

Lập trình Web

Thời gian:

- Lý thuyết: 30 tiết
- Thực hành: 30 tiết
- Tự học: 15 tiết

Khoa Công nghệ thông tin

GV: TS. Vũ Đức Lung

lungvd@uit.edu.vn

Điểm đánh giá

- Điểm chuyên cần: 10%
- Điểm làm đồ án môn học: 30%
 - Sinh viên chia thành nhóm 3 người
 - Đăng ký với lớp trưởng, Danh sách gồm: stt nhóm, Họ & tên SV, MSSV, e-mail của trưởng nhóm (buổi sau lớp trưởng nộp lại)
- Điểm thi cuối kỳ: 60%

Mục tiêu môn học

- ❑ Nắm vững các khái niệm căn bản của quá trình thiết cài đặt và triển khai một ứng dụng web trên nền .NET
- ❑ Thiết kế trang Web và cài đặt một ứng dụng Web hoàn chỉnh bằng HTML, CSS, ASP.NET
- ❑ Triển khai và khai thác ứng dụng webservice trên Internet

Nội dung môn học

Chương 1 : Giới thiệu.

Chương 2: Lập trình hướng đối tượng trong ASP.NET

Chương 3: Web Server Control

Chương 4: Các đối tượng của ASP.NET

Chương 5: Xử lý dữ liệu với ADO.NET

Chương 6 : Web Services

Tài liệu tham khảo

1. Zak Ruvalcaba. Build Your Own ASP.NET Website Using C# & VB.NET. 2004, Pub. SitePoint.
2. Việt Văn book. Tự Học Asp.net 2.0 Từng Bước Một, Nxb Thống kê, 2007
3. Đỗ Lâm Thiên. Giáo trình “Lập trình ứng dụng web với ASP.NET”. NXB ĐHQG TP.HCM, 2007.
4. Phạm Hữu Khang. Lập trình ASP.NET 2.0, tập 5. NXB Lao Động 2007.

Website :

www.msdn.microsoft.com/netframework

www.asp.net

<http://dot.net.vn>

Chương 1: Giới thiệu

Nội dung:

- Giới thiệu .NET
- Mục đích .NET
- ASP với ASP.NET
- Cơ bản về ASP.NET

Giới thiệu .NET

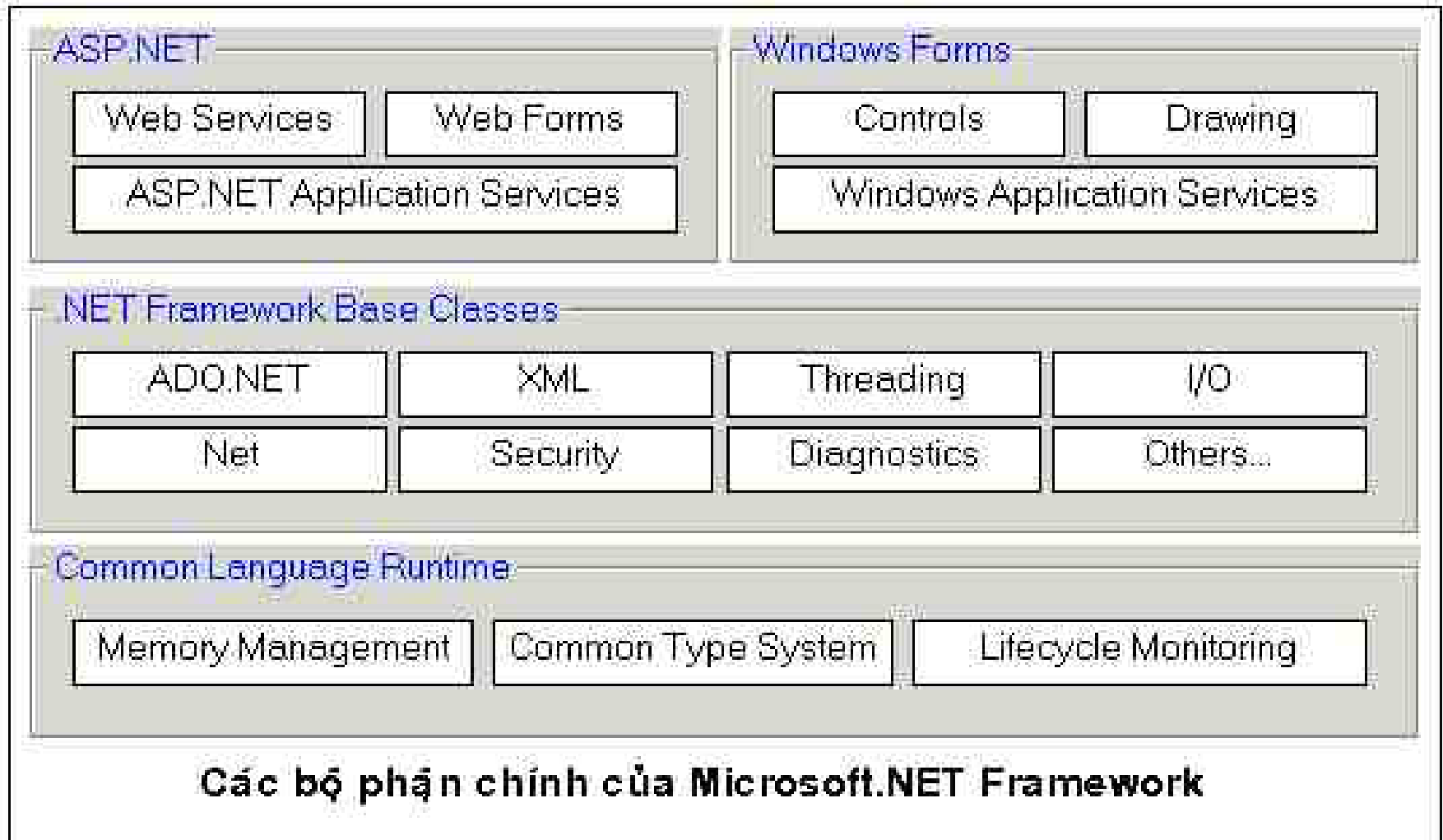
.NET là gì?

A new software platform for the desktop and the Web

Nguồn gốc của .NET

- HĐH = tập hợp COM (Component Object Model) tương tác với nhau
- CTrình = COM
- COM+
- 1998 IIS ver.4 => Project: Khung dịch vụ web thế hệ kế tiếp (Next Generation Web Service Framework - NGWSF)
- 11/2000 phát hành Beta 1 của .NET
- Cuối 2001 chính thức cho ra đời phiên bản đầu tiên

.NET = Framework + IDE



Giới thiệu .NET (tt)

- **.NET** gồm có hai phần
 - Framework 1.0, 1.1, 2.0, 3.0, ...
 - Integrated Development Environment (IDE)
- **Kiến trúc .NET có 3 tầng:**
 - Common Language Runtime (CLR)
 - Bộ thư viện các lớp cơ sở (.NET Framework Base Classes): String, Integer, Exception, ...
 - ASP.NET và Windows Forms



Giới thiệu .NET

.NET Framework

- .NET Framework là nền tảng cho Microsoft.NET Platform
- .NET Framework là môi trường cho việc xây dựng, triển khai và vận hành các ứng dụng Web cũng như Web Service
- .NET Framework giúp đơn giản hóa quá trình phát triển cũng như tích hợp sản phẩm
- □□ .NET Framework không phụ thuộc vào ngôn ngữ. Các ngôn ngữ đang được hỗ trợ: C++, C#, VB.NET, J#
- .NET Framework chứa Common Language Runtime (CLR) và các lớp thư viện cung cấp các dịch vụ cơ sở để xây dựng các ứng dụng

Mục đích .NET

Mô hình đồng nhất cho lập trình desktop
và Web

Truyền thống

Desktop programming

object-oriented
compiled (C/C++, Fortran, ...)
extensive class libraries

Web programming

ASP (not object-oriented)
interpreted (VBScript, Javascript, PHP, ...)
specialized libraries

.NET

Desktop and Web programming

object-oriented (even ASP.NET)
compiled (C#, C++, VB.NET, Fortran, ...)
uniform class library

Mục đích .NET (tt)

Tương tác giữa các ngôn ngữ lập trình

Trước đây:

Rất hạn chế

.NET

- binary compatibility between more than 20 languages (C#, C++, VB.NET, Java, Eiffel, Fortran, Cobol, ML, Haskell, Pascal, Oberon, Perl, Python, ...)

class in VB.NET

```
Public Class A
  Public x As Integer
  Public Sub Foo() ...
End Class
```

subclass in C#

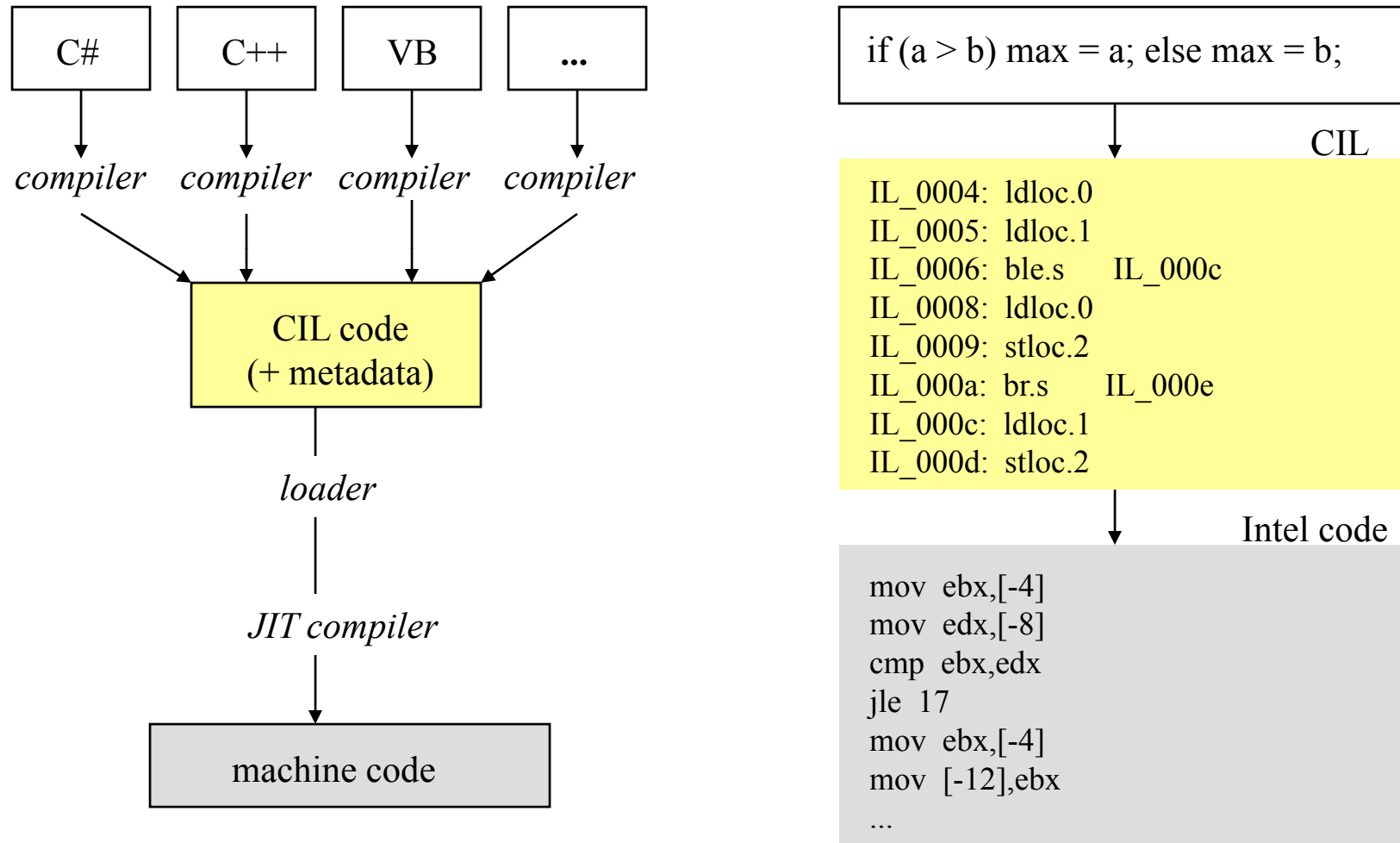
```
class B : A {
  public string s;
  public void Bar() {...}
}
```

used in Eiffel

```
class Client feature
  obj: B;
  ...
  create obj;
  obj.Bar;
  ...
end
```

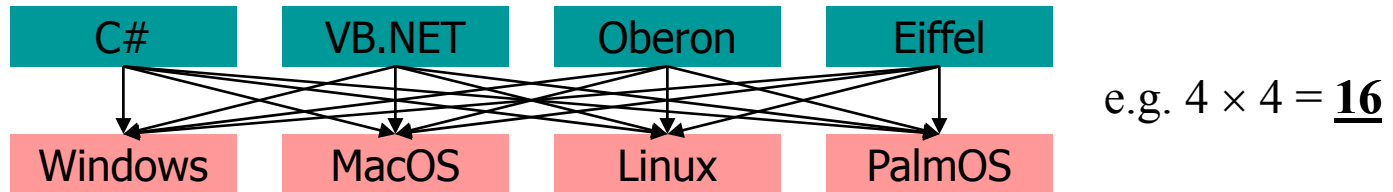
Mục đích .NET (tt)

Tương tác giữa các ngôn ngữ lập trình

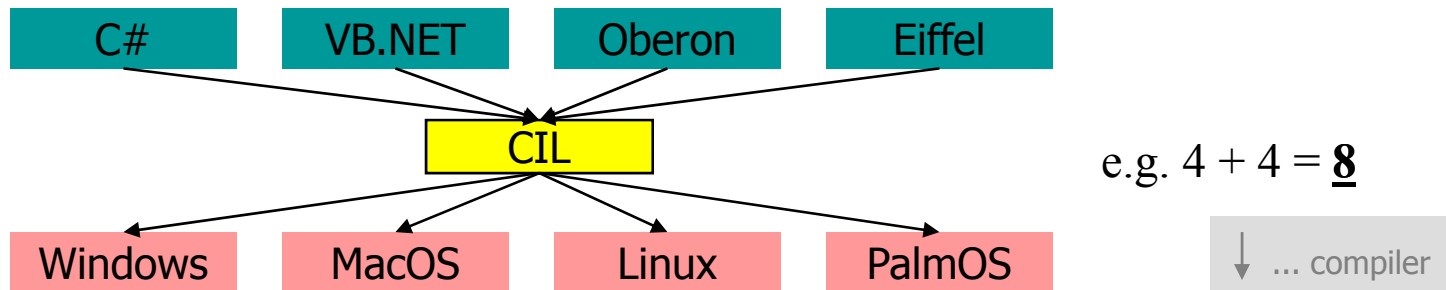


Lợi ích dùng CIL

- *w/o VM*: compilers for each language on each platform



- *w/ VM*: translation into intermediate language (with .NET: CIL)
one compiler per language and
one CLR (JIT compiler) per platform



Giới thiệu ASP.NET

- ❑ ASP.NET là một công nghệ có tính cách mạng dùng để phát triển các ứng dụng về mạng hiện nay cũng như trong tương lai (ASP.NET is a revolutionary technology for developing web applications)
- ❑ Một công nghệ server-side mới, mạnh mẽ tạo các trang web động (A new and powerful server-side technology for creating dynamic web pages)

Lợi ích của việc sử dụng ASP

- ☐☐☐ Công nghệ Server-side scripting
- ☐☐☐ Tự động biên dịch
- ☐☐☐ Cho phép tạo các ứng dụng web đơn giản nhanh chóng và dễ dàng
- ☐☐☐ Tạo trang web có tích hợp nội dung động
- ☐ Sử dụng hiệu quả các đối tượng COM: ADO (ActiveX Data Object), xử lý dữ liệu, FSO (File System Object), làm việc với hệ thống tập tin,...
- ☐ Hỗ trợ nhiều ngôn ngữ: JavaScript, VBScript

Bất lợi của ASP

- Chỉ hỗ trợ 2 loại ngôn ngữ : VBScript và JavaScript
- Thông dịch mã lệnh ASP
- Pha trộn code, HTML và text
- Tương thích trình duyệt
- Không hỗ trợ cache
- Cơ chế debug kém
- Tái sử dụng code kém

Các điểm nổi bật của ASP.NET

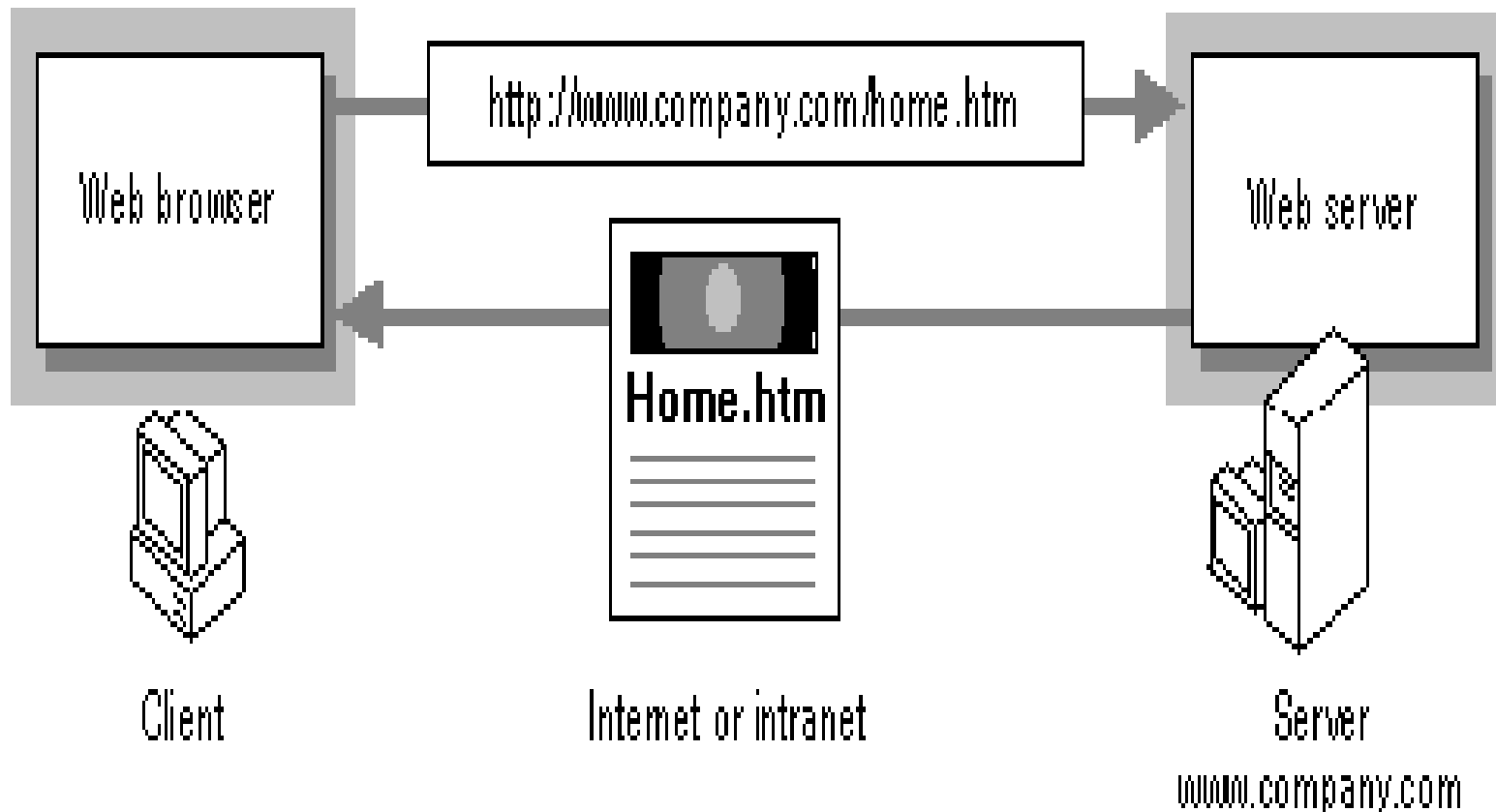
- ❑ Hỗ trợ đa ngôn ngữ : >20 C#, C++, VB.NET, Java, Eiffel, Fortran, Cobol, ML, Haskell, Pascal, Oberon, Perl, Python, ...
- ❑ Biên dịch các trang trước, giúp làm tăng tốc độ thực hiện
- ❑ ASP.NET sử dụng code behind
- ❑ Quản lý trạng thái của các control
- ❑ Có cơ chế hỗ trợ debug
- ❑ Hỗ trợ tái sử dụng code thông qua cơ chế kế thừa
- ❑ Sử dụng cơ chế server-side caching
- ❑ Tự động nhận dạng trình duyệt người dùng đang sử dụng
- ❑ Bộ thư viện phong phú .NET: làm việc với XML, Web Service, ADO.NET
- ❑ Global.asax hỗ trợ nhiều sự kiện hơn

KHÁC BIỆT GIỮA ASP.NET VÀ ASP

- Tập tin của ASP.NET - .ASPX, còn tập tin của ASP - .ASP.
- Tập tin của ASP.NET được phân tích ngữ pháp (parsed) bởi XSPISAPI.DLL, còn ASP ASP.DLL.
- ASP.NET dùng event driven, còn ASP được thi hành theo thứ tự tuần tự từ trên xuống dưới.
- ASP.NET xử dụng trình biên dịch (compiled code) nên rất nhanh, còn ASP dùng trình thông dịch (interpreted code) do đó hiệu suất và tốc độ phát triển cũng thua sút hẳn.
- ASP.NET : 25 ngôn ngữ lập trình mới với .NET và chạy trong môi trường biên dịch (compiled environment), còn ASP: VBScript và JavaScript
- ASP.NET yểm trợ tất cả các browser và quan trọng hơn nữa là yểm trợ các thiết bị lưu động (mobile devices).

