New Web Site* dialog box appears



Creating a Web Application

- The *New Web Site* dialog box lists possible Web sites templates
 - Select *ASP.NET Empty Web Site* from the list
- Choose a folder for the project
 - If File System, can choose to use any folder on local computer or network
 - If HTTP, application will be located on a Web site set up by IIS
 - If FTP, must use Web site on remote computer
- When created, an empty Web site contains only one file named **Web.config**

Opening an Existing Web Application

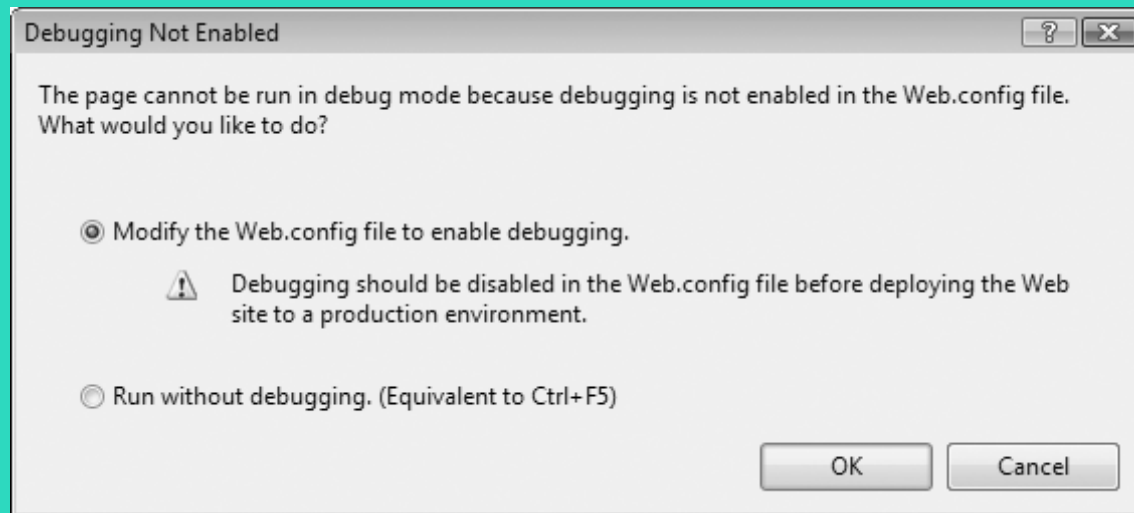
- To open an existing Web application
 - Select project from *Recent Projects* window
- If project doesn't appear in *Recent Projects* click
 - *Open: Web Site...* link in *Recent Projects*
 - Or click *Open Web Site* on *File* menu
- Either of the two previous options display an *Open Web Site* dialog box
 - Navigate to folder containing Web site
 - Click *Open*

Running a Web Application Project

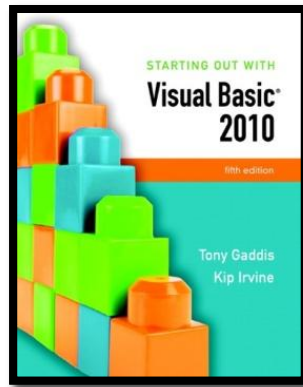
- Can change default browser for your project
 - Right-click project name in Solution Explorer
 - Select *Browse With...* from shortcut menu
- To run your Web application
 - Click *Run Without Debugging* on *Debug* menu
- Web forms allow constants called **static text**
 - No label control required like a Windows form

Running a Web Application Project

- Must configure a project for debug capability
 - Message box shown when first running a project in debug mode
 - Clicking *OK* adds option to **Web.config** file that enables debugging



- Tutorial 11-1, you create the *Click* application



Section 11.3

WEB SERVER CONTROLS

Web Server controls are similar to controls used in Windows applications. You use Web Server controls to make ASP.NET Web applications interactive.

Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

Web Server Controls Overview

- Make ASP.NET dynamic and interactive
- Work like HTML controls but far more flexible
 - Class based with properties, methods, events
 - Similar to Windows form controls, making it easy for VB programmers to learn
- Frequently used Web controls:

Button

Label

TextBox

CheckBox

DropDownList

ImageButton

RadioButton

CheckBoxList*

Image

LinkButton

RadioButtonList*

ListBox

Calendar

* Those noted with asterisk have no Windows form equivalent

Web Server Controls Overview

- Web controls properties similar to those of Windows form controls including
 - Text, Enabled, Visible, Font, ReadOnly, and so on.
- There are some important differences
 - Windows control Name property same as the ID property for Web controls
 - Web controls have an AutoPostBack property
 - Web controls lose runtime properties when the user moves away from that page
 - Must *save state* to retain runtime properties

