Reusable Code For ASP.NET : User Control

Chapter 8

What Are You Going To Learn?

- Creating User Control
- Loading User Controls Programmatically
- Custom Control

What is a User Control?

- It is an ASP.NET page that has been converted into a control
- file extension = .ascx

User Control

- A user control can contain both HTML and Web Controls.
- You can place multiple Web Form control in a user control and expose them as a single control.
- A user control can contain the same event-handling subroutines as a normal ASP.NET page.

Advantages (Pros)

Reusability

- Enable you to reuse the same content and programming logic on multiple ASP.NET pages.
- Used for repetitive elements (such as headers, menus, login controls and etc.) on page.

Code reduction and encapsulation

- Reducing the amount of code per page by encapsulation repetitive elements.

Improve performance

- enable fragment caching

Limitations (Cons)

- User controls aren't ideal for:
  - Separation of presentation HTML from the code blocks (server control).
  - Encapsulation of data access methods in a reusable package (Data access class/component).
  - Creating a control that can be reused more widely than in just the application (which can be achieved by custom control).

Question

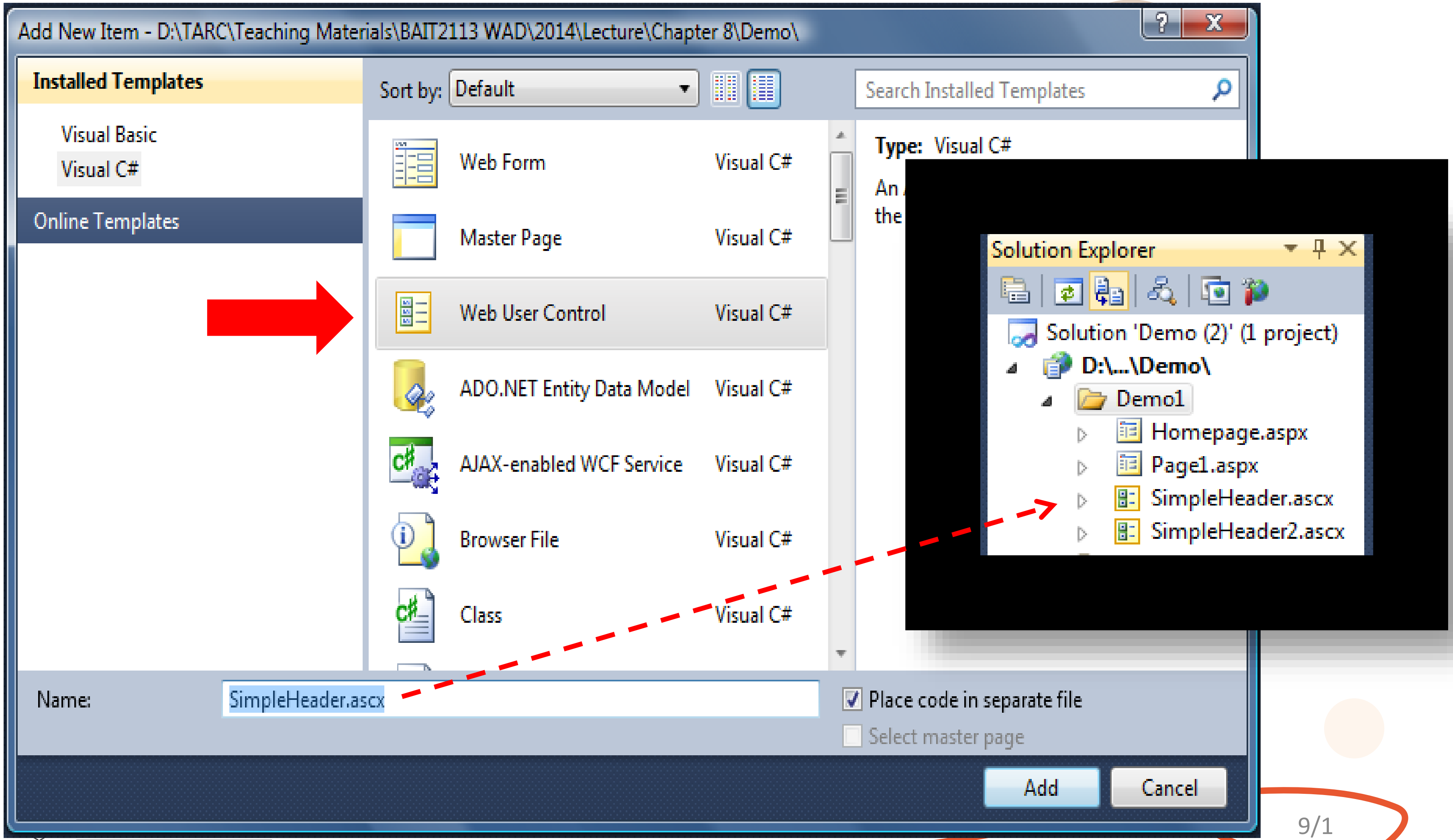
1. Discuss the reasons of using User Controls in a Website over Master Page.