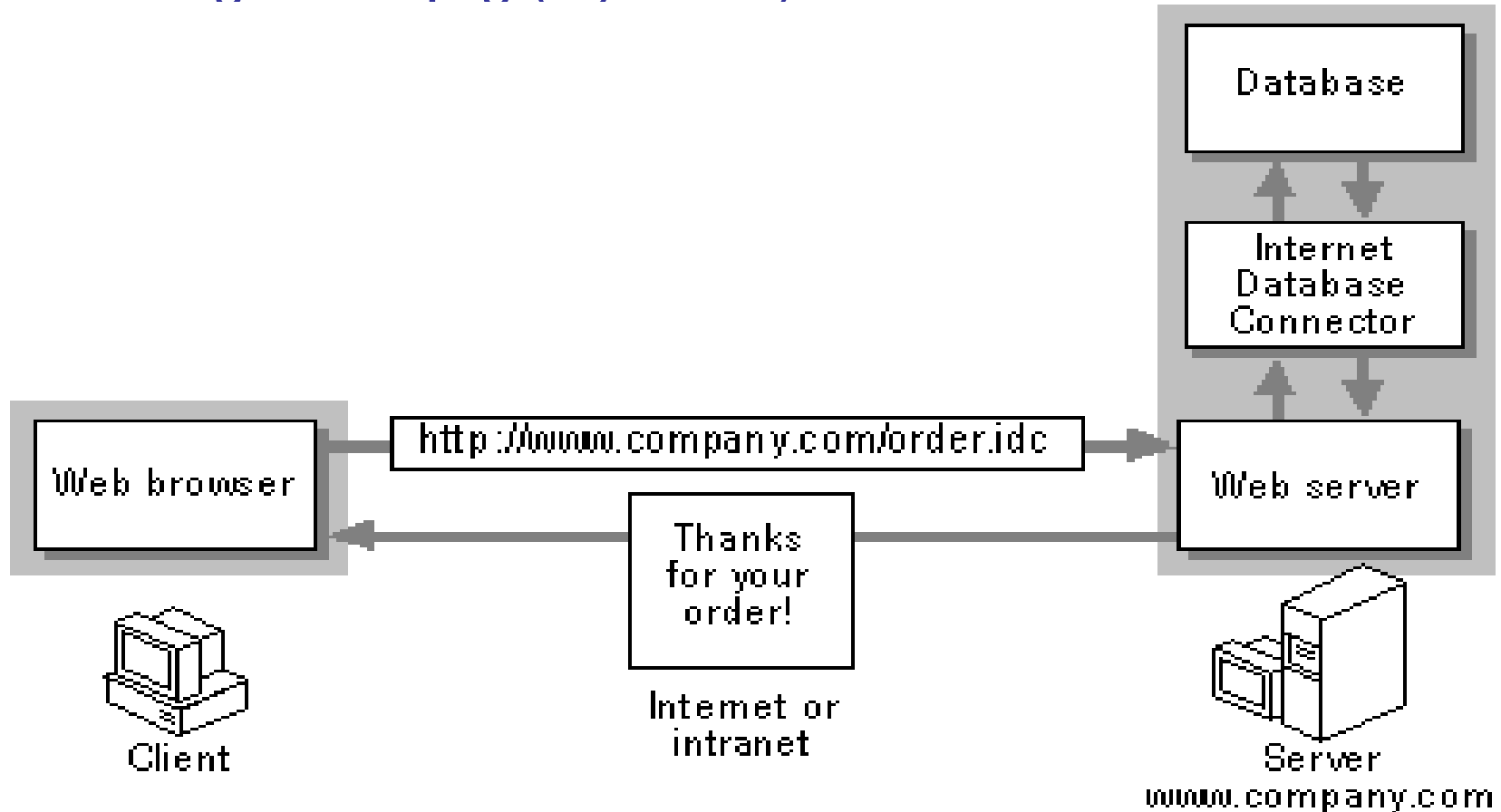
Cách thức làm việc

☐ Trang HTML

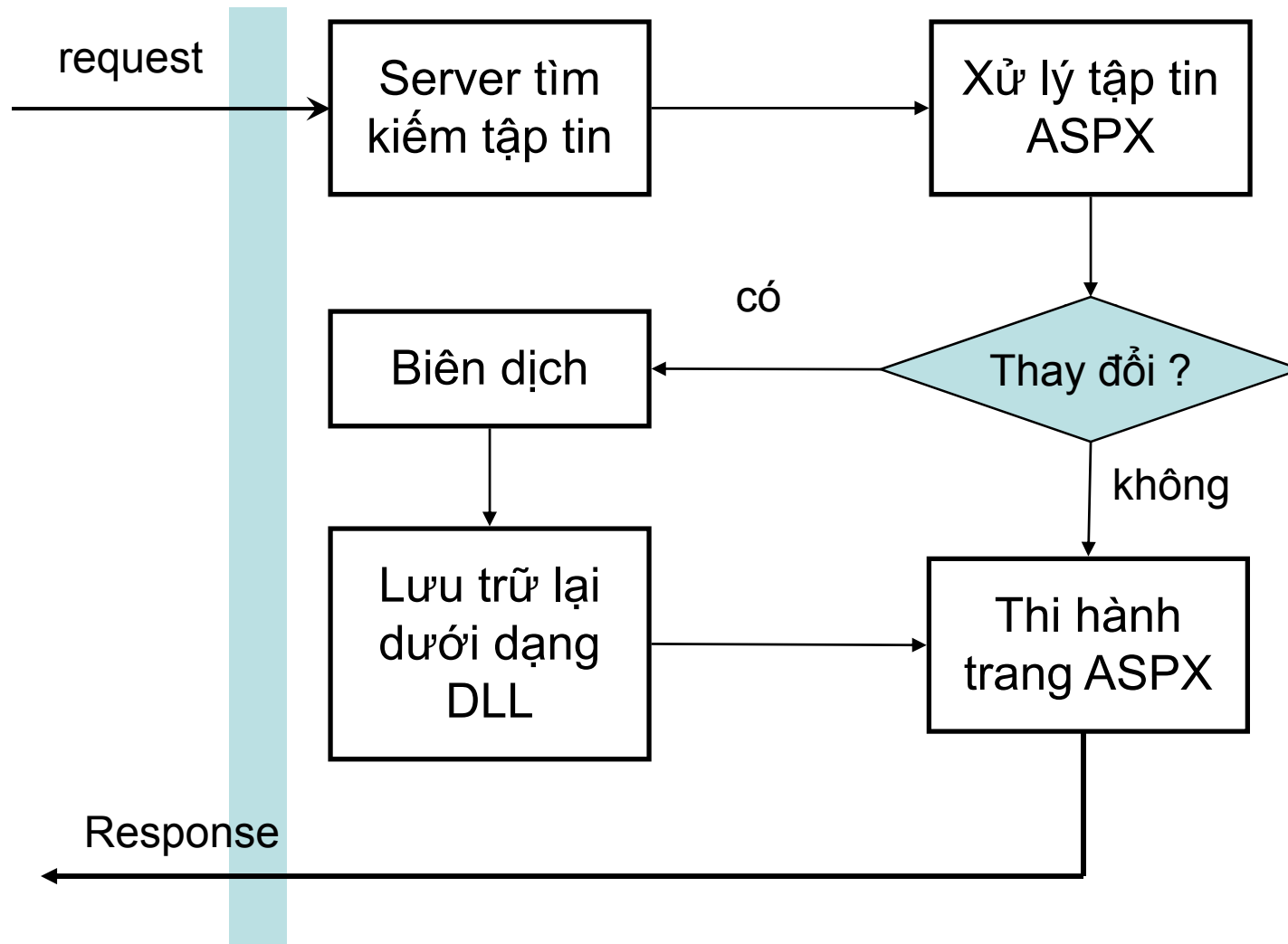


Cách thức làm việc

Trang web động (Dynamic)



Quá trình xử lý tập tin ASPX

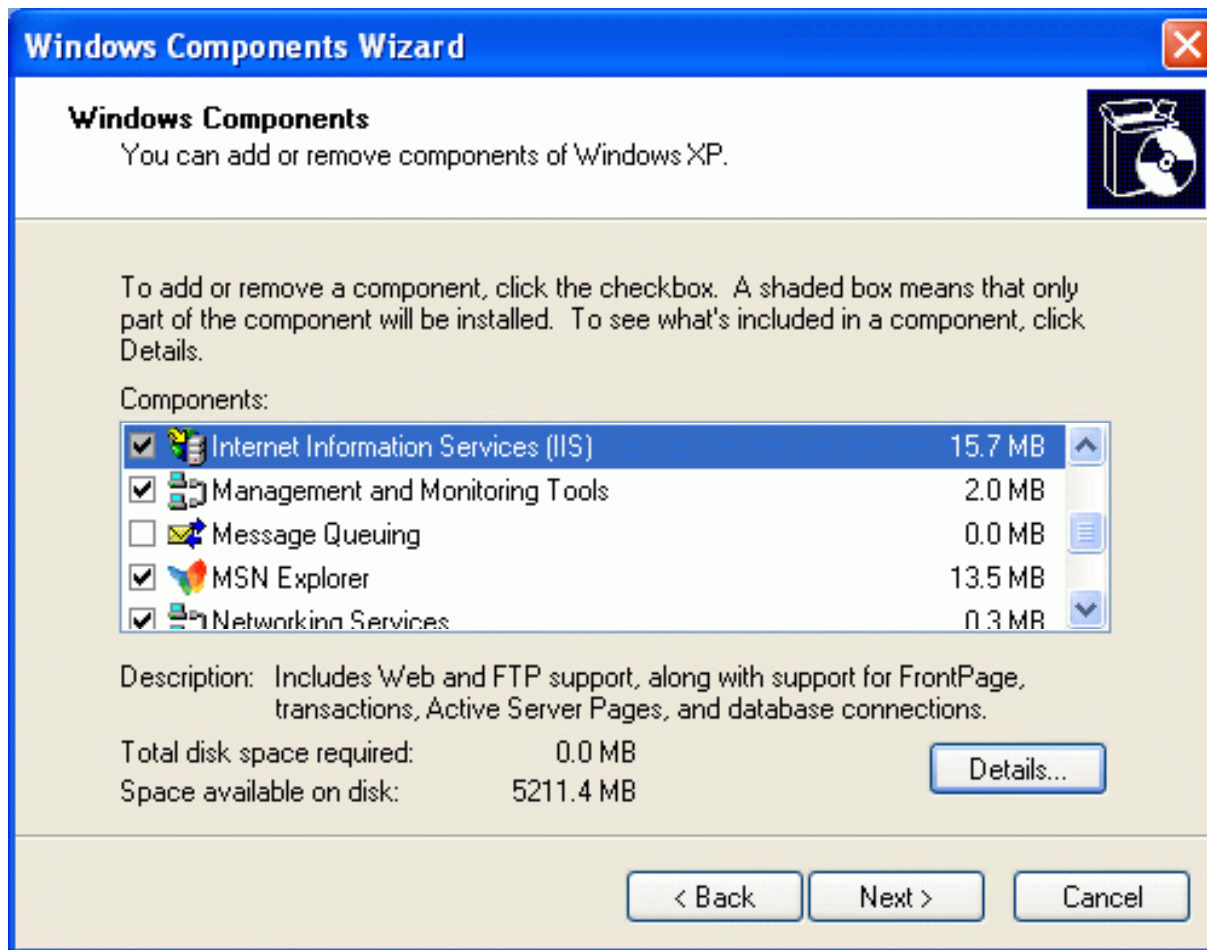


Cài đặt các chương trình đòi hỏi cho ASP.NET

- ❑ **Internet Information Server (IIS)** và bố trí Virtual Directory
 - Windows XP tích hợp IIS 5.5
 - Nếu chưa có thì phải cài thêm
- ❑ **MS Visual Studio.NET** - trong trường hợp này thì MS Visual Studio.NET đã cài sẵn .NET Framework SDK cho ta dùng với ASP.NET hoặc là dùng Web Matrix
- ❑ **.NET Framework Software Development Kit (SDK)**

Cài đặt IIS trong WinXP

1. Start->Settings-> Control Panel-> Add/Remove Programs và nhấp đôn (click) Add/Remove Windows Components, xong chọn Internet Information Server như sau:



IIS – Internet Information Server

□ IIS:

- Ứng dụng server
- Chuyển giao thông tin bằng giao thức chuyển đổi siêu văn bản HTTP

□ IIS làm được gì?

- Xuất bản 1 Home page lên internet
- Tạo các giao dịch thương mại trên internet
- Truy xuất cơ sở dữ liệu từ xa

IIS – Internet Information Server

□ Hoạt động của IIS:

Quá trình dịch và điều phối các trang ASP.Net như sau:

- IIS dịch URL sang path cục bộ (vd: d:\domains\thisSite\wwwroot\site\index.aspx)
- Sử dụng Common LanguageRuntime compiler dịch sang ngôn ngữ trung gian MSIL, rồi sang mã nhị phân để thực thi.
- Mã nhị phân được chứa tạm vào 1 vùng. Nếu lần gọi sau cần sử dụng mà trang web ko có gì thay đổi thì sẽ lấy sử dụng mà ko cần dịch lại. Nếu có thay đổi thì sẽ dịch lại và cập nhật vào vùng này.
- Server trả về kết quả tương ứng với trình duyệt của client.
- Trình duyệt client xử lý phần hiển thị.

Giới thiệu về WEB MATRIX

Web Studio → Mongoose → Project Saturn → Tahiti Project → Microsoft ASP.NET Web Matrix.

-Free ware

- Tạo ra từ C#

- 800 class (.NET Framework >3000)

- Rất nhỏ: 1.3MB (dạng đóng gói)

KHÁC BIỆT GIỮA VISUAL STUDIO.NET VÀ WEB MATRIX

- VS.NET dùng làm đủ mọi thứ, WM chỉ làm ASP.NET
- WM support FTP, VS không (1.0,1.1)
- WM có ASP.NET Web Matrix Server, còn VS có ASP.NET Development Server
- VS support IntelliSense và Debugging, còn WM không

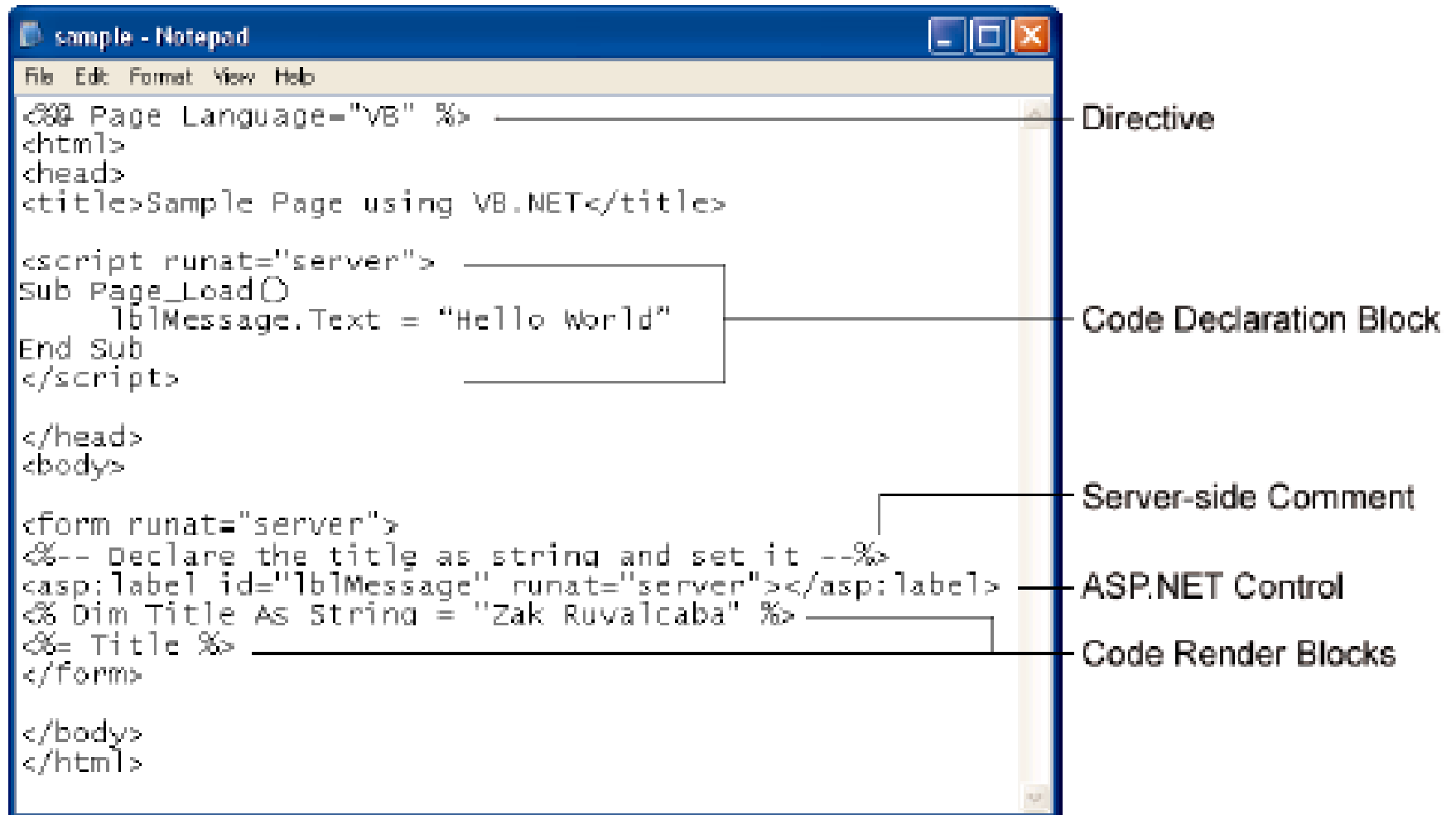
Cấu trúc trang ASP.NET

- Ngôn ngữ lập trình trong ASP.NET
- Directives
- Code declaration blocks
- Code render blocks
- ASP.NET server control
- server-side comments
- Server-side include directives
- Literal text and HTML tags

Ngôn ngữ lập trình ASP.NET

- **VB.NET** : dễ hiểu và được cho là đơn giản nhất, nên đi theo cách này nếu đã sử dụng thành thạo VB
- **C#** : Gần với Java và C++, có nhiều chức năng hơn. Nên chọn C# nếu am hiểu sâu về C++ và Java
- **J#**: Ít được sử dụng, nhưng sẽ tốt với ai thông thạo Java
- Các ngôn ngữ khác: C++.NET, Smalltalk.NET, COBOL.NET, Eiffel.NET, Perl.NET, Component Pascal.NET, Mercury.NET, Oberon.NET, Python.NET,...

ASP.NET Page structure (tt)



The image shows a Notepad window titled "sample - Notepad" containing ASP.NET code. The code is annotated with lines pointing to various components:

- Directive:** Points to the first line of code: `<% Page Language="VB" %>`
- Code Declaration Block:** Points to the code block within a script tag: `Sub Page_Load()
 lblMessage.Text = "Hello World"
End Sub`
- Server-side Comment:** Points to the comment line: `<!-- Declare the title as string and set it -->`
- ASP.NET Control:** Points to the control declaration: `<asp:label id="lblMessage" runat="server"></asp:label>`
- Code Render Blocks:** Points to the code block within a code block: `<% Dim Title As String = "Zak Ruvalcaba" %>`

```
<% Page Language="VB" %>
<html>
<head>
<title>Sample Page using VB.NET</title>

<script runat="server">
Sub Page_Load()
    lblMessage.Text = "Hello World"
End Sub
</script>

</head>
<body>

<form runat="server">
<!-- Declare the title as string and set it -->
<asp:label id="lblMessage" runat="server"></asp:label>
<% Dim Title As String = "Zak Ruvalcaba" %>
<%= Title %>
</form>

</body>
</html>
```


Directives

- ❑ Tự động khai báo mỗi khi trang aspx được tạo ra
- ❑ Cung cấp cho ASP.NET những thông tin đặc biệt để ASP.NET biết cách mà đối xử cũng như những thông tin dùng trong tiến trình biên dịch (during the compiling process)
- ❑ Cung cấp các chỉ dẫn thường được sử dụng trong trang như: *Page, Assembly, Control, Implements, Import, Master, OutputCache, PreviousPageType, Reference* và *Register*
- ❑ Cho phép *Import* các class vào để dùng trong trang web này

**VD: <%@ Page Language="VB"
AutoEventWireUp="true"
CodeFile="Default.aspx.vb"
Inherits="_Default" %>**

Code declaration blocks

VB.NET

```
<script runat="server">  
Sub mySub()  
    ' Code here  
End Sub  
</script>
```

lblMessage.Text = "Hello LH"

lblMessage.Text = "Hello LH" ;

C#

```
<script runat="server">  
void mySub() {  
    // Code here  
}  
</script>
```

Other attributes

Tags “< script>.....</script>” cho phép các thuộc tính khác

VB.NET

```
<script runat="server" language="VB">
```

C#

```
<script runat="server" language="C#">
```

VB.NET

```
<script runat="server" language="VB" src="mycodefile.vb">
```

C#

```
<script runat="server" language="C#" src="mycodefile.cs">
```

Code render blocks

- Dùng trực tiếp trong html code trong `<% và %>`
- Là phương pháp cổ điển ASP, thường dùng `Response.Write()`

VB.NET

```
<% Dim Title As String = "Zak Ruvalcaba" %>  
<%= Title %>
```

This equates to the following C#:

C#

```
<% String Title = "Zak Ruvalcaba"; %>  
<%= Title %>
```

ASP.NET server control

Có 4 dạng Server control cơ bản:

- ASP.NET controls
- HTML controls
- Validation controls
- User control

Tất cả ASP.NET controls phải nằm trong

```
<form runat="server">
```

để thực hiện đúng chức năng ngoại trừ
HtmlGenericControl và Label Web control

server-side comments

Tương tự như HTML truyền thống dùng

`<!---->`, ASP.NET dùng `<%----%>`

Ví dụ: `<%-- Hàm tính diện tích hình vuông --%>`

VB.NET

```
<!--  
<button runat="server" id="myButton" onServerClick="Click">Click  
Me</button>  
<% Title = "New Title" %>  
-->
```

Server-side include directives

- Dùng đặc tính file

```
<!-- #INCLUDE file="myinclude.aspx" -->
```

- Virtual server-side include

```
<!-- #INCLUDE virtual="/directory1/myinclude.aspx" -->
```


Literal text and HTML tags

VB.NET

```
<%@ Page Language='VB' %>
<html>
<head>
<title>Sample Page</title>

<script runat="server">
Sub ShowMessage(s As Object, e As EventArgs)
    lblMessage.Text = "Hello World"
End Sub
</script>

</head>
<body>

<form runat="server">
<%-- Declare the title as string and set it --%>
<asp:Label id="lblMessage" runat="server" />
<% Dim Title As String = "Zak Ruvalcaba's Book List" %>
<%= Title %>
</form>

</body>
</html>
```