How Web Controls Are Processed

- ASP.NET functions differently from HTML
- The Web server executes the VB code found behind the ASP.NET Web page
- When a browser requests an **.aspx** Web page
 - Server reads/interprets Web controls on page
 - VB statements in code-behind file executed
 - Web page of standard HTML tags and controls built using **.aspx** Web controls and VB code
 - HTML Web page sent back to browser

Label and Text Controls

- Label control displays data from program
 - Use only if label text will change at runtime
 - If text does not change, set up as *static text*
- TextBox control holds text input by user
 - TextMode property can be:
 - SingleLine: permits a single line of input
 - MultiLine: permits multiple lines of input
 - Password: characters typed appear as asterisks
 - Deal with browser compatibility issues using:
 - Columns property to control TextBox width
 - Rows property to specify entry of multiple lines

CheckBox Control

- Functions almost identically to CheckBox in Windows forms
 - Text property sets text visible to user
 - Evaluate Checked property at runtime to determine if control checked by user
 - TextAlign lets you position text
- Tutorial 11-2 creates a Web sign-up for a *Student Picnic* application

Handling Events in Web Forms

- Events fire differently in Web forms
 - Page_Load event fires each time a page is displayed instead of just the first time
 - Page_Load fires before other events such as *TextChanged*
- Mouse click on a control with *AutoPostBack* property set to true sends form back to server
 - Useful if server should react to a mouse click such as selecting an item from a list box
 - Occurs automatically for Button, LinkButton, and ImageButton controls

HyperLink Control

- Provides a link to navigate to another page
 - *Text* property specifies text shown for link
 - *NavigateURL* property holds destination URL
 - *Target* property determines if a new browser window is opened to display the new page
 - Set equal to *_blank* to open a separate window

ImageButton, LinkButton, and RadioButtonList

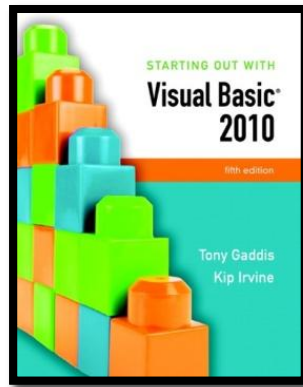
- *ImageButton* provides a clickable image
 - Generates a click event
 - *ImageUrl* property specifies path to image
- *LinkButton* behaves like a hyperlink but generates a click event
- *RadioButtonList* is a group of radio buttons
 - Functions similar to a *ListBox*
 - Has *SelectedIndex* & *SelectedValue* properties

ListBox Control

- Very similar to the Windows ListBox control
 - Has an Items collection
 - Has the ListBox properties *SelectedIndex*, *SelectedItem*, and *SelectedValue*
 - *SelectionMode* property specifies whether multiple list items may be selected
- *SelectedIndexChanged* event handling
 - Must set *AutoPostBack* to true if this event should fire immediately upon a user selection
 - If not, event fires only after another control causes form to be posted back to the server

CheckBoxList and DropDownList

- *CheckBoxList* control looks like group of check boxes but works like a *ListBox*
 - Has an *Items* collection
 - Has the *ListBox* properties *SelectedIndex*, *SelectedItem*, and *SelectedValue*
 - Each item has a Boolean *Selected* property
- *DropDownList* similar to *ComboBox* except:
 - Initial value of *SelectedIndex* always zero so the first item is always displayed
 - Must select item from list, cannot key entry



Section 11.4

DESIGNING WEB FORMS

HTML tables can be used to design a Web application's user interface. HTML tables provide a convenient way to align the elements of a Web form.