Question

Between user control and Master Page, evaluate which way is better to suit the following requirements for developing a website.

1. You wish to add common functionality in all pages. For example, every page will consist of a common header and footer.
2. There are 2 different groups of users who can login to a website. However, different group of users will see different menu after logging in.

Creating a User Control



Add New Item - D:\TARC\Teaching Materials\BAIT2113 WAD\2014\Lecture\Chapter 8\Demo\

Installed Templates
- Visual Basic
- Visual C#

Online Templates

Sort by: Default

Search Installed Templates

Web Form	Visual C#
Master Page	Visual C#
Web User Control	Visual C#
ADO.NET Entity Data Model	Visual C#
AJAX-enabled WCF Service	Visual C#
Browser File	Visual C#
Class	Visual C#

Type: Visual C#

Name: SimpleHeader.ascx

[✓] Place code in separate file

[ ] Select master page

Solution Explorer
- Solution 'Demo (2)' (1 project)
  - D:\...\Demo\
    - Demo1
      - Homepage.aspx
      - Page1.aspx
      - SimpleHeader.ascx
      - SimpleHeader2.ascx

Adding a User Control on a Page

1. register a user control on a page:

```
<%@ Register TagPrefix="SuperCompany" TagName="Header"
Src="SimpleHeader.ascx" %>
```

2. adding the user control:

```
<SuperCompany:Header ID="ctlHeader" runat="server"/>
```

Each control must have a **unique ID** if it is used for many times in a page.

Question

- Based on information about a Web user control below, write the necessary code on a Web form (aspx file) to display the control.
  - TagPrefix: myMenu
  - TagName: MainMenu
  - Src: Menu.ascx

Question

Provided the code snippets below that display 3 user controls on an aspx page, write the necessary codes to register the respective user controls of “header.ascx”, “menu.ascx” and “login.ascx”.

```
<control:Header ID="ctlHeader" runat="server"/>
<control:Menu ID = “ctlMenu” runat = “server” />
<login:Main ID= “ctlLogin” runat= “server” />
```