Xử lý sự kiện

Xử lý sự kiện trong ASP

```
<html>
<head>
  <title>Sample Page using VBScript</title>
</head>
<body>
<form method="post" action="sample.asp">
  <input type="text" name="txtName" />
  <input type="Submit" name="btnSubmit" text="Click Me" />
<%
If Request.Form("txtName") <> "" Then
  Response.Write(Request.Form("txtName"))
End If
%>

</form>
</body>
</html>
```

Xử lý sự kiện (tt)

Xử lý sự kiện trong ASP.NET

VB.NET

File: ViewState.aspx

```
<html>
<head>
<title>Sample Page using VB.NET</title>
<script runat="server" language="VB">
Sub Click(s As Object, e As EventArgs)
    lblMessage.Text = txtName.Text
End Sub
</script>
</head>

<body>
<form runat="server">
    <asp:TextBox id="txtName" runat="server" />
    <asp:Button id="btnSubmit" Text="Click Me" OnClick="Click"
        runat="server" />
    <asp:Label id="lblMessage" runat="server" />
</form>
```

Ví dụ xử lý sự kiện

```
<%@ Page Language="VB" %>
```

```
<script runat="server">
```

```
Sub tbMessage_Change (Sender AS Object, E As EventArgs)  
    lblmessage.Text = "Chào bạn " + tbMessage.Text
```

```
End Sub
```

```
</script>
```

```
<html>
```

```
<body>
```

```
<font size="5">Simple Application</font><hr><p>
```

```
<form runat="server">
```

```
Please enter your name:
```

```
<asp:textbox id="tbMessage" OnTextChanged="tbMessage_Change" runat=server/>
```

```
<asp:button id="btSubmit" Text="Submit" runat=server/><p>
```

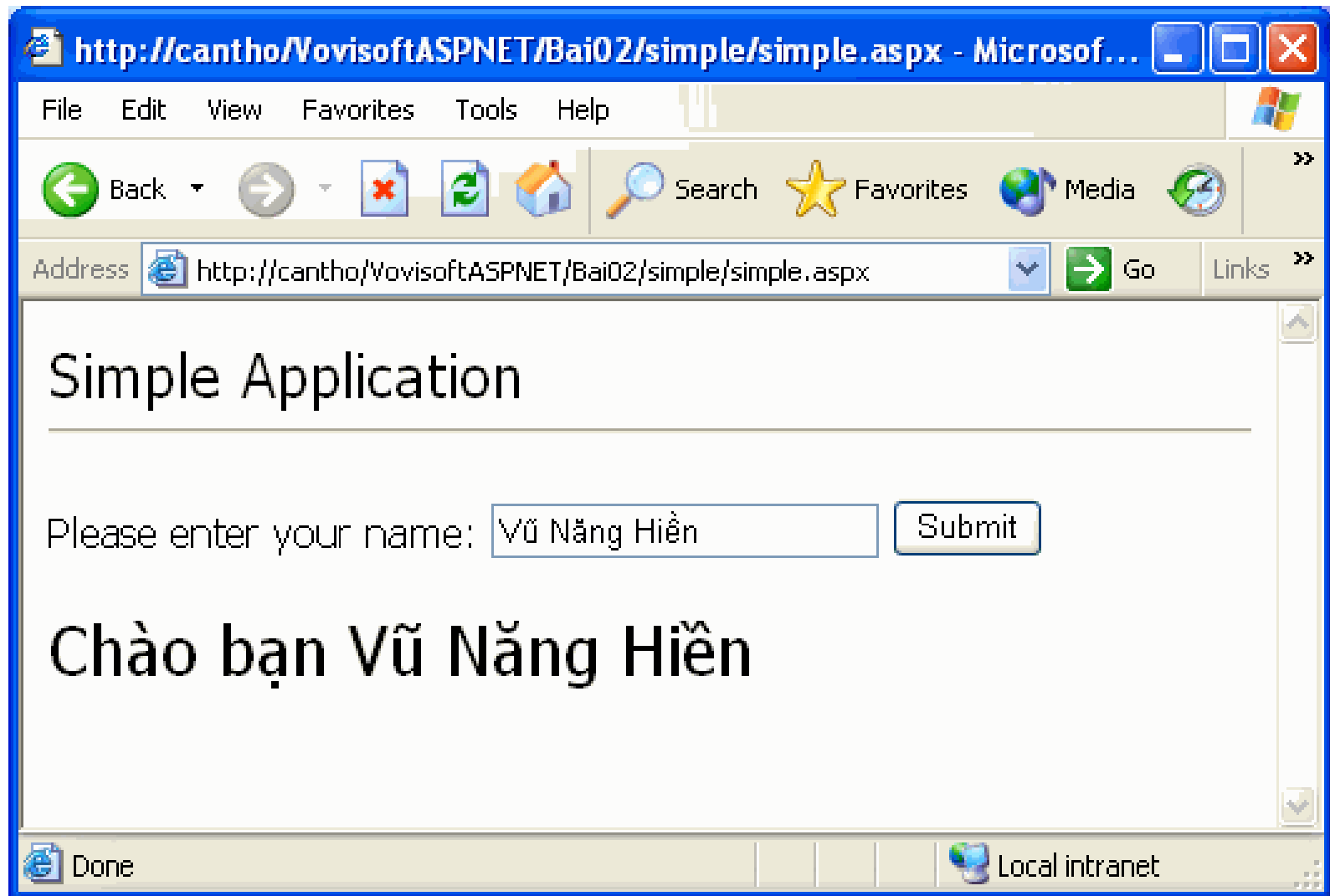
```
<asp:label id="lblMessage" font-size="20pt" runat=server/>
```

```
</form>
```

```
</body>
```

```
</html>
```

Kết quả ví dụ

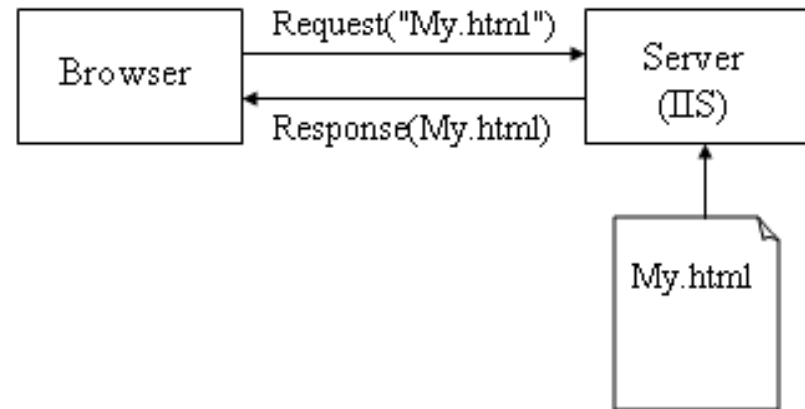


Trang html tĩnh

Pure HTML

My.html

```
<html>
  <head>
    <title>Simple HTML page</title>
  </head>
  <body>
    <h1>Welcome</h1>
    You are visitor number 1!
  </body>
</html>
```



Trang aspx động đơn giản

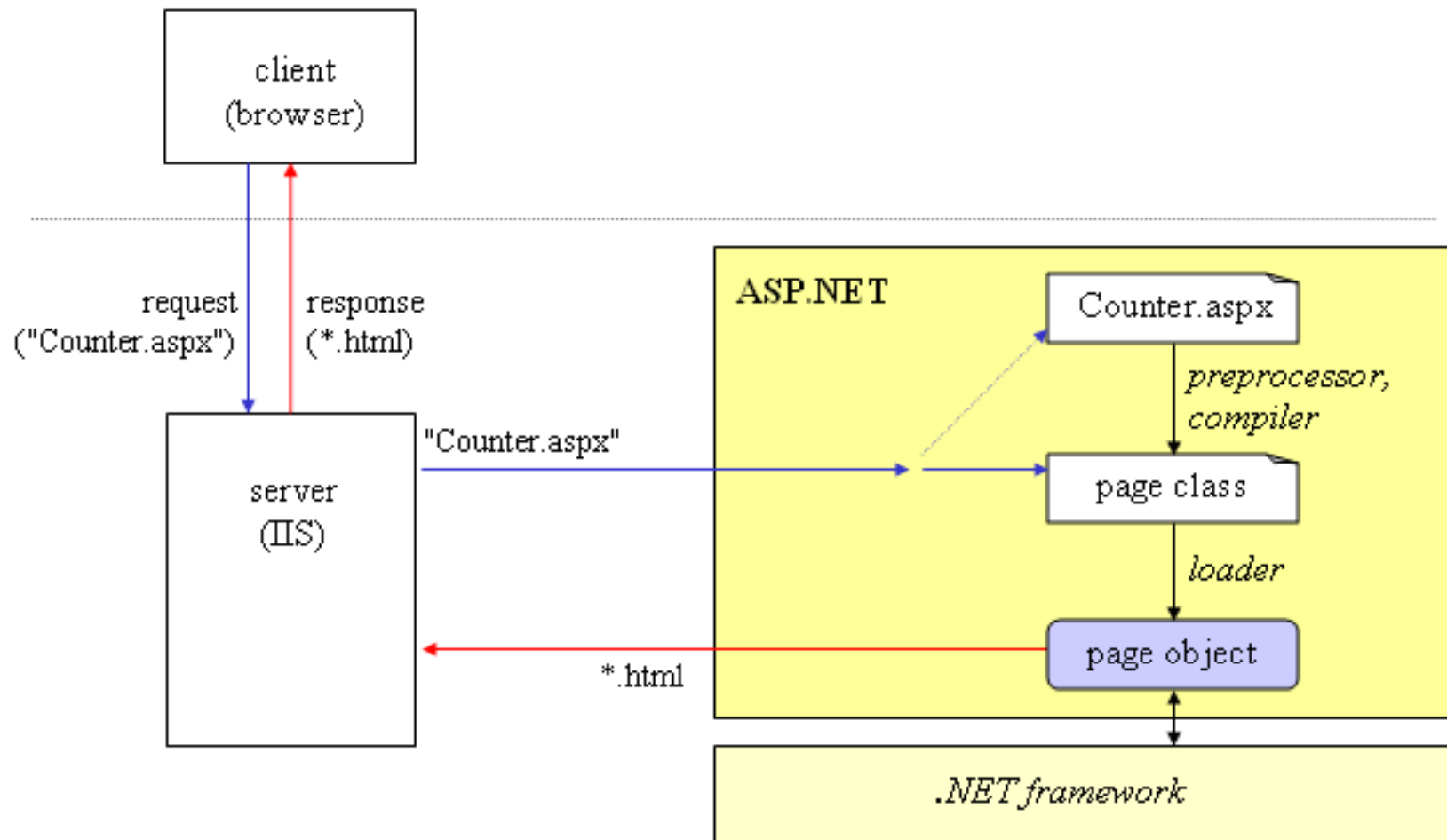
Computed values can be inserted into HTML code

Counter.aspx

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.IO" %>
<html>
  <head> <title>Page counter</title> </head>
  <body>
    <h1>Welcome</h1>
    You are visitor number <%
    FileStream s = new FileStream("c:\\Data\\Counter.dat", FileMode.OpenOrCreate);
    int n;
    try {
      BinaryReader r = new BinaryReader(s);
      n = r.ReadInt32();
    } catch { n = 0; } // if the file is empty
    n++;
    s.Seek(0, SeekOrigin.Begin);
    BinaryWriter w = new BinaryWriter(s);
    w.Write(n); s.Close();
    Response.Write(n);
    %> !
  </body>
</html>
```



Cách hoạt động



HTML code được trả về từ server

Counter.aspx

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.IO" %>
<html>
  <head><title>Page Counter</title></head>
  <body>
    <h1>Welcome</h1>
    You are visitor number <%
      FileStream s = new FileStream(...);
      ...
      Response.Write(n);
    %> !
  </body>
</html>
```

Returned HTML code

```
<html>
  <head><title>Page counter</title></head>
  <body>
    <h1>Welcome</h1>
    You are visitor number 6 !
  </body>
</html>
```

- does not contain any script code
- any browser can display this HTML

Code trong script tag

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.IO" %>
<html>
  <head>
    <title>Page counter</title>
    <script Language="C#" Runat="Server">
      int CounterValue() {
        FileStream s = new FileStream("c:\\Data\\Counter.dat", FileMode.OpenOrCreate);
        ...
        n = r.ReadInt32();
        n++;
        ...
        return n;
      }
    </script>
  </head>
  <body>
    <h1>Welcome</h1>
    You are visitor number <%=CounterValue()%> !
  </body>
</html>
```

Counter.aspx



short form for
Response.Write(CounterValue());

Code behind

```
<%@ Page Language="C#" Inherits="CounterPage" CodeFile="CounterPage.cs" %>
<html>
  <head> <title>Page counter</title> </head>
  <body>
    <h1>Welcome</h1>
    You are visitor number <%=CounterValue()%> !
  </body>
</html>
```

CounterPage.cs

```
using System.IO;

public partial class CounterPage : System.Web.UI.Page {
    public int CounterValue() {
        FileStream s = new FileStream("c:\\Data\\Counter.dat", FileMode.OpenOrCreate);
        ...
        n = r.ReadInt32();
        n++;
        ...
        return n;
    }
}
```

Trang ASP.NET đầu tiên

VB.NET

File: FirstPage.aspx

```
<html>
<head>
<title>My First ASP.NET Page</title>
<script runat="server" language="VB">
Sub Page_Load(s As Object, e As EventArgs)
    lblTime.Text = DateTime.Now.ToString()
End Sub
</script>
</head>

<body>
<p>Hello there!</p>
<p>The time is now: <asp:Label runat="server" id="lblTime" /></p>
</body>
</html>
```

Trang ASP.NET đầu tiên (tt)

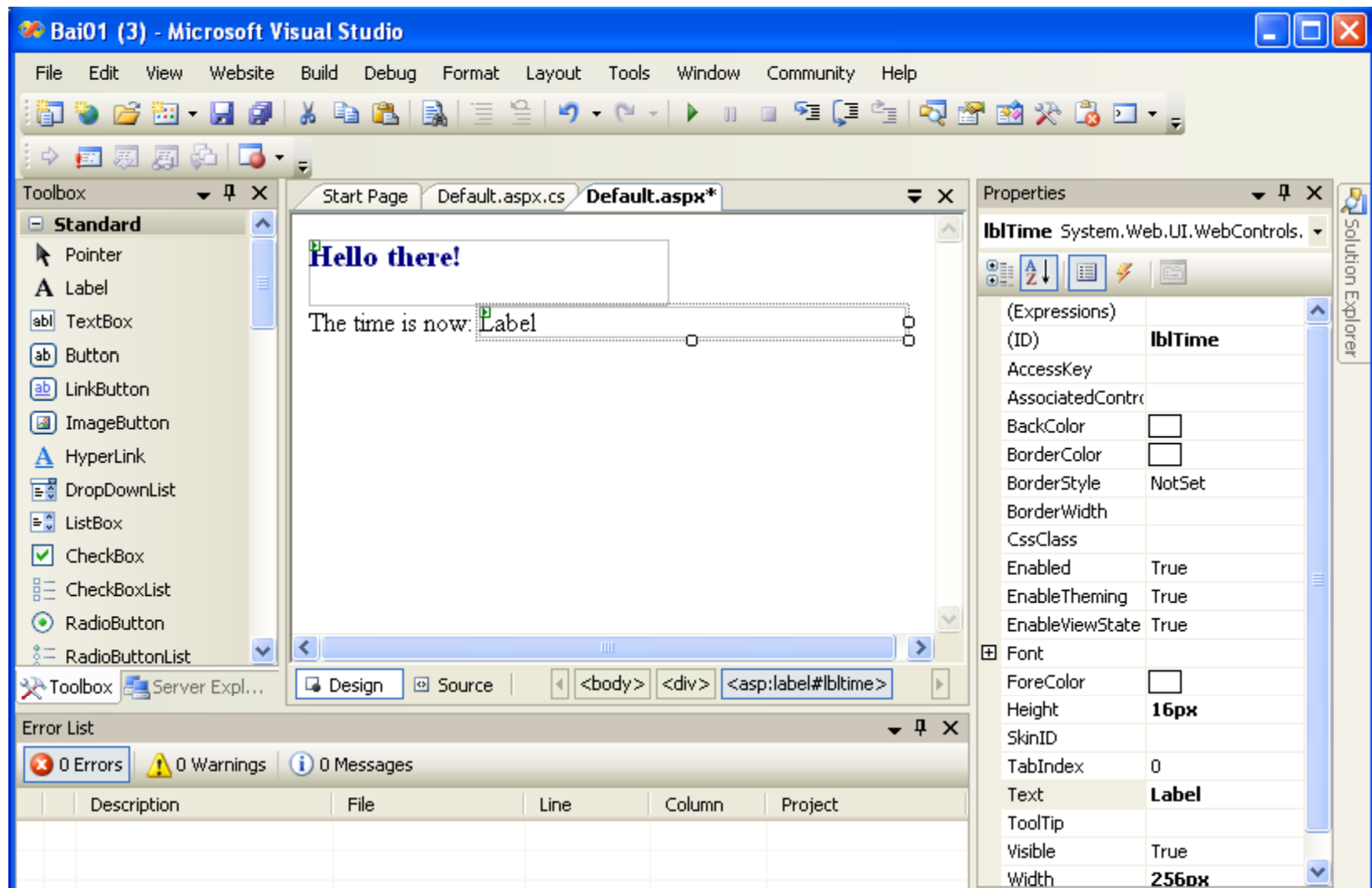
C#

File: FirstPage.aspx

```
<html>
<head>
<title>My First ASP.NET Page</title>
<script runat="server" language="C#">
protected void Page_Load(Object s, EventArgs e)
{
    lblTime.Text = DateTime.Now.ToString();
}
</script>
</head>

<body>
<p>Hello there!</p>
<p>The time is now: <asp:Label runat="server" id="lblTime" /></p>
</body>
</html>
```

Trang ASP.NET đầu tiên (tt)



Trang ASP.NET đầu tiên (tt)

□ Code tự động sinh ra

– trong file *default.aspx*

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeFile="Default.aspx.cs" Inherits="_Default" %>
```

– Trong file *default.aspx.cs*

```
using System;  
using System.Data;  
using System.Configuration;  
using System.Web;  
using System.Web.Security;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using System.Web.UI.WebControls.WebParts;  
using System.Web.UI.HtmlControls;  
public partial class _Default : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
    }  
}
```

```
lblTime.Text = DateTime.Now.ToString();
```



Chương 2: Web Server Control

-HTML Server Control:

HtmlAnchor, HtmlButton, HtmlLabel, HtmlTextbox,...

- ASP.NET Server Control:

+ ASP.NET Page

+ Điều khiển cơ bản

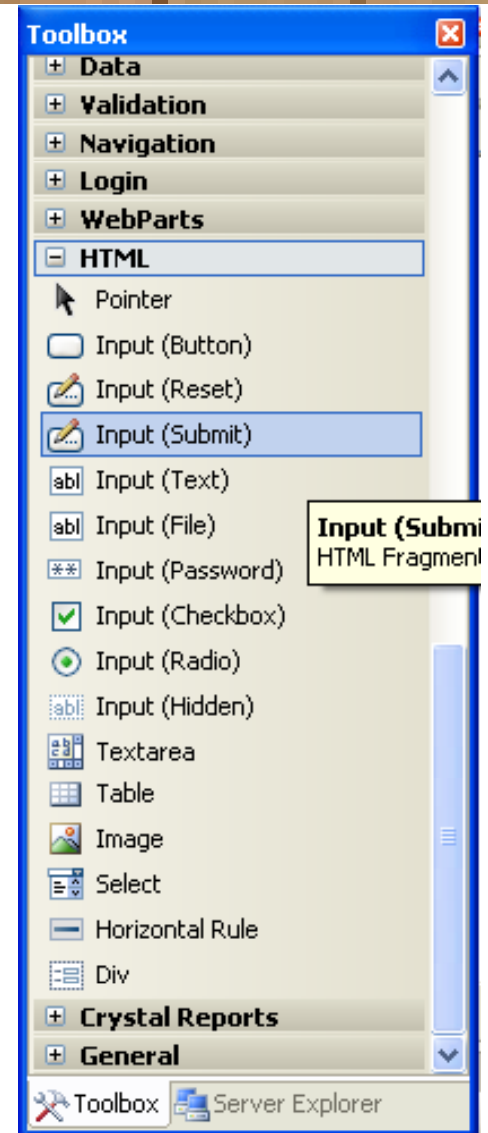
+ Điều khiển kiểm tra dữ liệu

+ Một số điều khiển khác

Phần I: HTML Server Control

- Dùng để thể hiện giao diện web
- Tất cả các class HTML control chứa trong namespace `System.Web.UI.HtmlControls`
- HTML Server control là những tag HTML tạo ra
 - Duy trì tương thích với các tag HTML cũ.
 - Thêm vào thuộc tính `runat = "server"`
 - Tất cả HTML Server Control phải được đặt trong
 - Tag `<form>` với thuộc tính `runat = "server"`

ASP.NET



ASP.NET Server Control

- ASP.NET Server Control là những tag đặc biệt của riêng ASP.NET.
- Các control này cũng sẽ được xử lý trên server, và đòi hỏi phải có thuộc tính runat = “server”
- Không tương ứng với HTML tag nào.
- Có thể dùng thể hiện các thành phần phức tạp.

ASP.NET Server control vs HTML control

Tính năng	ASP.NET Server control	HTML control
Server event	Kích hoạt được một số sự kiện cụ thể trên Server	Chỉ có thể kích hoạt các sự kiện mức trang trên server (post-back)
Quản lý trạng thái	Dữ liệu nhập vào control được lưu giữ lại sau mỗi request	Dữ liệu không được lưu giữ lại, phải tự lưu và điền vào sử dụng script
Tương thích	Tự động nhận diện loại trình duyệt và tạo hiển thị cho phù hợp	Không tự động nhận diện trình duyệt
Các thuộc tính	.NET Framework cung cấp một tập các thuộc tính cho mỗi control, cho phép thay đổi phần hiển thị và hành vi thông qua mã lệnh	Chỉ có các thuộc tính chuẩn của HTML

Tại sao sử dụng HTML server control??