Addison Wesley
is an imprint of

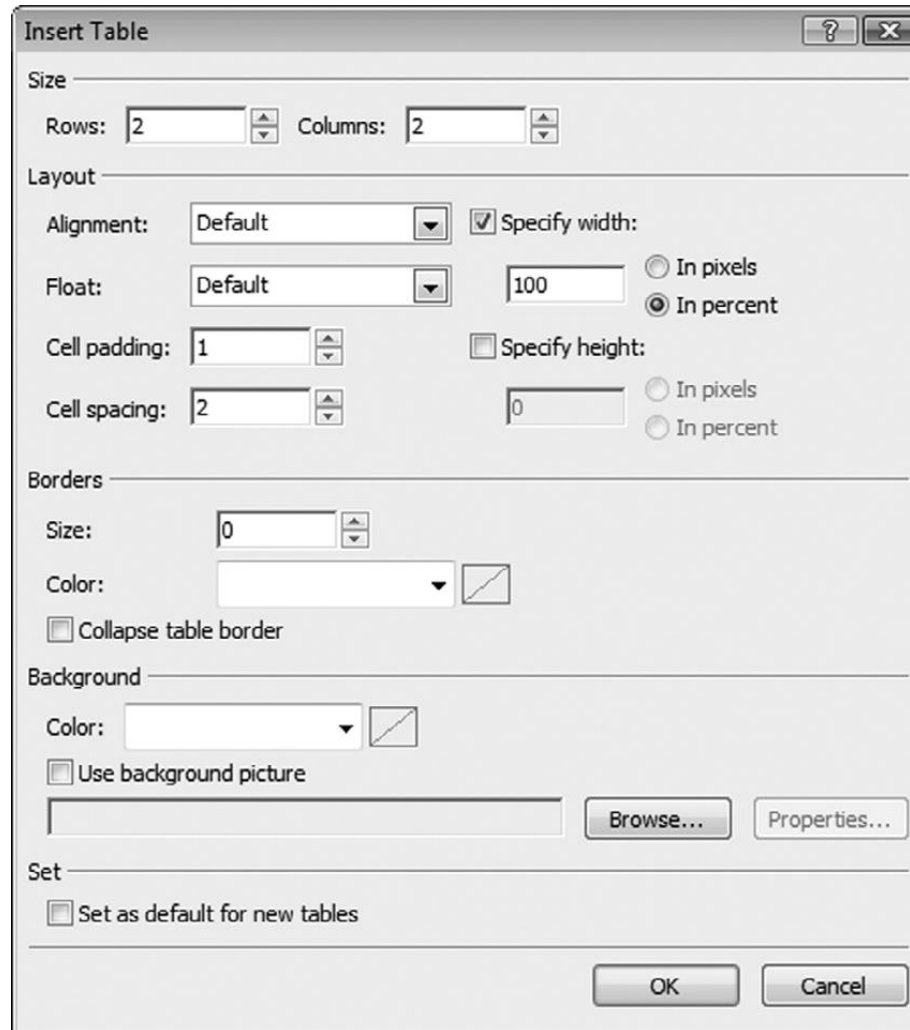


© 2011 Pearson Addison-Wesley. All rights reserved.

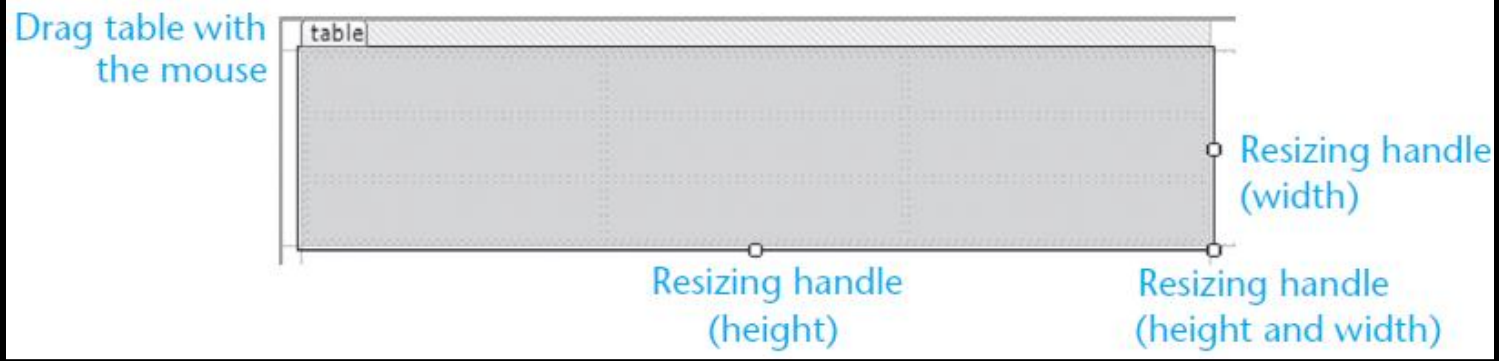
Using Tables to Align Text and Controls

- Essential tool in Web form design
- Creates a grid of rows and columns
- Text and controls placed inside cells of the grid
 - Permits text and controls to be aligned
 - Align by right or left justifying each column
- Blank columns may be used for spacing
- Click *Insert Table* in *Table menu* to show *Insert Table* dialog box