TARUMT TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

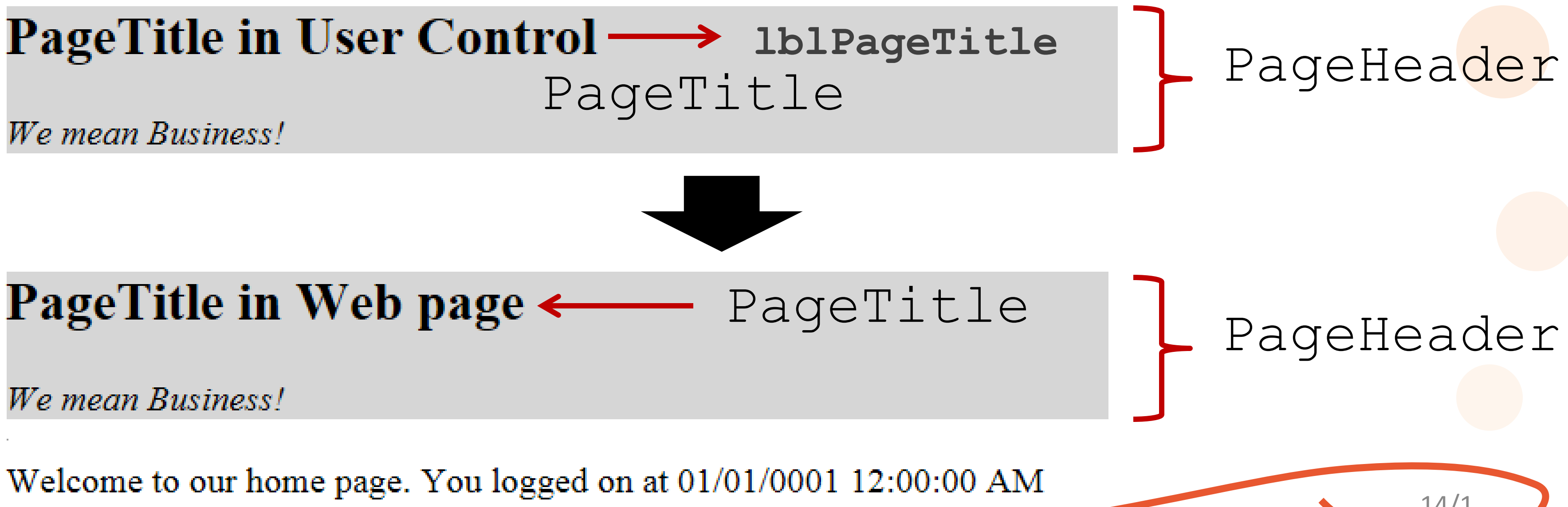
Properties and Methods in User Controls

- User control's content can be static or dynamic
- You can expose properties in a user control
- All <u>public variables</u> declared in the user control file are exposed <u>as properties</u> of the user control
- <u>Function</u> and subroutines contained in the user control are exposed <u>as methods</u> of the user control.

Properties in User Controls

- We can assign a value to a user control property programmatically.

Consider we would like to create a property called `PageTitle` for the `PageHeader` user control:



Welcome to our home page. You logged on at 01/01/0001 12:00:00 AM

Properties in User Controls

- In PageHeader.ascx.cs

Create a property called `PageTitle` for the `PageHeader` user control:

```
public partial class PageHeader : System.Web.UI.UserControl
{
  public string PageTitle = “PageTitle in User Control";

  protected void Page_Load(object sender, EventArgs e)
  {   lblPageTitle.Text = PageTitle;
  }
}
```

$\longrightarrow$ display the PageTitle on the label

The *public* variable will be accessed by all pages that contains the `PageHeader` user control.

Properties in User Controls

- In Homepage.aspx

Set the property called `PageTitle` for the `PageHeader` user control:

```
<%@ Register TagPrefix="Page" TagName="Header" Src="PageHeader.ascx" %>

<Page:Header ID="ctlHeader1" runat="server"
        PageTitle="PageTitle in Web page"/>
```

→ display the PageTitle on the label in PageHeader control

To overwrite the default title, you can set the value of the `PageTitle` property in the user control tag

Question

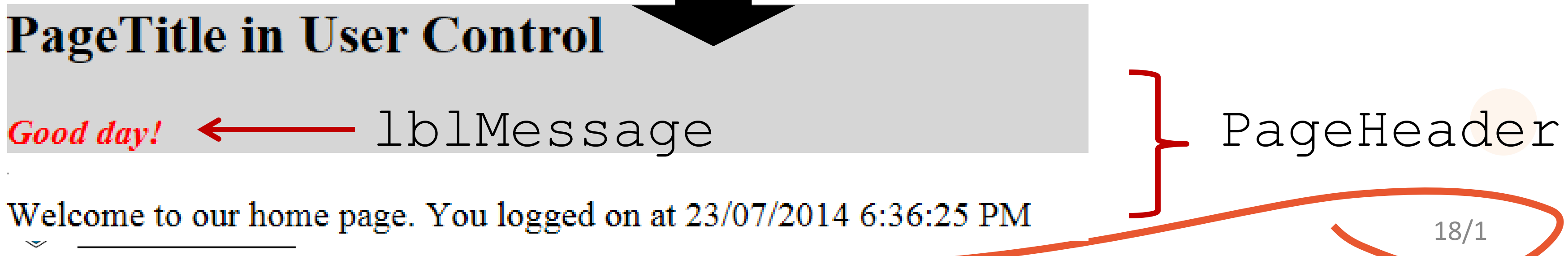
- Assume that the user control “Convert.ascx” consists of a property called MinValue. Demonstrate in code how to include the user control in an ASP.NET Web page, and set the MinValue to 1 declaratively, with appropriate TagPrefix and TagName settings.