Khác biệt trong HTML tag

➤ Server control:

– `<asp:controlname id="some_id" runat="server"/>`

➤ HTML control:

– HTML tag

Ví dụ:

```
- { <asp:TextBox id="txtText" runat="server"></asp:TextBox>
- { <INPUT type="text" id="textfield1">

- { <asp:Button id="btnShow" runat="server"
  { Text="Show"></asp:Button>
- { <INPUT type="button" value="Show">
```

HtmlAnchor

Thiết lập một a server-side HTML `` tag.

VD: `Click Here`

HtmlButton

Tạo một `<button>` tag.

```
<button id="myButton" OnServerClick="Click"
runat="server">Click Here</button>
```

HtmlImage

```

```

```
<button id="myButton" runat="server" OnServerClick="Click">Click
Here</button>
```

```
<script runat="server" language="VB">
Sub Click(s As Object, e As EventArgs)
    myimage.Source = "welcome.gif"
End Sub
</script>
```

Phần II: ASP.NET Web control

Điều khiển sự kiện và chương trình con (Control events and Subroutines)

- Sự kiện sinh ra: bấm chuột, bấm phím ENTER,
load 1 trang web...=>Bộ xử lý sự kiện – Handler

```
<form runat="server">
```

```
<asp:Button id="btn1" runat="server"  
OnClick="btn1_Click" Text="Click Me" />
```

```
<asp:Label id="lblMessage" runat="server" />
```

```
</form>
```

Các sự kiện thường gặp

- **OnClick**, **OnCommand**, **OnLoad**
- **OnInit**
- **OnPreRender**: được xảy ra sau khi Control được nạp, và trước khi control được render (tức là hiển thị ra client/browser)
- **OnUnload**
- **OnDispose**: button được giải phóng khỏi bộ nhớ
- **OnDataBinding**

Page Events – các sự kiện trang

Page_Init -> Page_Load -> Các sự kiện của các control nếu có ->Page_Unload tới đây page được tạo xong và chuyển về hiển thị trên client

- **Page_Init**: Được gọi khi trang được khởi tạo với những cài đặt cơ bản đầu tiên.
- **Page_Load, Page_PreRender, Page_UnLoad**

```
protected void Page_Load(object sender, EventArgs e)  
{  
}
```

Một số thuộc tính của Page

Thuộc tính *IsPostBack*:

- *False*: Page load lần đầu
- *True*: Page load lại

Thuộc tính *SmartNavigation*:

- *True*: giữ nguyên vị trí đang đọc khi reload
- *False* (mặc định): về vị trí đầu

Phần II: ASP.NET Web control

2.1. Web control cơ bản

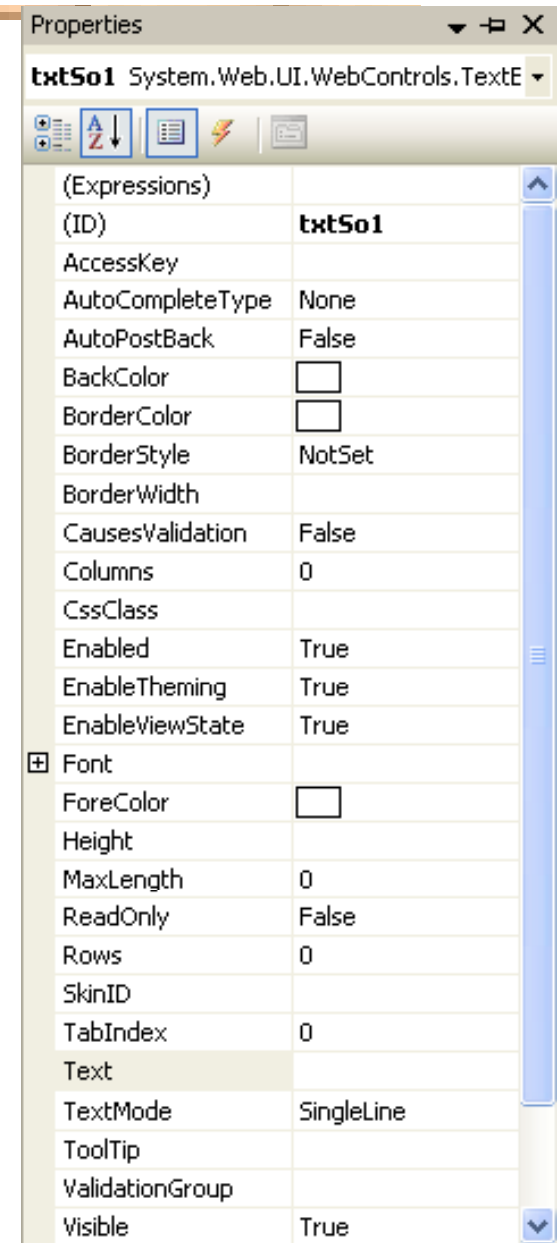
Label	RadioButton
TextBox	RadioButtonList
Button	CheckBox
Image	CheckBoxList
ImageButton	DropDownList
LinkButton	ListBox
HyperLink	Panel
	Placeholder

Thuộc tính chung của các Web control

Thuộc tính	Kiểu	Ý nghĩa
ID	Chuỗi KT	Tên control, duy nhất
AccessKey	String	Ký tự chuyển nhanh đến control, phím nóng
BackColor	Color	Màu nền
BorderColor	Color	Màu đường viền
BorderStyle	BorderStyle	kiểu đường viền
BorderWidth	Unit	độ rộng đường viền
Visible	Boolean	Control có thấy hay không, mặc định – true
Font	FontInfo	Font cho control
ForeColor	Color	Màu font
Height	Unit	chiều cao
Width	Unit	độ rộng
AutoPostBack	boolean	Khi được thiết lập là True, mỗi khi người dùng thay đổi dữ liệu TextBox sẽ kích hoạt sự kiện post-back về server

2.1.1. Label, TextBox, Button

- **Label**: hiển thị nội dung
- **Textbox**: chỉ cho đọc, cho phép nhập chuỗi ký tự thông thường, cho phép nhập mật mã, nhập nhiều dòng văn bản. Các thuộc tính căn bản – *Text, TextMode (SingleLine, MultiLine, Password), Rows, MaxLength, Wrap, AutoPostBack*.
- **Button**: Button thường, ImageButton (ImageUrl, ImageAlign, AlternateText), LinkButton



2.1.1. Label, TextBox, Button (tt)

- **Label**

- `<asp:Label id="Label1" runat="server">Please input text</asp:Label>`

- **Buttons (Button, LinkButton, ImageButton)**

- `<asp:Button id="Button1" runat="server" Text="Button"></asp:Button>`

- `<asp:LinkButton id="LinkButton1" runat="server">LinkButton</asp:LinkButton>`

- `<asp:ImageButton id="ImageButton1" runat="server"></asp:ImageButton>`

- **TextBox**

- `<asp:TextBox id="TextBox1" runat="server"></asp:TextBox>`

2.1.2. ListBox và DropDownList

Các thuộc tính quan trọng

-Items:

Add, Insert, Count, Contains, Remove, RemoveAt, Clear

-Rows

-SelectionMode (trong ListBox): Single, Multiple

-Xử lý mục chọn: SelectedIndex, SelectedItem,
SelectedValue

ListBox và DropDownList

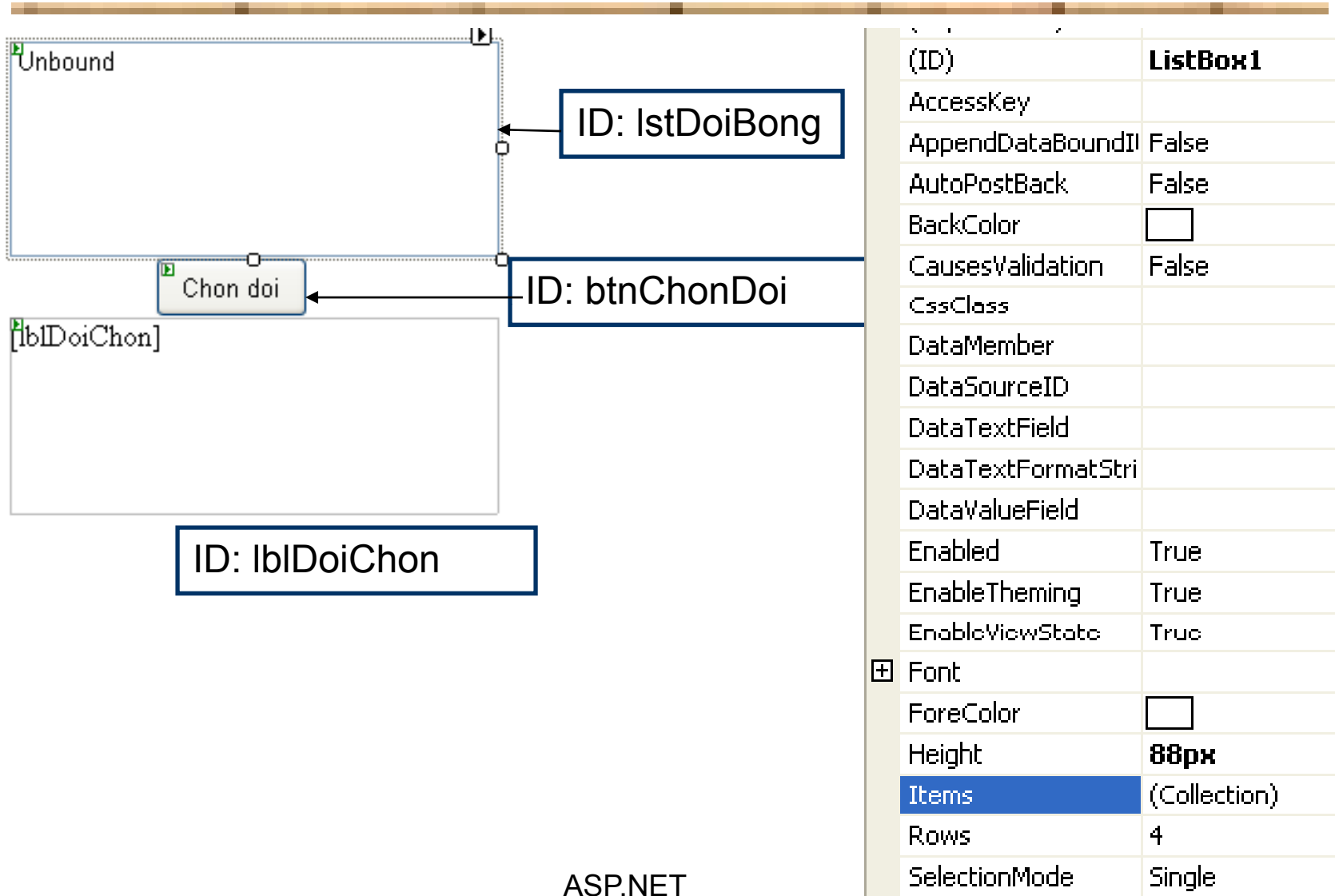
VD: ListBox

```
<asp:ListBox id="listTechnologies" runat="server"
SelectionMode="Multiple">
<asp:ListItem Text="ASP.NET" Value="aspnet" />
<asp:ListItem Text="JSP" Value="jsp" />
<asp:ListItem Text="PHP" Value="php" />
<asp:ListItem Text="CGI" Value="cgi" />
<asp:ListItem Text="Coldfusion" Value="cf" />
</asp:ListBox>
```

VD: DropDownList

```
<asp:DropDownList id="ddlFavColor" runat="server">
<asp:ListItem Text="Red" value="red" />
<asp:ListItem Text="Blue" value="blue" />
<asp:ListItem Text="Green" value="green" />
</asp:DropDownList>
```


2.1.2. ListBox và DropDownList



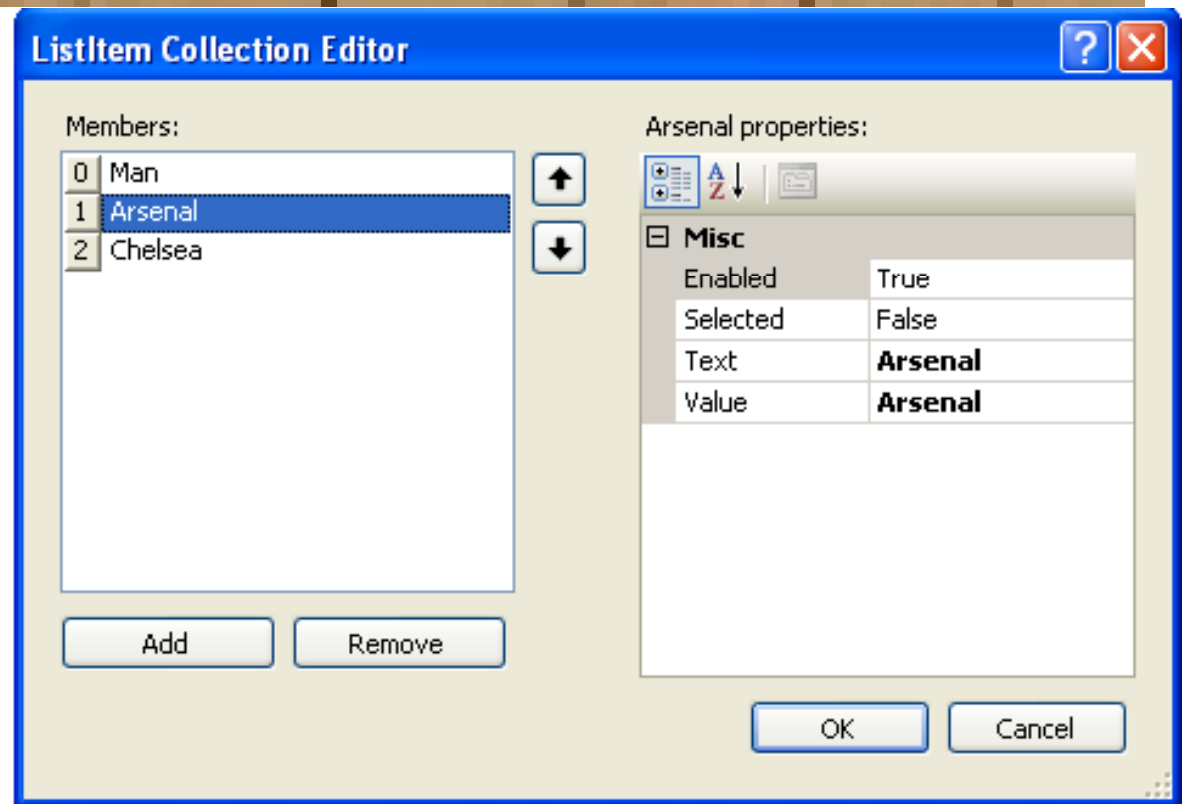
The image shows a Visual Studio IDE with a design view of a web page. A **ListBox** control is selected, and its properties window is open on the right. The design view shows a container with the text "Unbound" and a button labeled "Chon doi". The **ListBox** control is positioned below the button. The properties window shows the following properties for **ListBox1**:

Property	Value
(ID)	ListBox1
AccessKey	
AppendDataBoundItems	False
AutoPostBack	False
BackColor	<input type="text"/>
CausesValidation	False
CssClass	
DataMember	
DataSourceID	
DataTextField	
DataTextFormatString	
DataValueField	
Enabled	True
EnableTheming	True
EnableViewState	True
Font	
ForeColor	<input type="text"/>
Height	88px
Items	(Collection)
Rows	4
SelectionMode	Single

The design view also shows the following IDs for the controls:

- ID: lstDoiBong** (points to the ListBox control)
- ID: btnChonDoi** (points to the "Chon doi" button)
- ID: lblDoiChon** (points to the container area)

Thêm dữ liệu vào ListBox



Trong thời gian chạy

```
protected void btnShow_Click(object sender, EventArgs e)
{
    ListBox1.Items.Add(txtSource.Text);
    DropDownList1.Items.Add(txtSource.Text);
}
```

Lấy dữ liệu được chọn

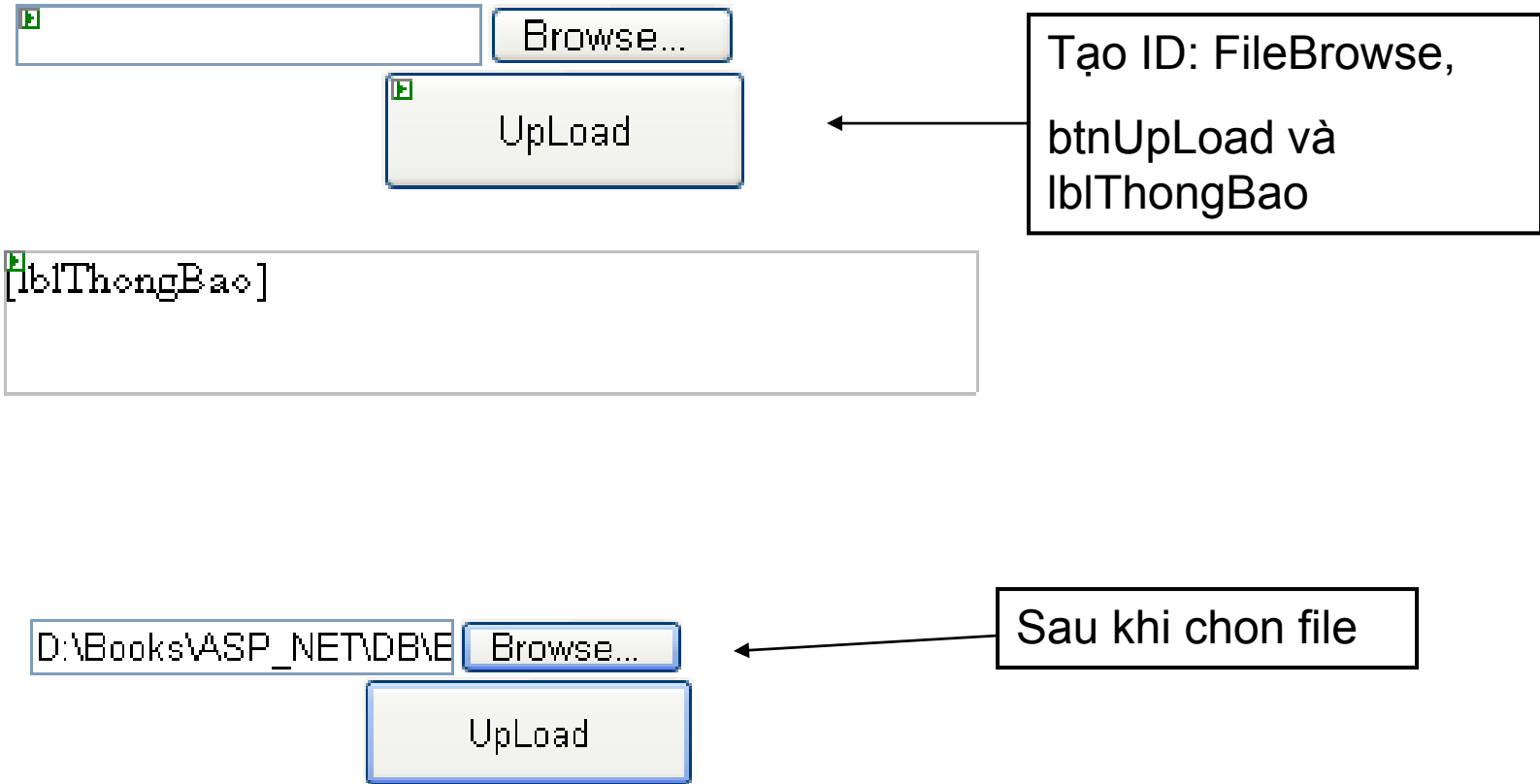
- Dùng thuộc tính **SelectedItem** để lấy mục dữ liệu được chọn hiện tại trong List

```
protected void Page_Load(object sender, EventArgs e)
{
    // Test if there is a selected item.
    if (ListBox1.SelectedItem == null)
        // Display the selected item.
        Label1.Text = "The selected item is: " +
                    ListBox1.SelectedItem.Text;
    else
        Label1.Text = "No item is selected.";
}
}
```

Các thuộc tính quan trọng của ListBox

- **AutoPostBack**
- **Items**
 - Rows
 - SelectionMode
 - Single
 - Multiple
- **Xử lý mục chọn:**
 - SelectedIndex, SelectedItem, SelectedValue
 - Add, Insert
 - Count
 - Contains
 - Remove, RemoveAt, Clear

2.1.3. Upload file



VD: Upload file

```
Protected Sub btnUpload_Click(...)...
```

```
    Dim sTapTin As String
```

```
    Dim FileName As String
```

```
    sTapTin = FileBrowse.PostedFile.FileName
```

```
    FileName = System.IO.Path.GetFileName(sTapTin)
```

```
    FileBrowse.PostedFile.SaveAs(Server.MapPath("Upload\") &_  
FileName)
```

```
    lblThongBao.Text = "Da Upload thanh cong"
```

```
End Sub
```

2.1.4. CheckBox, RadioButton

Các thuộc tính chung:

- Checked: trạng thái mục chọn
- Visible: ẩn/hiện control
- Text: Dòng chữ bên control
- TextAlign: Vị trí hiển thị của control so với dòng chữ (Right, Left)
- GroupName (RadioButton): tên nhóm

2.1.4. CheckBox, RadioButton

Sở thích	<input type="checkbox"/> Ca nhạc <input checked="" type="checkbox"/> Du Lịch	← Element CheckBox
Quốc tịch	<input checked="" type="radio"/> Việt Nam <input type="radio"/> Nước Ngoài	← Element RadioButton GroupName: QuocTich

```
<asp:CheckBox id="chkCaNhac" Text=" Ca nhạc" runat="server" />
```

```
<asp:RadioButton id="rbVietNam" GroupName="QuocTich" Text="Việt Nam"  
runat="server" />
```

```
<asp:RadioButton id="rbNuocNgoai" GroupName="QuocTich" Text="Nước  
Ngoài" runat="server" />
```


2.1.5. CheckBoxList, RadioButtonList

The image shows two ASP.NET controls and a dialog box. The top control is a **CheckBoxList** titled "Theo bạn môn học nào không nên học trong HK1:" with five items: "Kiến trúc máy tính", "Tin học đại cương", "Hệ điều hành", "ASP.NET", and "Không ý kiến". The bottom control is a **RadioButtonList** titled "Đánh giá tài liệu học tập môn học:" with five items: "1 điểm", "2 điểm", "3 điểm", "4 điểm", and "Không ý kiến". Below these is the **ListItem Collection Editor** dialog box, which is currently editing the "3 điểm" item. The dialog shows a list of members with "3 điểm" selected, and a properties table for the selected item.