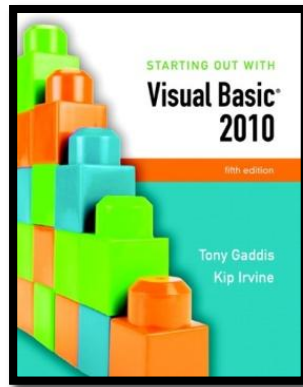
Insert Table Dialog Box Example



Adjusting Row Heights and Column Widths



- Click and drag to adjust row height or column width
- Insert rows or columns with *Insert* on *Table* menu
- Can set cell *Align* property to center, left, or right
- Adjacent cells can be merged together
 - Drag mouse over cells to be merged
 - Select *Merge Cells* from *Layout* menu
- Tutorial 11-3 aligns controls with HTML table



Section 11.5

APPLICATIONS WITH MULTIPLE WEB PAGES

A Web application may use multiple Web pages to display data and interact with the user.

Addison Wesley
is an imprint of

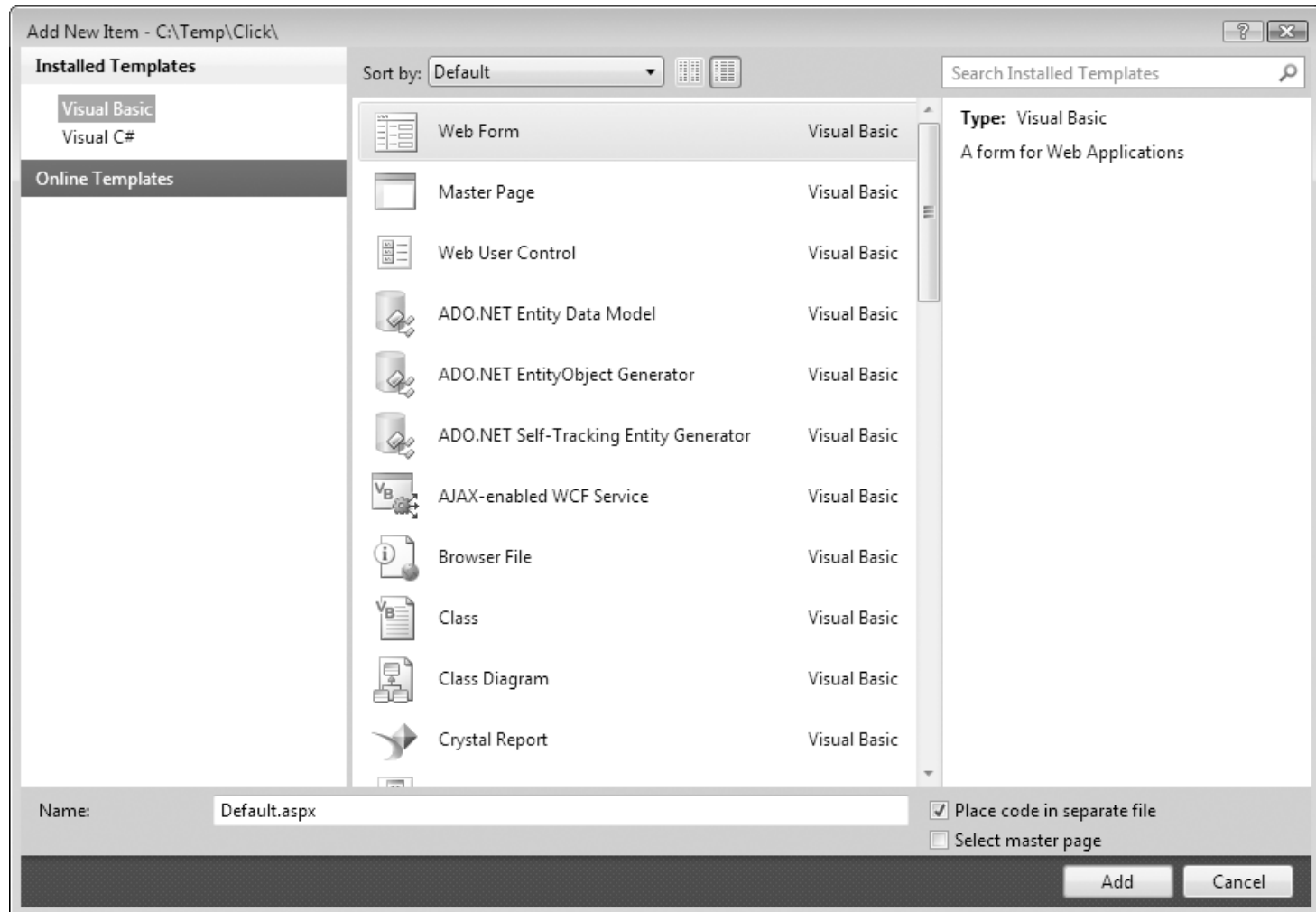


© 2011 Pearson Addison-Wesley. All rights reserved.