Method in User Controls

- We can also expose methods in a user control programmatically.

Consider we would like to create a method called `CheckTime` for the `PageHeader` user control.
Rule:

```
IF      CurrentTime < 12.00pm
THEN    Display "Good morning!"
ELSE    Display "Good day!"
```


PageTitle in User Control

Good day!

Welcome to our home page. You logged on at 23/07/2014 6:36:25 PM

Method in User Controls

- In PageHeader.ascx.cs

Create a function called `checkTime` for the `PageHeader` user control:

```
public partial class PageHeader : System.Web.UI.UserControl
{
   public DateTime loginDate;

   public void checkTime()
   {
      string message = "Good day!";
      if (loginDate.Hour < 12)
         message = "Good morning!";
      lblMessage.Text = message;
   }
}
```

The *public* method will be accessed by all pages that contains the `PageHeader` user control.

display the message on the label in PageHeader control

Method in User Controls

- In Homepage.aspx.cs

Consider we would like to create a property called `PageTitle` for the `PageHeader` user control:

set the loginDate property of PageHeader control

```
protected void Page_Load(object sender, EventArgs e)
  {
      ctlHeader.loginDate = DateTime.Now;
      lblDate.Text = Convert.ToString(ctlHeader.loginDate);
      ctlHeader.checkTime();
  }
```

call the *checkTime* method of PageHeader control

Loading User Controls Programmatically

- User control can be generated dynamically (programmatically) by using LoadControl() method.

```
Consider we would like to load a specific user control based on
user's gender.
Rule:
IF      gender = 'male'
THEN    Load "MaleAd.ascx"
ELSE    Load "FemaleAd.ascx"
```

Loading User Controls Programmatically

- In Homepage.aspx.cs

load a specific user control based on user's gender.

```
protected void Page_Load(object sender, EventArgs e)
{
   if(!IsPostBack)
   {   string gender = "male";
       Control ctlControl;
       if(gender == "male")
           ctlControl = LoadControl("maleAd.ascx");
       else
           ctlControl = LoadControl("femaleAd.ascx");
       plhAd.Controls.Add(ctlControl);
   }
}
```

load control dynamically

adding the control to a PlaceHolder control called *plhAd*

Question

- You plan to display a personalized advertisement banner on the home page (Default.aspx) of a Website based on the gender of the user. If the user is a **male**, the user control named **“watch.ascx”** will be displayed in a PlaceHolder control named “**phBanner”.** If the user is a **female**, then the **“makeup.ascx”** control will be displayed.
- When the user accesses to the home page, the gender will be retrieved from a Cookie named “gender” and will be stored into a session variable named “Sex”. However, **if the cookie does not exist**, then a user control named **“general.ascx”** will be displayed.
- Demonstrate your code in Page_Load event handler to achieve the requirements above.

TARUMT TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Custom Control

Three ways to create a custom control:

derived custom control

- deriving from an existing control.

composite control

- grouping existing controls together into a new compiled control.

full custom control

- deriving from System.Web.UI.WebControls.WebControl.
- Composite controls are most similar to user controls.

User Control vs. Custom Control

User Controls	Custom controls
These are like .aspx pages	These are .DLL files
Extension is .ascx	these are precompiled components
supports Caching	does not support caching
Easier to create	Harder to create
Good for static layout	Good for dynamic layout

User Control vs. Custom Control

User Controls	Custom controls
▪ Limited support for consumers who use a visual design tool	▪ Full visual design tool support for consumers
▪ A separate copy of the control is required in each application	▪ Only a single copy of the control is required, in the global assembly cache
▪ Cannot be added to the Toolbox in Visual Studio	▪ Can be added to the Toolbox in Visual Studio

Question

- Briefly describe the differences between a user control and a custom control.