Theo bạn môn học nào không nên học trong HK1:

- Kiến trúc máy tính
- Tin học đại cương
- Hệ điều hành
- ASP.NET
- Không ý kiến

Đánh giá tài liệu học tập môn học:

- 1 điểm
- 2 điểm
- 3 điểm
- 4 điểm
- Không ý kiến

ListItem Collection Editor

Members:

0	1 điểm
1	2 điểm
2	3 điểm
3	4 điểm
4	Không ý kiến

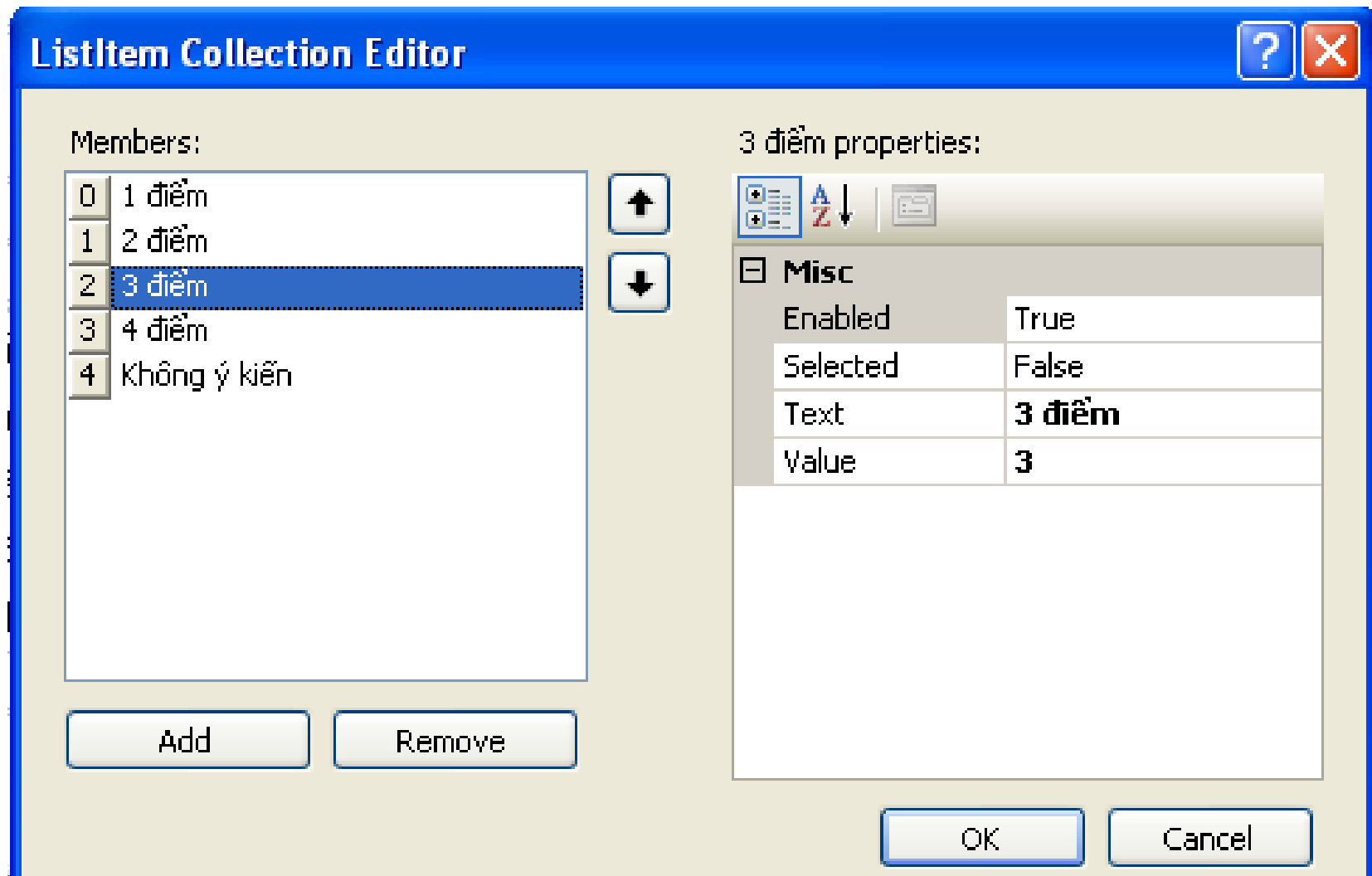
3 điểm properties:

Misc	
Enabled	True
Selected	False
Text	3 điểm
Value	3

Buttons: Add, Remove, OK, Cancel

ASP.NET

2.1.5. CheckBoxList, RadioButtonList



2.1.5. CheckBoxList, RadioButtonList

- **Items**: Add, Insert, Count, Contains, Remove, RemoveAt, Clear

RepeatColumns: số cột hiển thị

RepeatDirection: Hình thức hiển thị (vertical, Horizontal)

ID_Control.SelectedItem.Text

ID_Control.SelectedItem.Value

2.1.5. CheckBoxList, RadioButtonList

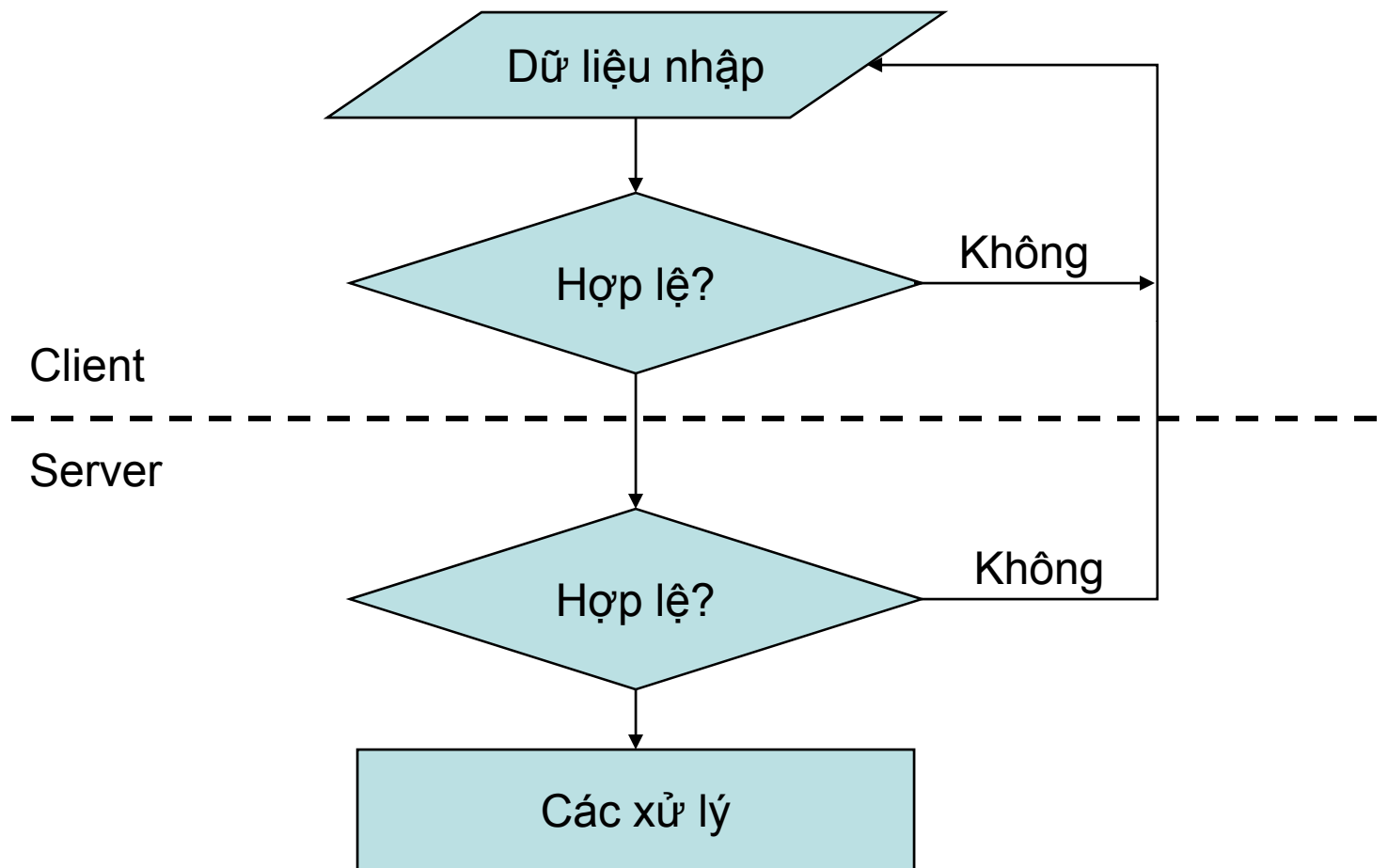
VD CheckBoxList:

```
<asp:CheckBoxList id="chkIFavDrinks" runat="server">  
  <asp:ListItem Text="Pizza" Value="pizza" />  
  <asp:ListItem Text="Tacos" Value="tacos" />  
  <asp:ListItem Text="Pasta" Value="pasta" />  
</asp:CheckBoxList>
```

VD RadioButtonList:

```
<asp:RadioButtonList id="radIFavColor" runat="server">  
  <asp:ListItem Text="Red" Value="red" />  
  <asp:ListItem Text="Blue" Value="blue" />  
  <asp:ListItem Text="Green" Value="green" />  
</asp:RadioButtonList>
```

2.2. Điều khiển kiểm tra dữ liệu Validation web controls



Sơ đồ xử lý kiểm tra dữ liệu nhập tại Client và Server

Validation control

Validation control kiểm tra tính đúng đắn của dữ liệu do client nhập vào trước khi trang được gửi về cho server

Validation control	Sử dụng khi
RequiredFieldValidator	Kiểm tra nếu dữ liệu trong control khác rỗng
CompareValidator	Kiểm tra nếu mục dữ liệu nhập trong control giống với control khác
RangeValidator	Kiểm tra nếu mục dữ liệu nhập trong control nằm trong khoảng 2 giá trị
RegularExpressionValidator	Kiểm tra nếu mục dữ liệu nhập trong control thỏa 1 công thức định dạng chỉ định
CustomValidator	Kiểm tra tính đúng đắn của dữ liệu nhập vào control sử dụng client-side script hoặc a server-side code, hoặc cả 2
ValidationSummary	Hiển thị tất cả các lỗi kiểm tra xảy ra trong trang

Validation control (tt)

Các thuộc tính chung

-ControlToValidate: Tên control cần kiểm tra

-Text: Chuỗi thông báo xuất hiện khi có lỗi

-ErrorMessage: Chuỗi thông báo xuất hiện trong control validation summary

-Display: Qui định hình thức hiển thị

+None: không hiển thị thông báo lỗi (vẫn có kiểm tra dữ liệu)

+ Static, Dynamic

Validation control (tt)

Sử dụng Validation Control

- **Các bước:**
 - Kéo thả 1 validate control vào Web form
 - Thiết lập các thuộc tính cho validate control:
 - ControlToValidate là control bạn muốn kiểm tra
 - ErrorMessage : Thông báo lỗi
 - Text : Hiển thị của validate control
 - Sử dụng ValidationSummary control để hiển thị tất cả các lỗi xảy ra trong trang

Mặc dù việc kiểm tra xảy ra ở client, nhưng nó chỉ thực hiện khi có 1 sự kiện post-back xảy ra

2.2.1. Compare validator control

- Dùng so sánh giá trị của một control với giá trị của một control khác hoặc với 1 giá trị được xác định trước
- Dùng control này để kiểm tra ràng buộc miền giá trị, kiểu dữ liệu lên thuộc tính
- Note!!!không nhập dữ liệu, control sẽ không thực hiện kiểm tra vi phạm

```

<html>
<body>

<form runat="server">
<table border="0" bgcolor="#b0c4de">
  <tr valign="top">
    <td colspan="4"><h4>Compare two values</h4></td>
  </tr>
  <tr valign="top">
    <td><asp:TextBox id="txt1" runat="server" /></td>
    <td> = </td>
    <td><asp:TextBox id="txt2" runat="server" /></td>
    <td><asp:Button Text="Validate" runat="server" /></td>
  </tr>
</table>
<br />

<asp:CompareValidator
  id="compval"
  Display="dynamic"
  ControlToValidate="txt1"
  ControlToCompare="txt2"
  ForeColor="red"
  BackColor="yellow"
  Type="String"
  EnableClientScript="false"
  "
  Text="Validation Failed!"
  runat="server" />

</form>

</body>
</html>

```

Compare two values

=

Validation Failed!

2.2.2. CustomValidator control

- Cho phép viết phương thức xử lý kiểm tra lỗi
- ClientValidationFunction: Chỉ ra tên hàm xử lý lỗi client-side
- OnServerValidate: Hàm xử lý lỗi server-side
- ControlToValidate: ID của control cần kiểm tra

```

<html> <body>
<form runat="server">
<asp:Label runat="server" Text="Enter a username: "
/>
<asp:TextBox id="txt1" runat="server" />
<asp:Button Text="Submit" runat="server"/>
<br /> <asp:Label id="mess" runat="server"/> <br />
<asp:CustomValidator
ControlToValidate="txt1"
OnServerValidate="user"
Text="A username must be between 8 and 16
characters!"
runat="server"/> </form>
</body> </html>

```

```

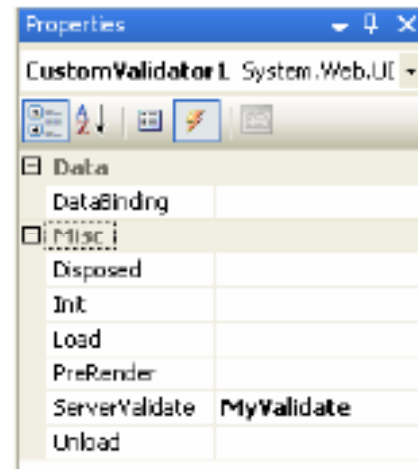
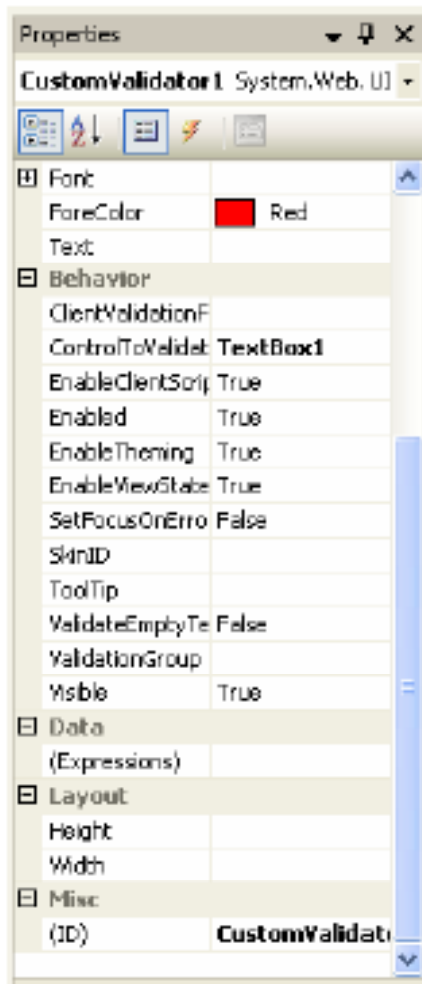
<script runat="server">
Sub user(source As object,args As
ServerValidateEventArgs)
    if len(args.Value)<8 or
len(args.Value)>16 then
        args.IsValid=false
    else
        args.IsValid=true
    end if
End Sub
</script>

```

Enter a username:

A username must be between 8 and 16 characters!

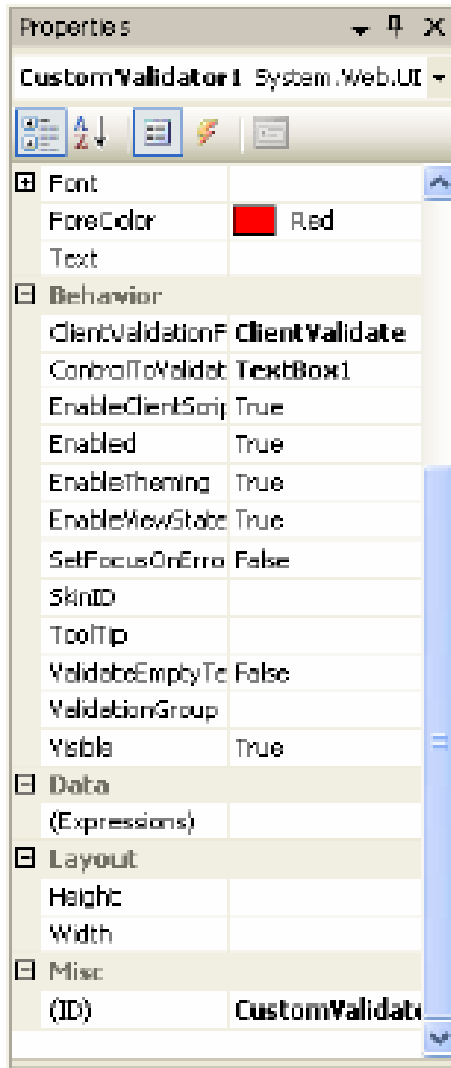
Ví dụ: Tự Kiểm tra trên Server



protected void MyValidate(object source, ServerValidateEventArgs args)

```
{  
    args.IsValid = false;  
    if (TextBox1.Text == "abc")  
        args.IsValid = true;  
}
```

Ví dụ: Tự kiểm tra trên Client



```
<script language="javascript">  
    function ClientValidate(e, args)  
    {  
        args.IsValid = false;  
        if (args.Value == "abc")  
            args.IsValid = true;  
    }  
</script>
```


2.2.3. RangeValidator Control

-Được dùng để kiểm tra giá trị nhập vào bởi user có nằm trong một khoảng dữ liệu (numbers, dates, characters) không

Enter a number from 1 to 100:

Page is not valid!!

The value must be from 1 to 100!

ControlToValidate	txtSo2
CssClass	
CultureInvariantVal	False
Display	Static
EnableClientScript	True
Enabled	True
EnableTheming	True
ableViewState	True
ErrorMessage	RangeValidator
Font	
ForeColor	 Red
Height	
MaximumValue	100
MinimumValue	10
SetFocusOnError	False
SkinID	
TabIndex	0
Text	So lon hon 10 nh
ToolTip	
Type	Integer ▼
ValidationGroup	String
Visible	Integer
Width	Double
	Date
ype	Currency

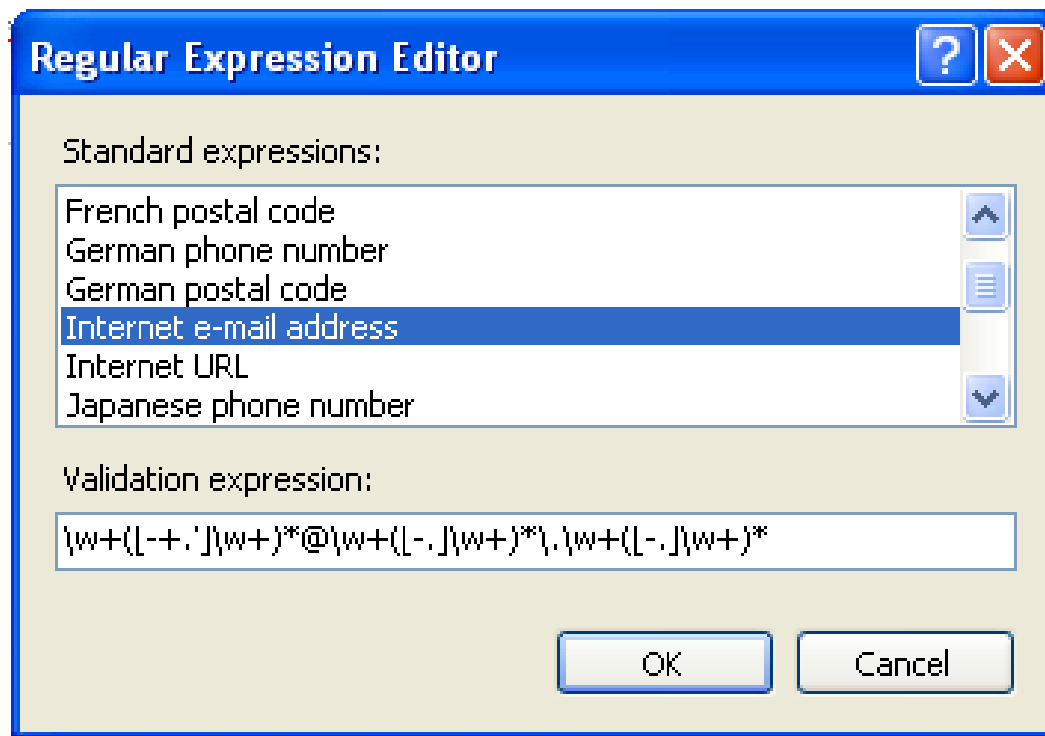
```
<script runat="server">  
Sub submit(.....)  
  If Page.IsValid Then  
    lbl1.Text="Page is valid."  
  Else  
    lbl1.Text="Page is not valid!!"  
  End If  
End Sub  
</script>
```

```
<html>  
<body>  
  
<form runat="server">  
Enter a number from 1 to 100:
```

```
<asp:TextBox id="tbox1" runat="server" />  
<br /><br />  
<asp:Button Text="Submit" OnClick="submit"  
runat="server" />  
<br /><br />  
<asp:Label id="lbl1" runat="server" />  
<br />  
<asp:RangeValidator  
  ControlToValidate="tbox1"  
  MinimumValue="1"  
  MaximumValue="100"  
  Type="Integer"  
  EnableClientScript="false"  
  Text="The value must be from 1 to 100!"  
runat="server" />  
</form>  
  
</body>  
</html>
```


2.2.4. Regular Expression Validator

Dùng để kiểm tra giá trị của control phải theo một mẫu được qui định trước trong đặc tính ValidationExpression như: địa chỉ e-mail, số điện thoại, mã vùng, số CMND,...



Enter a US zip code:

The page is not valid!