Adding New Web Forms to a Project

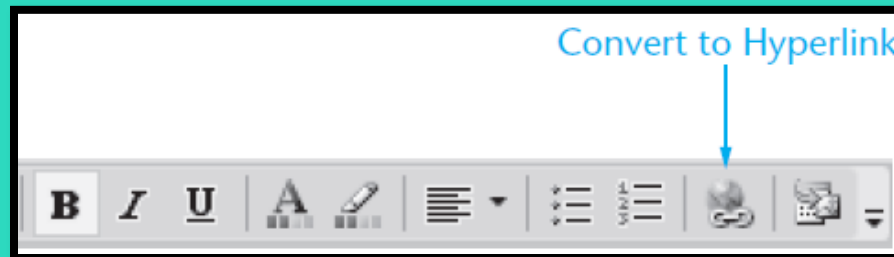
- Two ways to add a new Web form to a project
 - Select *Add New Item* from *Web Site* menu
 - Right-click project in Solution Explorer and select *Add New Item*
- Either displays the *Add New Item* window
 - Select Web Form icon
 - Enter name of page
 - Be sure *Place code in separate file* checked

Adding a Web Form to a Project



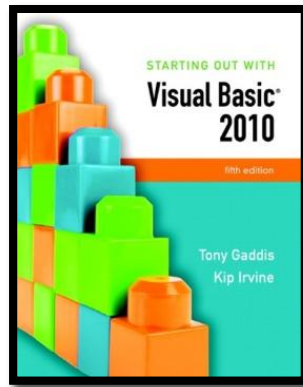
Moving Between Pages

- To allow the user to move between pages:
 - Specify URL of target page in *NavigateURL* property of a *HyperLink* control
 - Use *Response.Redirect* method in click event of a Button, ImageButton, or LinkButton
 - Use *HyperLink* button on *Formatting* toolbar to convert static text into a hyperlink



Calling Response.Redirect

- Allows programmer to display another Web page using code in a click event handler
- Transfer to Web page *Page_two.aspx* using:
`Response.Redirect("Page_two.aspx")`
- Complete URL needed to display a page on another server:
`Response.Redirect("http://microsoft.com")`
- Tutorial 11-4 adds a description form to the *Kayak Tour* application



Section 11.6

USING DATABASES

ASP.NET provides several Web controls for displaying and updating a database from a Web application.

Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

Web Forms Database Access

- Web forms database access differs from that used for Windows forms
- Dataset not used
- *DataSource* control used instead
- Access databases use *AccessDataSource* control
- SQL Server databases use *SqlDataSource* control
- *DataSource* controls update database directly
- No Update method required as with a dataset

GridView Database Connection Setup

- *Data Source* option in *GridView Tasks* menu allows database connection to be configured
 - Copy database file to App_Data folder
 - Select Data Source, use *Database*
 - Select database file from the App_Data folder
 - Configure Select statement for the SQL query
 - If query requires multiple tables, must create custom SQL using Query Builder
 - Places a *DataSource* control on the Web form
- Tutorial 11-5 configures a *GridView* control

Using a DetailView Control to Modify Table Rows

- Found in *Data* section of *Toolbox* window
- *GridView* only displays database tables
- *DetailsView* can be used to view, edit, delete, or add rows to a database table
- Connect to data source just as with *GridView*
- Allows you to create an effective update program without writing any code
- Tutorial 11-6 demonstrates how to use the *DetailsView* control

SQL Queries Inside the SqlDataSource Control

- To see how a SqlDataSource is represented in HTML code
 - Click the Source tab for the *Default.aspx* page and look for the `sqlDataSourceControl`
 - The following code shows the beginning of the code that defines the *MembersDataSource* from Tutorial 11-6:

```
<asp:SqlDataSource ID="MembersDataSource" runat="server"
```
 - A connection string definition:

```
ConnectionString="<%$ ConnectionStrings:karateConnectionString %>"
```
 - The DeleteCommand property:

```
DeleteCommand="DELETE FROM [Members] WHERE [ID] = @ID"
```
 - The InsertCommand property:

```
InsertCommand="INSERT INTO [Members] ([ID], [Last_Name],  
[First_Name], [Phone], [Date_Joined]) VALUES (@ID, @Last_Name,  
@First_Name, @Phone, @Date_Joined)"
```