The zip code must be 5 numeric digits!

```
<script runat="server">
sub submit(.....)
  if Page.IsValid then
    lbl.Text="The page is valid!"
  else
    lbl.Text="The page is not valid!"
  end if
end sub
</script>
```

```
<html>
<body>

<form runat="server">
Enter a US zip code:
<asp:TextBox id="textbox1" runat="server" />
<br /><br />
<asp:Button text="Submit" OnClick="submit"
runat="server" />
<br /><br />
<asp:Label id="lbl" runat="server" />
<br />
<asp:RegularExpressionValidator
  ControlToValidate="textbox1"
  ValidationExpression="\d{5}"
  EnableClientScript="false"
  ErrorMessage="The zip code must be 5 numeric
digits!"
runat="server" />
</form>

</body>
</html>
```

2.2.5. Required Field Validator Control

- Dùng kiểm tra giá trị trong control phải được nhập (có giá trị khác với giá trị đầu)
- Sử dụng control này để kiểm tra ràng buộc dữ liệu khác rỗng (“”)
- Thuộc tính InitialValue: giá trị khởi đầu, mặc định là rỗng

```
<html>
<body>

<form runat="server">
Name: <asp:TextBox id="name"
runat="server" />
<br />
Age: <asp:TextBox id="age"
runat="server" />
<br /><br />
<asp:Button runat="server"
Text="Submit" />
<br /><br />
<asp:RequiredFieldValidator
ControlToValidate="name"
Text="The name field is required!"
runat="server" />
</form>

</body>
</html>
```

Name:

Age:

The name field is required!

2.2.6. Validation Summary Control

Hiển thị ra bảng lỗi tổng hợp của tất cả các control kiểm tra trên trang web

- HeaderText: Dòng tiêu đề của thông báo lỗi
- ShowMessageBox: True hiển thị như cửa sổ MessageBox, False – không hiển thị
- ShowSummary: True – hiển thị thông báo lỗi, False - không

Ví dụ Validation Control

Web Calculator

Số 1: Nhập số vào trước khi tính

Số 2: Số lớn hơn 10 nhỏ hơn 100

• Error message 1.
• Error message 2.

(Expressions)	
(ID)	RequiredFieldV
AccessKey	
BackColor	<input type="text"/>
BorderColor	<input type="text"/>
BorderStyle	NotSet
BorderWidth	
ControlToValid	txtSo1
CssClass	
Display	Static
EnableClientSc	True
Enabled	True
EnableTheming	True
EnableViewSta	True
ErrorMessage	Nhap so vao tr
Font	
ForeColor	<input type="color" value="Red"/> Red
Height	
InitialValue	
SetFocusOnErr	False
SkinID	
TabIndex	0
Text	Nhap so vao tr

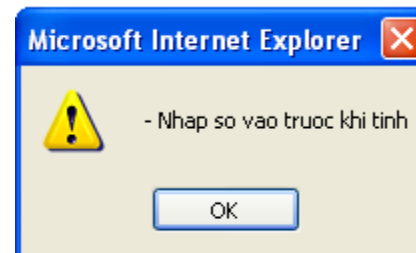
Web Calculator

Số 1: Nhập số vào trước khi tính

Số 2:

- Nhập số vào trước khi tính

Tổng:



2.3. Một số các control quan trọng khác

a) Literal Control

- Hiển thị chuỗi văn bản trên trang web như Label, nhưng khác ở chỗ không dùng thêm tag ` ...`

b) Panel và Placeholder Controls

- Chứa các điều khiển khác. Thuộc tính thường dùng là Visible. Panel cho phép kéo các control vào bên trong lúc thiết kế, còn Placeholder thì không
- Để thêm control vào trong Placeholder dùng
`ID_PlaceHolder.Controls.Add(ID_Control)`

2.3. Một số các control quan trọng khác

c) AdRotator Control

- Tạo ra các banner quảng cáo cho trang web. Tự động thay đổi các hình ảnh (đã được thiết lập trước) mỗi khi có yêu cầu,PostBack về server
- AdvertisementFile: Tên file dữ liệu (.xml) cho control, có dạng sau:

```
<Advertisements>
```

```
<Ad>
```

```
  <ImageUrl>Hinh\51.jpg</ImageUrl>
```

```
  <NavigateUrl>http://www.asp.net</NavigateUrl>
```

```
  <AlternateText>ASP.NET Site</AlternateText>
```

```
  <Keyword>Technology</Keyword>
```

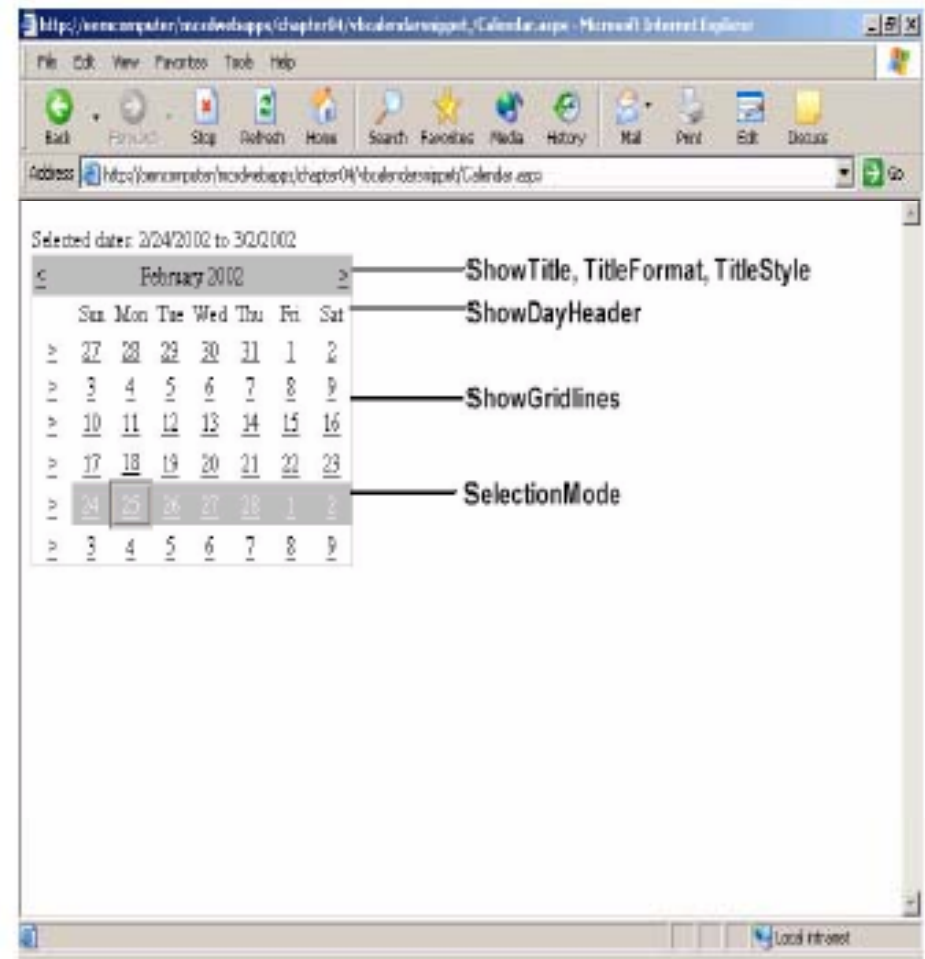
```
  <Impressions>60</Impressions>
```

```
</Ad>
```

```
</Advertisements>
```

Calendar

- Sử dụng Calendar control để lấy thông tin về Ngày tháng
- Để lấy hoặc thiết lập giá trị ngày tháng trên Calendar control, sử dụng hàm xử lý sự kiện `SelectionChanged` và thuộc tính `SelectedDate` hoặc `SelectedDates`





Chương 3: Xử lý dữ liệu với ADO.NET

- Giới thiệu về ADO.NET
- Các đối tượng trong ADO.NET
- Điều khiển liên kết dữ liệu

VnExpress - The thao - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://vnexpress.net/Vietnam/The-thao/

Nh?ng trang WEB thong hay dung nh?t VnExpress - Vietnam News Daily VnExpress - The thao

VN EXPRESS
TIN NHANH VIETNAM

Dành cho Quảng cáo
Điện thoại: 090 436 1114 (HN) / 090 810 7277 (HCM)
(Thông tin **Rao vặt** vui lòng không liên hệ theo số này)

Trang Thể thao Thứ bảy, 5/4/2008, 06:30 GMT+7

Trang nhất
Xã hội
Thể giới
Kinh doanh
Văn hóa
Thể thao
Bóng đá
Tennis
Chân dung
Pháp luật
Đời sống
Khoa học
Vi tính
Ôtô - Xe máy
Bản đọc viết
Tâm sự
Rao vặt
Cười

Search

Ngôi sao
Sổ hoá
Game thủ

Trang Thể thao

Huyền thoại Lippi lớn tiếng giải oan cho Ronaldo

 Cho rằng siêu sao của MU đã quá coi thường mình, một số cầu thủ của Roma mới dọa sẽ "chơi" lại trong trận tứ kết lượt về Champions League. Vậy nhưng, HLV của đội tuyển vô địch thế giới lại khuyên họ nên biết kiềm chế cảm xúc.

> [Sao Roma dọa 'triệt ha' Ronaldo](#)

Arsenal muốn biến Liverpool thành 'nạn nhân' Milan số hai

 Sau trận hòa Liverpool hôm giữa tuần, Arsenal giờ lâm vào tình thế tương tự như sau khi tiếp Milan ở lượt đi vòng 16 đội. Chuyển bại thành thắng một lần nữa là điều "pháo binh" Adebayor tự tin có thể làm được cùng đồng đội.

> [Arsenal vẫn tin yêu cuộc đời](#)

'Điểm yếu của Chelsea u. Đó là thiếu Mourinho'

 Sau chiến thắng 2-1 bất ngờ của Fenerbahce trước Chelsea, tiền đạo Mateja Kezman đã "hả hê" bình luận như vậy về đội bóng cũ.

AUDIO HOÀNG HẢI
Audio Hi-end
Tivi LCD, Plasma
Điều hòa - Máy giặt
Tầng 1&2 - 23D Hai Bà Trưng

Gửi Tiền HDBank
TRÚNG **VÀNG**
Nguyễn Kỳ

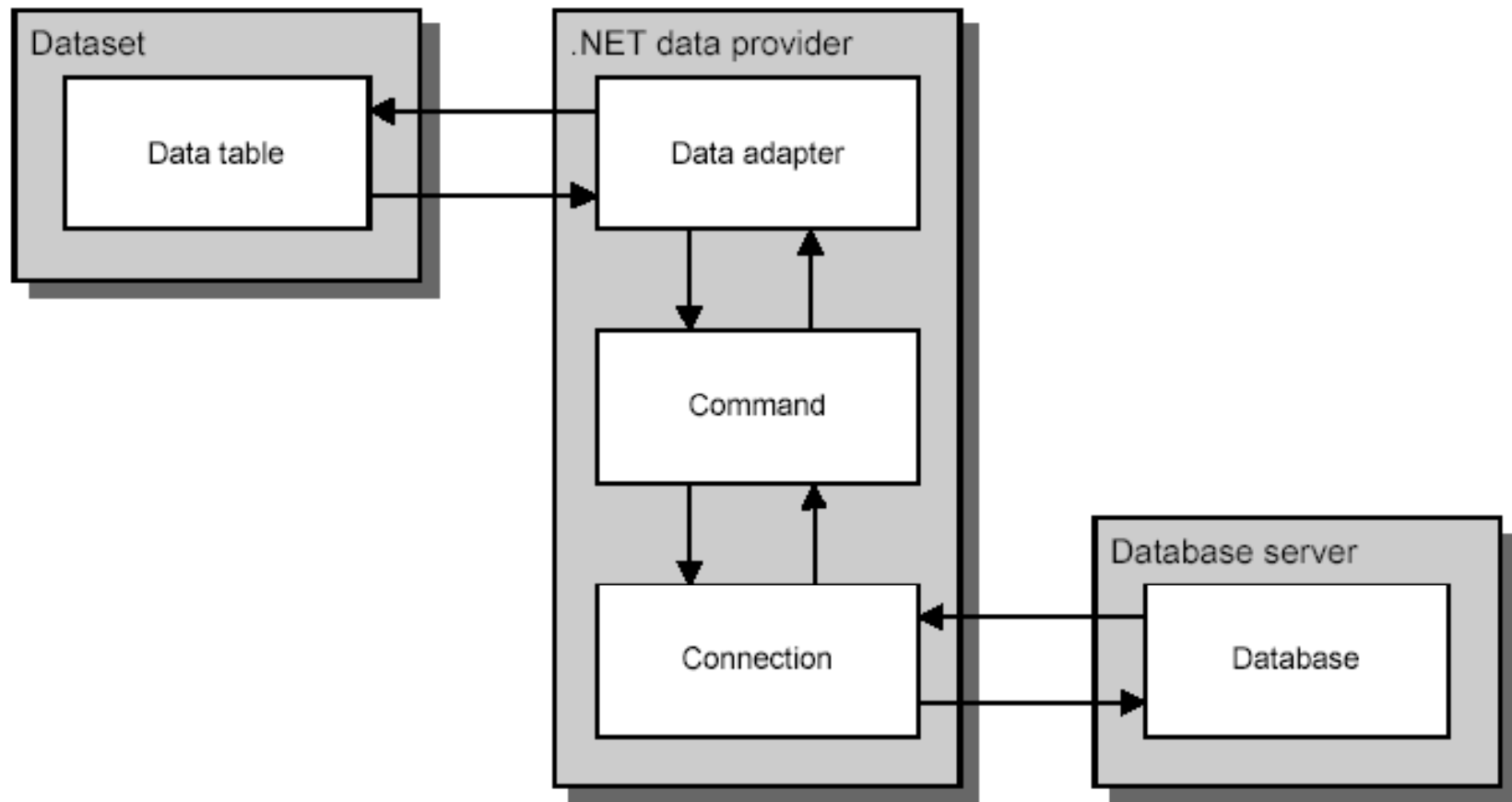
Tặng sinh lực đàn ông

3.1. Giới thiệu ADO.NET

- ADO.NET (Active Data Object) là một phần của .NET Framework, cung cấp các dịch vụ xử lý dữ liệu
- Làm việc với dữ liệu không kết nối
- Dữ liệu được lưu trữ trong một CSDL thu nhỏ gọi là DataSet =>tăng tốc độ tính toán, giảm sử dụng tài nguyên trên Database server.
- Khả năng xử lý dữ liệu dạng chuẩn XML =>làm việc với nhiều ứng dụng khác

3.1. Giới thiệu ADO.NET

Basic ADO.NET objects

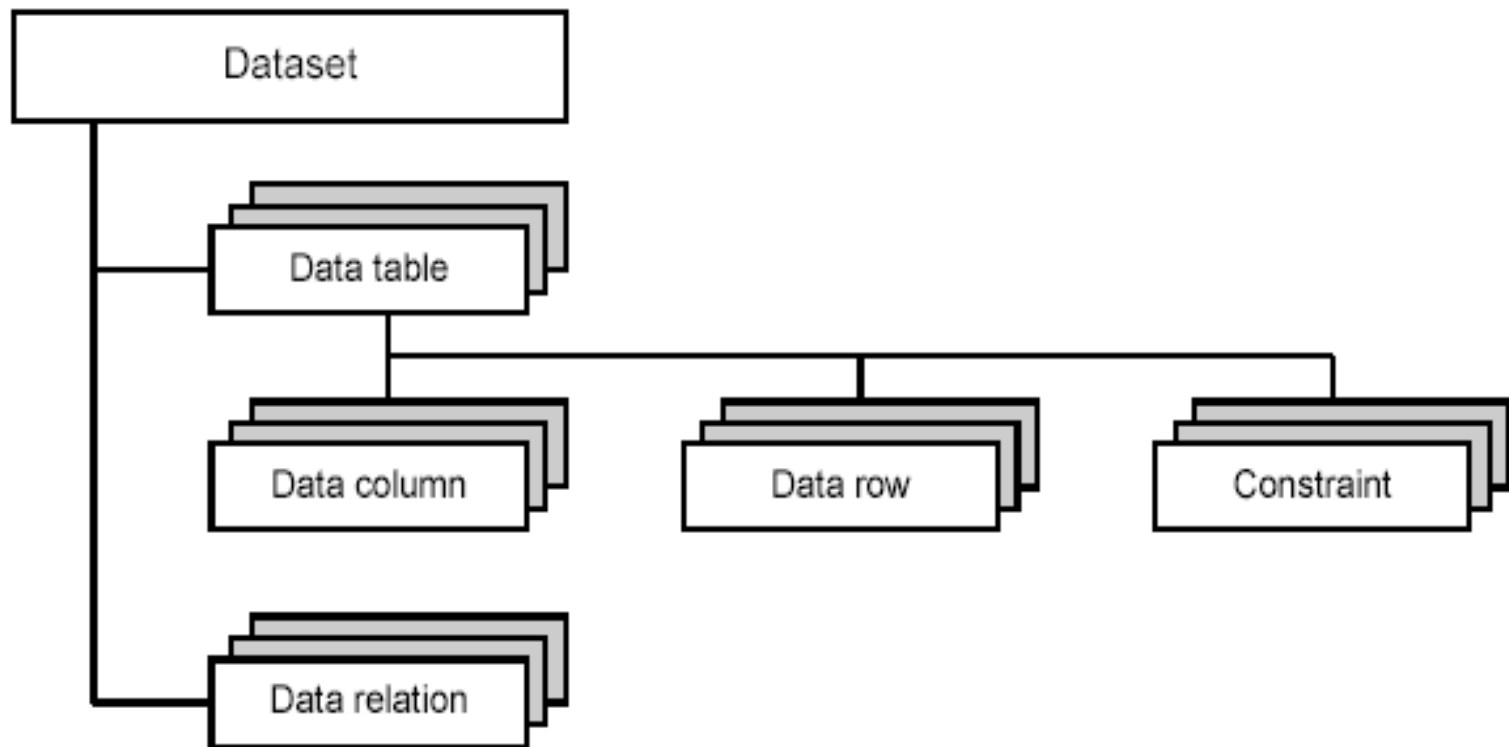


3.1. Giới thiệu ADO.NET

- Kiến trúc ADO.NET có hai đối tượng chính: Dataset (data table) và .NET data provider (Data Adapter, DataReader, Command, Connection)
- Dataset lưu data từ Database
- Để nhận data từ CSDL, DataAdapter dùng câu lệnh SELECT trong Command, để cập nhật data dùng INSERT, UPDATE, DELETE

Các đối tượng con của DataSet

The basic dataset object hierarchy

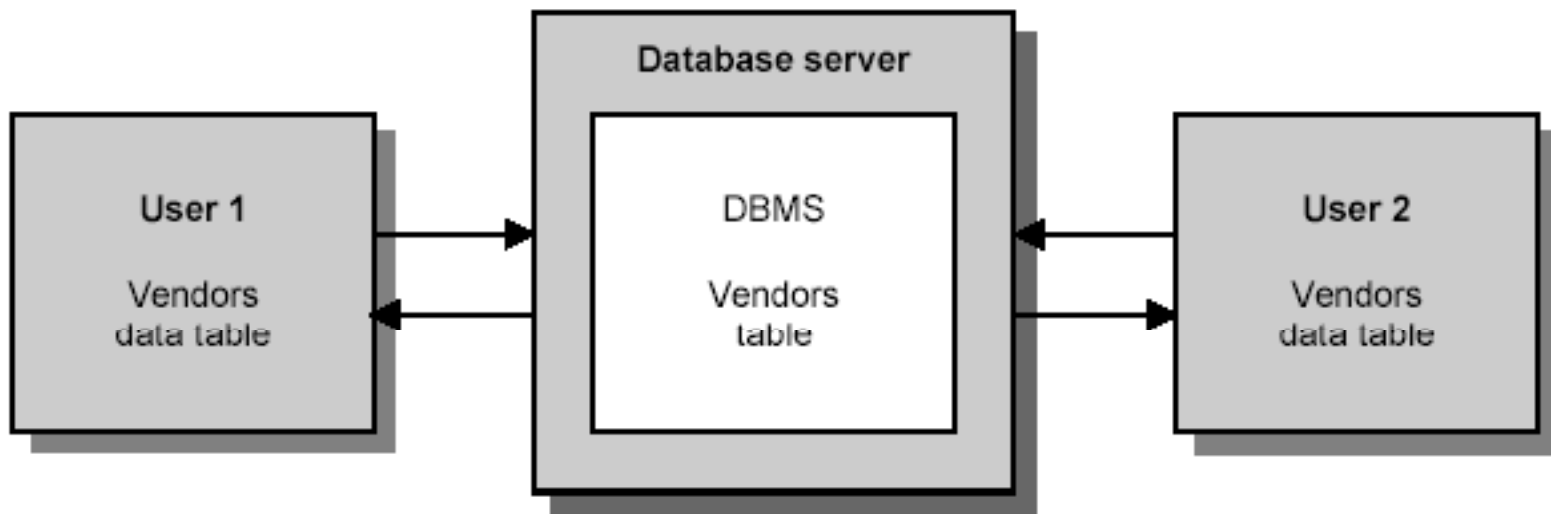


Các lợi ích Dataset

- Hoàn toàn trong bộ nhớ: Một Table trong Dataset là một Array of Rows, nên ta có thể dùng thẳng (direct access) một record bằng cách nói đến cái Row chứa nó, chứ không cần phải dùng MoveNext, MovePrev,..v.v.
- Làm nhẹ công tác của Database server. Tất cả mọi công tác sửa đổi dữ liệu đều được thực hiện trong Dataset.
- Dataset có thể được biểu diễn bằng một XML (eXtensible Marked Language)

Sự cạnh tranh và cấu trúc dữ liệu ngắt (Concurrency and the disconnected data architecture)

Two users who are working with copies of the same data



- Cách xử lý - có 2 cách: Optimistic concurrency (default) và “Last in wins”
- Làm thế nào để tránh bị trường hợp concurrency?

Các điểm nổi bật của ADO.NET

- Tương tác giữa nhiều hệ thống khác nhau (Interoperability)
- Hỗ trợ nhiều người dùng (Scalability)
- Mở rộng khả năng làm việc với CSDL (Productivity)
- Hiệu quả cao trong xử lý dữ liệu (Performance)

3.2. Các đối tượng trong ADO.NET

6.2.1. Khai báo và sử dụng không gian tên (namespace)

```
<%@ Import Namespace="System.Data.OleDb" %>
```

Namespace	Mục đích
-System.Data	-Các đối tượng và kiểu cơ bản phục vụ ADO.NET
-System.Data.OleDb	-Các lớp được thiết kế để làm việc với bất kỳ nguồn dữ liệu nào
-System.Data.SqlClient	- Thiết kế tối ưu cho CSDL SQL server

3.2.2. Đối tượng Connection

a) Data Provider

- Data Provider có sẵn trong ADO.NET

`System.Data.OleDb`

`System.Data.SqlClient`

- Ứng với mỗi tên miền có một connection tương ứng

`System.Data.OleDb.OleDbConnection`

`System.Data.SqlClient.SqlConnection`

b) Connection String

-Cung cấp thông tin cần thiết cho kết nối

-Phụ thuộc vào Data Provide

➤ **OLEDB Provider** có thể gồm các thành phần:

+ Provider

+ Data Source

+ User ID, Password

Ví dụ 1: Tạo connection đến CSDL Microsoft Access

Dim ConnString As String

**ConnString = "Provider=Microsoft.Jet.OLEDB.4.0 ;data source=" & _
Server.MapPath("./dbase/DBHuan.mdb")**

Dim ConnDB As New OleDbConnection(ConnString)

b) Connection String - OLEDB Provider

Ví dụ 2: Tạo Connection đến CSDL SQL Server

Dim ConnString As String

‘ Chuỗi kết nối

```
ConnString = "Provider=SQLOLEDB ;data source=" & _  
            <địa chỉ TCP/IP hay tên máy> & _  
            ";Initial Catalog =" & <tên database> & _  
            ";User ID =" & <tên người dùng> & _  
            ";Password =" & <mật khẩu>
```

‘ Tạo kết nối

Dim ConnDB As New OleDbConnection(ConnString)

b) Connection String - SQLClient Provider

Data Source

-Initial Catalog

-Integrated security: True (SSPI)/ False chỉ định dùng cơ chế bảo mật của Windows Login (True) hay cơ chế bảo mật của SQL Server (False)

-User ID

-Password

```
Data Source=.\SQLEXPRESS;  
AttachDbFilename=|DataDirectory|\khachhang.mdf;  
Integrated Security=True;User Instance=True
```

b) Connection String - SQLClient Provider

VD 3: Tạo Connection đến CSDL SQL Server

Dim ConnString As String

‘ Chuỗi kết nối

```
ConnString = "Data Source=" & _  
    <địa chỉ TCP/IP hay tên máy> & _  
    ";Initial Catalog =" & <tên database> & _  
    ";User ID =" & <tên người dùng> & _  
    ";Password =" & <mật khẩu>
```

‘ Tạo kết nối

Dim ConnDB As New SqlConnection(ConnString)

b) Connection String - SQLClient Provider

VD 4: Tạo Connection đến CSDL SQL Server

Dim ConnString As String

‘ Chuỗi kết nối

ConnString = “Data Source=” & _
<địa chỉ TCP/IP hay tên máy> & _
“;Initial Catalog =” & <tên database> & _
“;Integrated Security=SSPI”

‘ Tạo kết nối

Dim ConnDB As New SqlConnection(ConnString)

c) Các phương thức trên đối tượng Connection

-Open: Thực hiện kết nối với các thông tin đã khai báo trong Connection String

-Dispose: Xóa mọi tài nguyên liên quan đến Connection trên vùng nhớ

-Close: Đóng kết nối với nguồn dữ liệu

VD: ConnDB.Open(), ConnDB.Close()

ConnDB.Dispose()

Một số đặc tính quan trọng khác của OleDbConnection

Đặc tính	Dạng data	ý nghĩa
ConnectionTimeout	Int32	khoảng thời gian cố gắng tạo liên kết VD: ConnString="Provider....; Connect Timeout=11 "
Database	String	Trả về tên CSDL đã kết nối
ServerVersion	String	Trả về version của nguồn CSDL
State	StateConnection (0,1,2,4,8,16)	Trả về trạng thái hiện tại của đối tượng: actived hay hủy rồi

3.2.3. Đối tượng Command

-Khai báo và sử dụng đối tượng Command:

+ đối với SqlConnection:

```
Dim MyCom As SqlCommand
```

```
MyCom = New SqlCommand(ConnString, ConnDB)
```

+ đối với OleDbConnection

```
Dim MyCom As OleDbCommand
```

```
MyCom = New OleDbCommand(ConnString, ConnDB)
```

a) Các thuộc tính của Command

- CommandText**: Tên bảng, lệnh SQL hoặc Store Procedure
- CommandType**: Tương ứng với giá trị của CommandText
 - + Text: câu lệnh SQL (default)
 - + Storeprocedure: thủ tục nội
 - + TableDirect: tên bảng
- Connection**: Đối tượng Connection sử dụng cho Command
- Parameters**: Tập hợp các tham số dùng trong Command

b) Các phương thức thực hiện Command

➤ **ExecuteNonQuery**: Dùng để gọi thực hiện các câu truy vấn cập nhật INSERT, UPDATE, DELETE mà không quan tâm đến dữ liệu trả về.

VD: Xóa trường “AuthorName” có giá trị “Joe Smith” trong bảng “Authors” của CSDL “books.mdb”

```
Dim ConnString, SqlString As String
```

```
Dim ConnDB, MyCom
```

```
ConnString = "...server.MapPath(./books.mdb)
```

```
SqlString = "DELETE from Authors where AuthorName='Joe Smith'"
```

```
ConnDB = New OleDbConnection(ConnString)
```

```
ConnDB.Open()
```

```
MyCom = New OleDbCommand(SqlString,ConnDB)
```

```
MyCom.ExecuteNonQuery()
```

```
ConnDB.Close()
```


b) Các phương thức thực hiện Command

➤ **ExecuteReader**: Trả về một đối tượng DataReader để đọc dữ liệu mỗi lần một dòng với method Read(). DataReader đọc dữ liệu trực tiếp từ Database => duy trì kết nối cho đến khi đọc xong

Gọi: **MyCom.ExecuteReader()**

➤ **ExecuteScalar**: Trả về giá trị của cột đầu tiên và dòng đầu tiên

Gọi: **MyCom.ExecuteScalar()**

c) Parameter

Xét trường hợp sau:

```
SELECT CustomerID,CompanyName,Phone From Customers  
WHERE CustomerID = "FPT"
```

```
SELECT CustomerID,CompanyName,Phone From Customers  
WHERE CustomerID = ?
```

VD:

```
cmd.CommandText = "SELECT...WHERE CustomerID = ?"
```

```
Dim Par As OleDbParameter = cmd.CreateParameter()
```

```
Par.Value = "FPT"
```

```
cmd.Parameters.Add(Par)
```

c) Parameter

Các thuộc tính thường dùng nhất của OleDbParameter

-DataType: chỉ dạng dữ liệu

-Direction: Loại tham số với các giá trị

+ Input(mặc định) tham số đầu vào

+ Output: tham số đầu ra

+ InputOutput: tham số vào và ra

+ ReturnValue: loại tham số nhận giá trị trả về của một thủ tục nội hay một hàm

-OleDbType, SqlDbType: kiểu dữ liệu OleDb hoặc SqlDbType

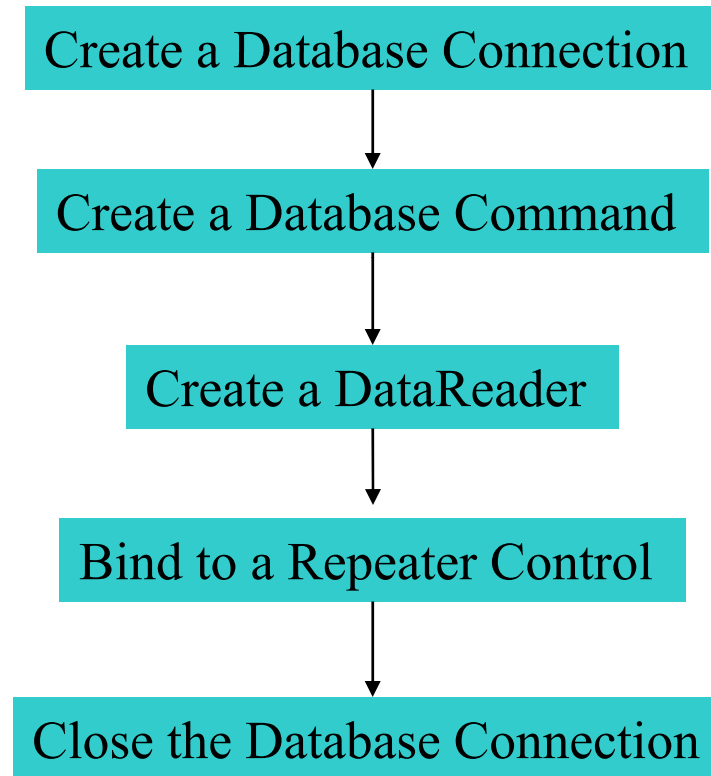
-Value: Giá trị của tham số có kiểu là object (String, Integer)

-ParameterName: Tên tham số

SELECT OrderID, CustomerID FROM Orders WHERE CustomerID =
@CustomerID

ParameterName có tên là @CustomerID.
ASP.NET

3.3. Quy trình truy xuất một CSDL đơn giản trong ASP.NET



Create a Database Connection

```
<%@ Page Language="VB" %>
<%@ Import Namespace="System.Data.OleDb" %>
<script runat="server">
    Sub Page_Load()
        Dim ConnString As String
        Dim dbconn, sql, dbcomm, dbread
        ConnString = "Provider=Microsoft.Jet.OLEDB.4.0 ;data source=" & _
            Server.MapPath("./dbase/DBHuan.mdb")
        dbconn = New OleDbConnection(ConnString)
        dbconn.Open()
    End Sub
</script>
```

Create a Database Command

```
<script runat="server">  
    Sub Page_Load()  
        Dim ConnString As String  
        Dim dbconn, sql, dbcomm, dbread  
        ConnString = "Provider=Microsoft.Jet.OLEDB.4.0 ;data source=" & _  
            Server.MapPath("./dbase/DBHuan.mdb")  
        dbconn = New OleDbConnection(ConnString)  
        dbconn.Open()  
        sql = "SELECT * FROM CatPlaces"  
        dbcomm = New OleDbCommand(sql, dbconn)  
    End Sub  
</script>
```

Create a DataReader

```
<script runat="server">  
  Sub Page_Load()  
    Dim ConnString As String  
    Dim dbconn, sql, dbcomm, dbread  
    ConnString = "Provider=Microsoft.Jet.OLEDB.4.0 ;data source=" & _  
                Server.MapPath("./dbase/DBHuan.mdb")  
    dbconn = New OleDbConnection(ConnString)  
    dbconn.Open()  
    sql = "SELECT * FROM CatPlaces"  
    dbcomm = New OleDbCommand(sql, dbconn)  
    dbread = dbcomm.ExecuteReader()  
  End Sub  
</script>
```

Bind to a Repeater Control and Close the Database Connection

.....

```
dbread = dbcomm.ExecuteReader()
```

```
repCatPlaces.DataSource = dbread
```

```
repCatPlaces.DataBind()
```

```
dbread.Close()
```

```
dbconn.Close()
```

```
End Sub
```

```
</script>
```


Repeater Control

```
<form id="Form1" runat="server">
<asp:Repeater id="repCatPlaces" runat="server">
<HeaderTemplate>
  <table border="1" width="100%">
  <tr>
  <th>Ma so Vung</th>
  <th>Ten Vung</th>
  <th>Ten vi tri</th>
  <th>Nam chup</th>
  <th>Ghi chu</th>
  </tr>
</HeaderTemplate>
```

```
<ItemTemplate>
  <tr>
  <td><%#Container.DataItem("IDPlace")%></td>
  <td><%#Container.DataItem("NameZona")%></td>
  <td><%#Container.DataItem("PlaceName")%></td>
  <td><%#Container.DataItem("YearOfPhoto")%></td>
  <td><%#Container.DataItem("Note")%></td>
  </tr>
</ItemTemplate>
<FooterTemplate>
  </table>
</FooterTemplate>
</asp:Repeater>
</form>
```

3.4. Tạo liên kết với CSDL tự động

-Demo tạo kết nối và hiển thị

3.5. DataSet

-DataSet gắn với Recordset của ADO

-Là một mô hình CSDL thu nhỏ

Các chức năng chính của DataSet:

-Làm việc với dữ liệu không liên kết

-Khả năng dịch chuyển, sắp xếp, tìm và lọc dữ liệu

-Cache những thay đổi

-Làm việc với XML-document

Các thuộc tính của DataSet:

-Relations: tập hợp các quan hệ DataRelation DataSet gắn với Recordset của ADO

- Tables: Tập hợp các bảng của DataSet (*DataColumn*, *DataRow*, *Constraint*)

Tạo đối tượng DataSet

```
Dim ds As New DataSet("DataSetName")
```

```
Dim strConn, strSQL As String
```

```
strConn = "Provider=SQLOLEDB;Data Source=(local)\NetSDK;" & _
```

```
"Initial Catalog=Northwind;Trusted_Connection=Yes;"
```

```
strSQL = "SELECT CustomerID, CompanyName, ContactName,  
Phone " & _
```

```
"FROM Customers"
```

```
Dim da As New OleDbDataAdapter(strSQL, strConn)
```

```
Dim ds As New DataSetQ
```

```
da.Fill(ds, "Customers")
```

Các phương thức của DataSet

-Thêm bảng vào DataSet

Tables.Add(<tên bảng>)

-Xóa bảng ra khỏi tập hợp Tables của DataSet

Tables.Remove(<bảng>)

- Xóa bảng có chỉ số là <chỉ số> ra khỏi tập hợp Tables

Tables.RemoveAt(<chỉ số>)

-Xóa tất cả các bảng ra khỏi tập hợp Tables

Tables.Clear()

-Kiểm tra bảng có thuộc về DataSet không

Tables.Contains(<tên bảng>)

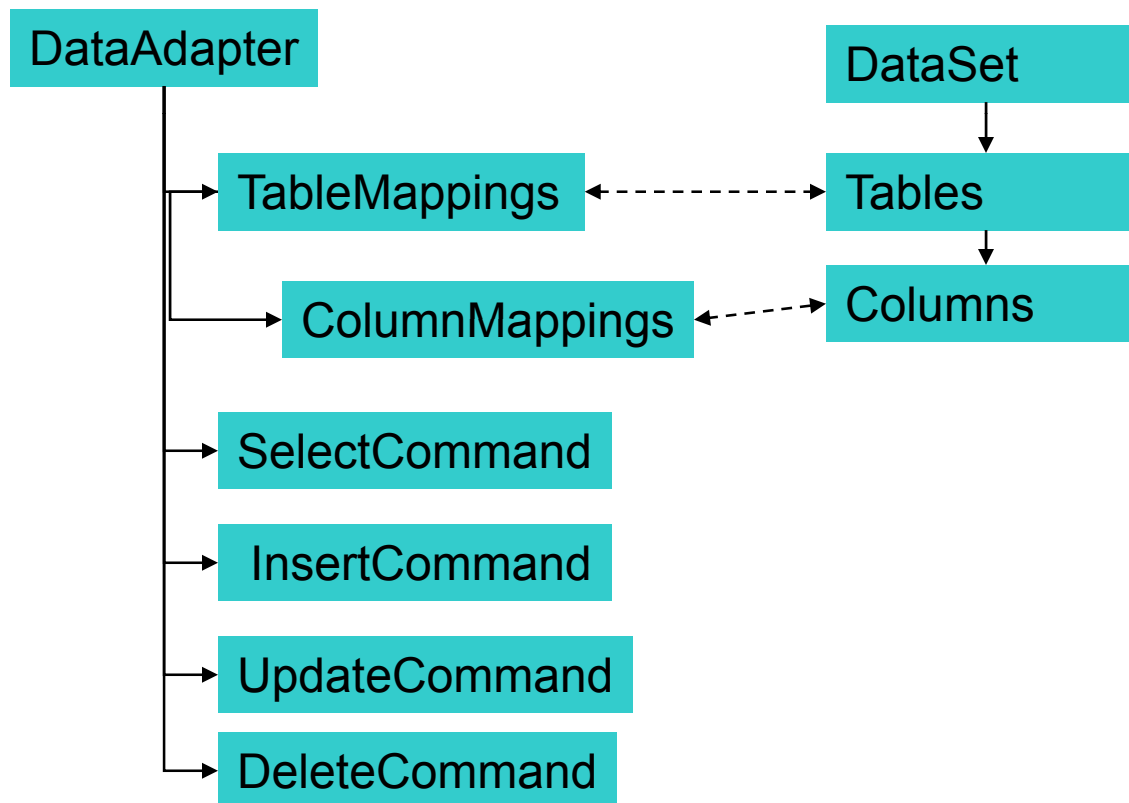
-Xóa bỏ mọi dữ liệu trên DataSet

Clear()

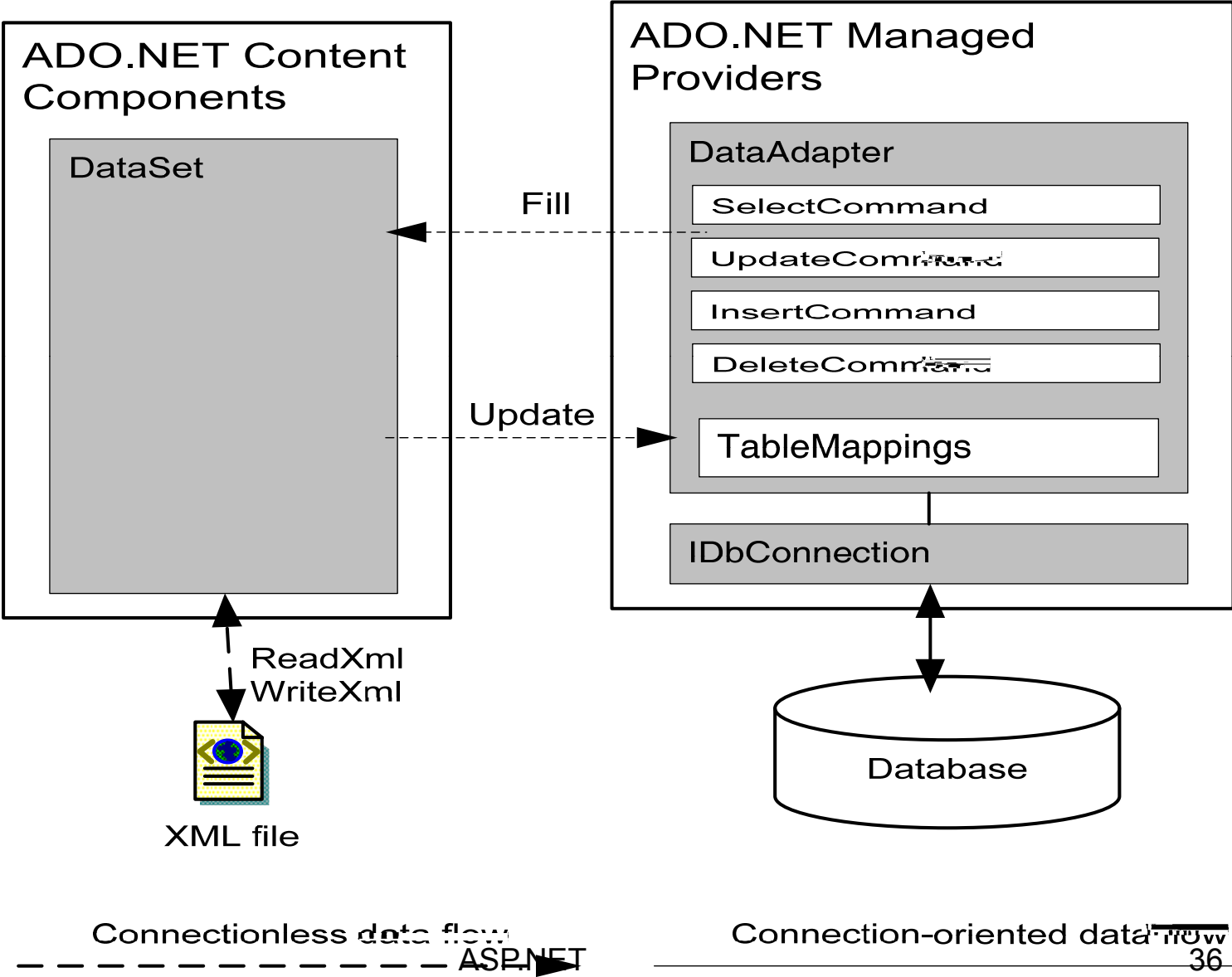
-Xóa bỏ DataSet : Dispose()

3.6. DataAdapter

-Lấy cấu trúc và dữ liệu của các bảng trong nguồn dữ liệu về cho DataSet và cập nhật các thay đổi về nguồn dữ liệu gốc

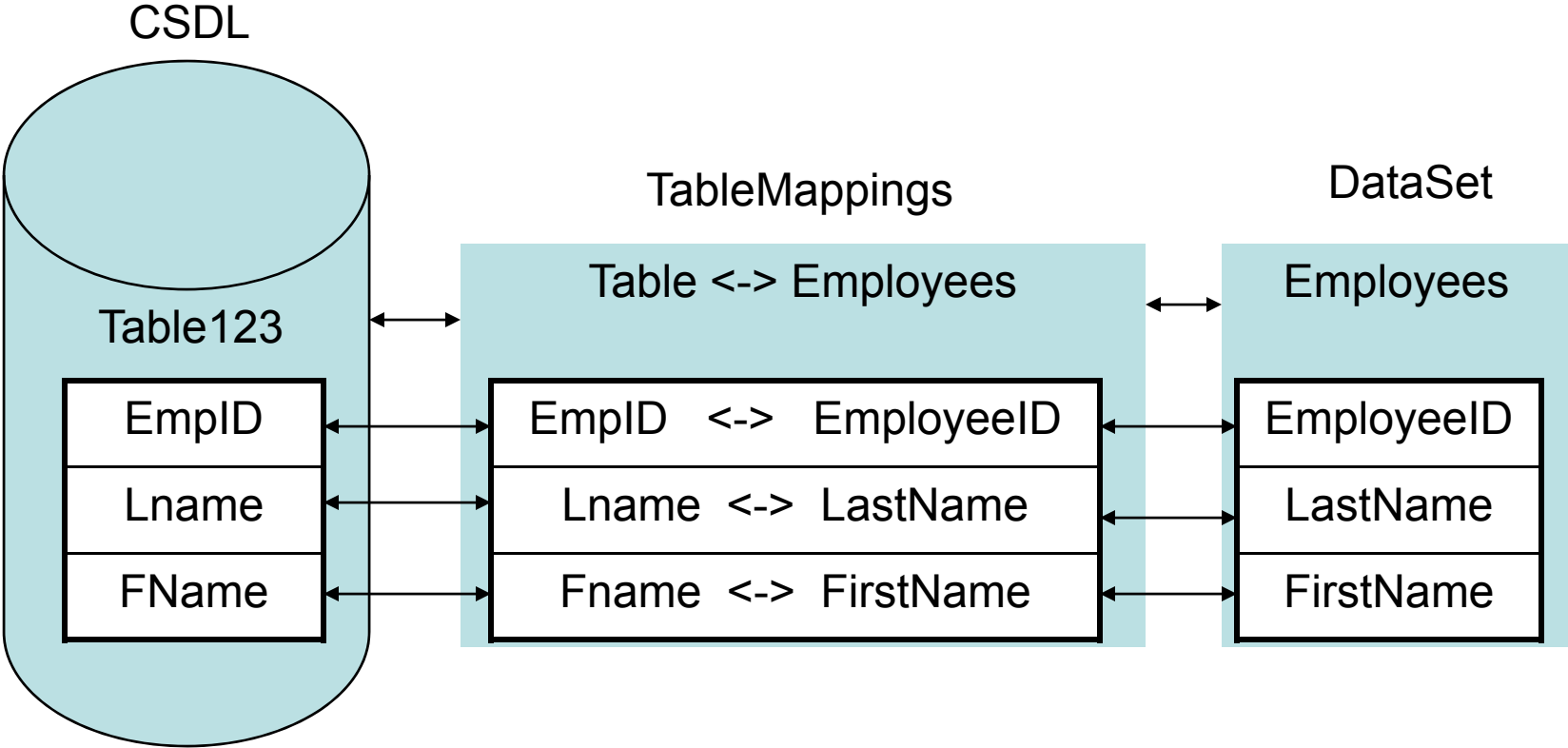


Architecture



TableMappings

```
SELECT EmpID AS EmployeeID, LName AS LastName,  
FName AS FirstName FROM Employees
```



TableMappings

Dim da As OleDbDataAdapter

‘khởi tạo SqlDataAdapter

Dim TblMap As DataTableMapping

Dim ColMap As DataColumnMapping

TblMap = da.TableMappings.Add("Table", "Employees")

ColMap = TblMap.ColumnMappings.Add("EmpID", "EmployeeID")

ColMap = TblMap.ColumnMappings.Add("LName", "LastName")

ColMap = TblMap.ColumnMappings.Add("FName", "FirstName")

Tạo DataAdapter

OleDbDataAdapter trong System.Data.OleDb.OleDbDataAdapter

SqlDataAdapter trong System.Data.OleDb.SqlDataAdapter

➤ Khai báo DataAdapter từ đối tượng Connection

Dim da As New OleDbDataAdapter(SqlString,dbconn)

➤ Khai báo DataAdapter từ chuỗi Connection

Dim da As New OleDbDataAdapter(SqlString,ConnectionString)

VD:

```
Dim ConnString = "Provider=Microsoft.Jet.OLEDB.4.0 ;data source=" & _  
                Server.MapPath("./dbase/DBHuan.mdb")  
dbconn = New OleDbConnection(ConnString)  
dbconn.Open()  
SqlString = "SELECT * FROM CatPlaces"
```

Dim da As New OleDbDataAdapter(SqlString,dbconn)

‘ hoặc Dim da As New OleDbDataAdapter(SqlString,ConnectionString)

Các chức năng của DataAdapter

a) Lấy dữ liệu từ nguồn về cho DataSet, DataTable qua method Fill

-Đổ dữ liệu vào DataSet có sẵn dưới dạng các bảng với tên mặc định là Table, Table1, Table2,...

(TênĐối tượngDataAdapter).Fill(<DataSet>)

-Đổ dữ liệu vào DataTable có sẵn

(TênĐối tượngDataAdapter).Fill(<DataTable>)

-Đổ dữ liệu vào DataSet cho bảng <tên dataTable>, nếu chưa có, bảng sẽ được tạo. Trả về số mẫu tin lấy về được

(TênĐối tượngDataAdapter).Fill(<DataSet>, <tênDataTable>)

-Chia thành từng trang

da.Fill(DataSet, intStartRecord, intNumRecords, "TênBảng")

da.Fill(DataSet, 0, 20, "Products")

Các chức năng của DataAdapter

b) Lấy cấu trúc dữ liệu từ nguồn

-Đổ cấu trúc dữ liệu vào DataSet có sẵn và trả về một tập hợp các bảng được thêm vào DataSet

da.FillSchema(<DataSet>, <kiểu cấu trúc>)

-Đổ cấu trúc dữ liệu vào DataTable có sẵn và trả về DataTable

da.FillSchema(<DataTable>, <kiểu cấu trúc>)

-Đổ cấu trúc dữ liệu vào DataSet cho bảng <TênDataTable>

da.FillSchema(<DataSet>, <tênDataTable>, <kiểu cấu trúc>)

-Phương thức này có tham số bắt buộc là SchemaType có 2 giá trị Mapped và Source xác định có sử dụng các TableMappings cho các cấu trúc đưa vào DataSet hay không.

Các chức năng của DataAdapter

c) Tạo bộ lệnh cập nhật cho DataAdapter

Dựa vào nội dung lệnh SelectCommand dùng đối tượng CommandBuilder để tự động tạo các lệnh còn lại.

System.Data.OleDb.OleDbCommandBuilder

System.Data.OleDb.SqlCommandBuilder

Dim Update As New <loạiCommandBuilder> (ĐốiTượngDataAdapter)

d) Cập nhật dữ liệu về nguồn

-Update(<dataset>)

-Update(<DataTable>)

-Update(<DataSet>, <tên bảng>)

3.7. DataTable

Thuộc tên miền System.Data.DataTable

Cách khai báo:

- + Dim BangX As New DataTable()
- + Dim BangX As New DataTable(<tên bảng>)

Các thuộc tính chính:

ChildRelations	Trả về tập hợp các quan hệ, trong đó bảng đóng vai trò bảng cha
Columns	Trả về các cột trong bảng (thuộc DataColumn)
Constraints	Trả về các ràng buộc trong bảng
DataSet	Trả về DataSet chứa bảng
DefaultView	Trả về DataView phát sinh từ bảng
ParentRelations	Tập hợp các quan hệ, trong đó bảng đóng vai trò bảng con
PrimaryKey	Mảng các cột chức năng làm khóa chính của bảng
Rows	Trả về tập hợp các dòng dữ liệu của bảng
TableName	Tên DataTable

Tập hợp Rows

Rows là tập hợp các dòng dữ liệu của bảng vì vậy mọi tham chiếu đến dòng đều thông qua tập hợp này

Các chức năng chính của Rows:

-Rows.Count

-Rows.Item(<chỉ số dòng>) hoặc Rows(<chỉ số dòng>)

-Rows.Item(<chỉ số dòng>, <chỉ số cột>) hoặc

Rows(<chỉ số dòng>, <chỉ số cột>) hoặc

Rows(<chỉ số dòng>, <tên cột>)

- Thêm dòng vào bảng

+ Dim NewRow As DataRow

+ NewRow = <DataTable>.NewRow()

+ <DataTable>.Rows.Add(<dòng>)

-Rows.Add(<mảng trị>)

-Rows.Remove (<dòng>), Rows.RemoveAt(<chỉ số>)

-Rows.Clear()

Tập hợp Columns

Tương tự như Rows, Columns là tập hợp các cột dữ liệu của bảng vì vậy mọi tham chiếu đến cột đều thông qua tập hợp này

- Columns.Count
- Columns.Item(<tên cột>)
- Columns.Item(<chỉ số cột>)
- Columns(<tên cột>)
- Columns(<chỉ số cột>)

3.8. DataView

Cung cấp các phương pháp thuận tiện cho việc liên kết dữ liệu với các điều khiển, sắp xếp,...

-Tạo DataView

+ New DataView()

+ New DataView(<bảng>)

+ New DataView(<bảng>, <biểu thức lọc>, <biểu thức sắp xếp>, <trạng thái dòng>)

VD:

```
Dim tbl As New DataTable("bảng")
```

```
Dim vue As DataView
```

```
vue = New DataView()
```

```
vue.Table = tbl
```

```
hoặc vue = New DataView(tbl)
```

Các thuộc tính của DataView

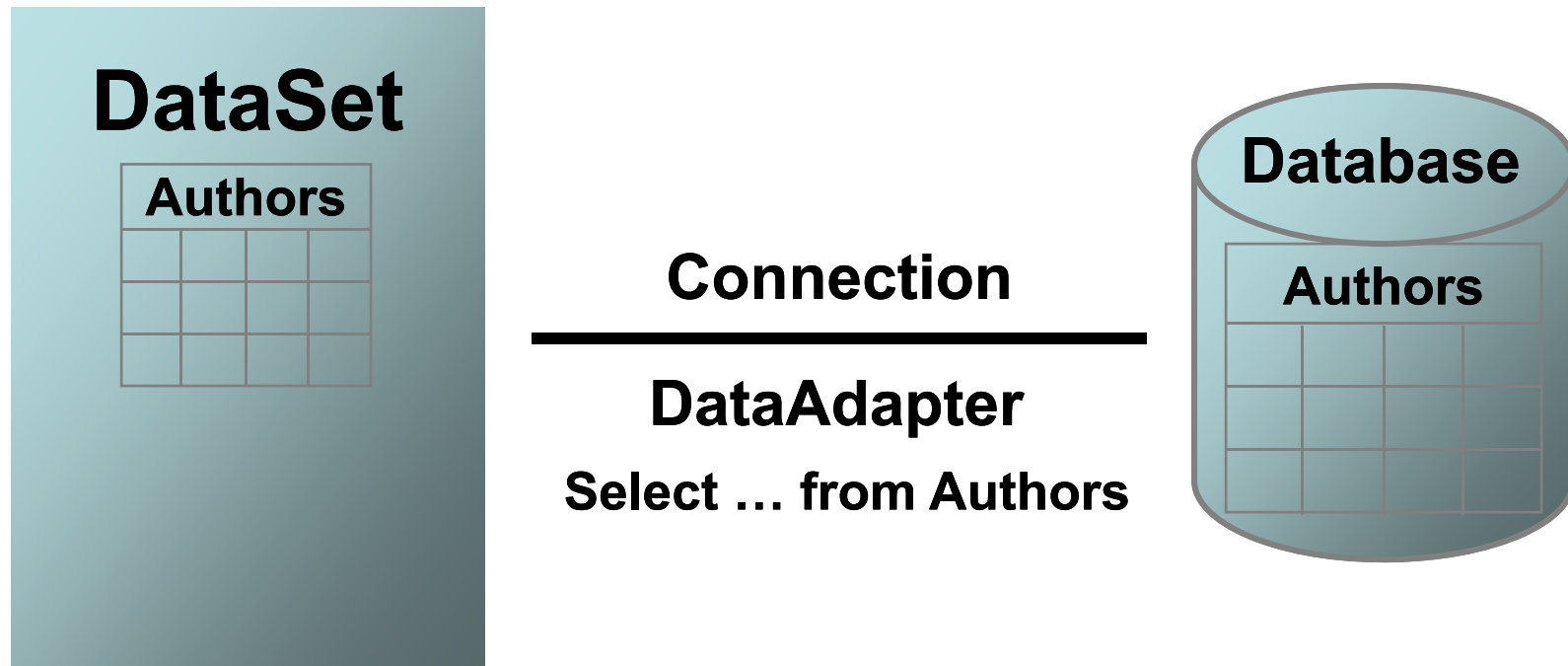
AllowDelete	cho phép thao tác xóa dòng trên DataView
AllowEdit	Cho phép thao tác sửa đổi trên DataView
AllowNew	Cho phép thao tác thêm mới bằng AddNew
Count	Số mẫu tin trên DataView
Item	Trả về dòng dữ liệu trên bảng theo tham số truyền vào
RowFilter	Biểu thức lọc
RowStateFilter	Trạng thái dòng dữ liệu trên DataView + Added: dòng mới thêm nhưng chưa cập nhật + CurrentRows: tất cả các dòng không thay đổi, mới và đã thay đổi + Deleted: dòng đánh dấu hủy + ModifiedCurrent + ModifiedOriginal + None, OriginalRows, Unchanged
Sort	sắp xếp dữ liệu trên DataView
Table	Tên bảng

Chương 4: Điều khiển liên kết dữ liệu

- **DataGrid (1.1, VS 2003)**
- **GridView**
- **DataList**
- **DetailsView**
- **Form View**
- **Repeater**
- **Các ví dụ ứng dụng**

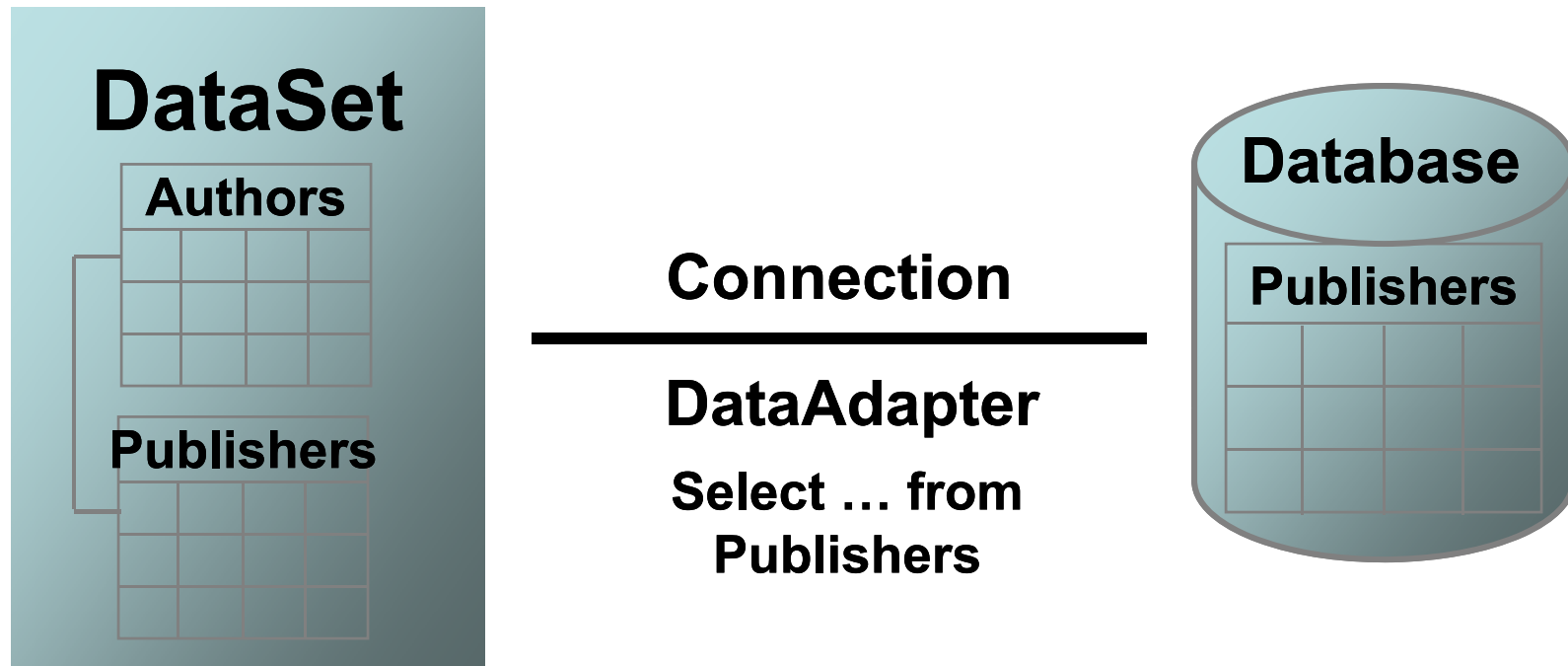
Data Controls

ADO.NET



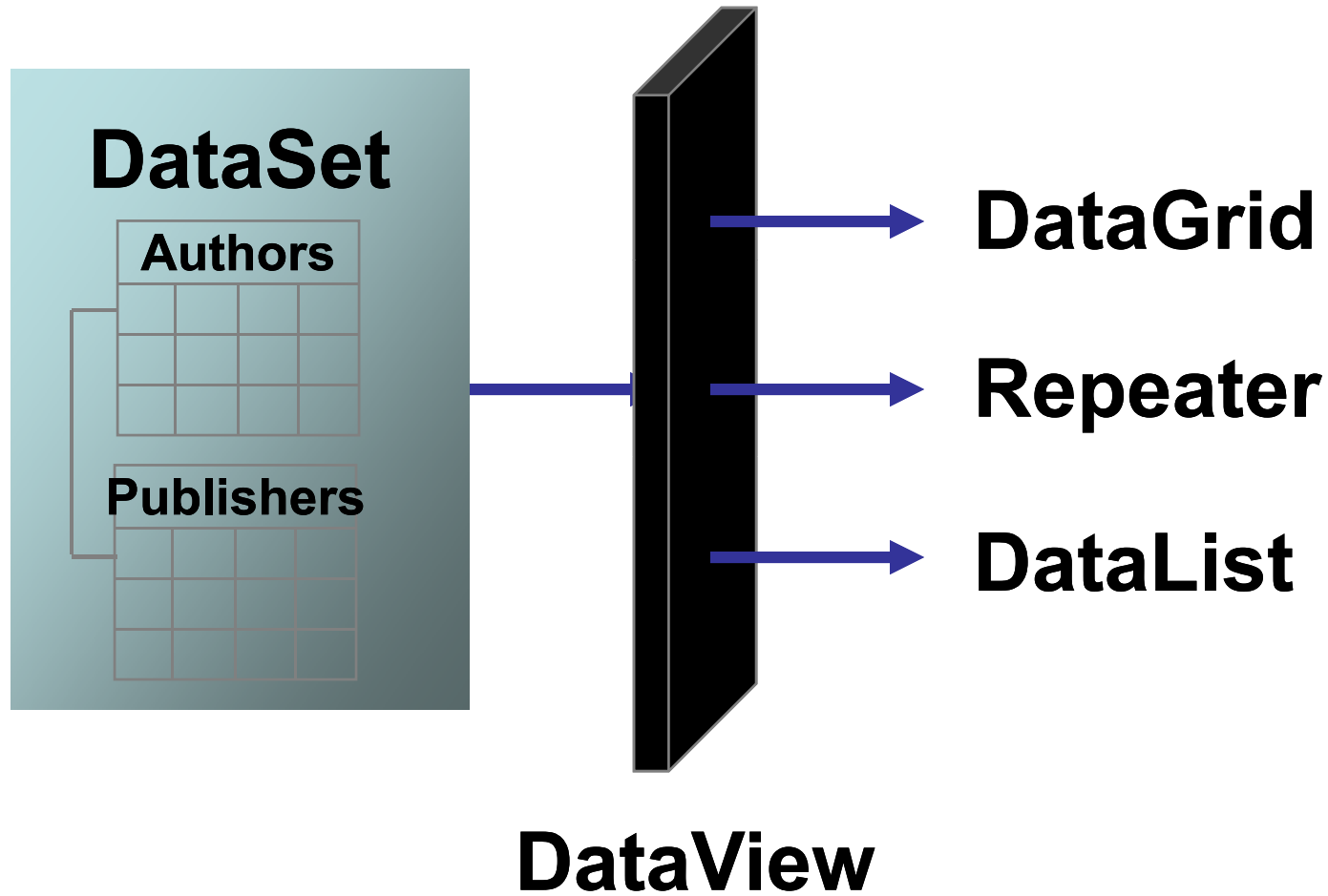
Data Controls

ADO.NET



Data Controls

ADO.NET



ASP.NET

4.1. Điều khiển DataGrid

Các lựa chọn trong khi thao tác định dạng lưới trong VS.NET

- Trang General
- Trang Columns
- Trang Paging
- Trang Format
- Trang Borders

4.1. Điều khiển DataGrid

➤ Xử lý phân trang - **Paging**

➤ Tùy biến các cột – *Column Template*

- sự kiện ItemDataBound của DataGrid

Gọi DataBind-> sự kiện *ItemDataBound* cho mỗi dòng dữ liệu

➤ Cập nhật dữ liệu trực tiếp trên lưới

- sự kiện ItemCommand của DataGrid

Command name = “Edit” => sự kiện EditCommand

“Update” – UpdateCommand

“Cancel” – CancelCommand

“Delete” – DeleteCommand

Select command =>ItemCommand

Lệnh cập nhật dữ liệu Update

Cách 1: Dùng đối tượng Command để cập nhật dữ liệu

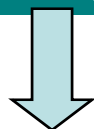
1. Lấy dữ liệu mà user vừa cập nhật vào



2. Tạo kết nối với CSDL



3. Tạo đối tượng Command để cập nhật
dữ liệu



4. Truyền tham số cho đối tượng
Command



5. Thi hành lệnh

ASP.NET

Lệnh cập nhật dữ liệu Update

Cách 1: Dùng đối tượng Command để cập nhật dữ liệu

Lấy dữ liệu mà user vừa cập nhật vào

```
Dim MaSoISBN As TextBox = e.Item.Cells(0).Controls(0)
Dim IDTacGia As TextBox = e.Item.Cells(1).Controls(0)
Dim Phai As CheckBox = e.Item.FindControl("Phai")
Dim MaSoSach As Integer =
    masterDataGrid.DataKeys(e.Item.ItemIndex)
```

Command

Thi hành lệnh

ASP.NET

Lệnh cập nhật dữ liệu Update

Cách 1: Dùng đối tượng Command để cập nhật dữ liệu

Lấy dữ liệu mà user vừa cập nhật vào

Tạo kết nối với CSDL

```
Dim ConnString As String
    ConnString = "Provider=Microsoft.Jet.OleDb.4.0; Data
Source=" & Server.MapPath("../dbase/books.mdb")
Dim Conn As New OleDbConnection(ConnString)
```

Command

Thi hành lệnh

ASP.NET

Lệnh cập nhật dữ liệu Update

Cách 1: Dùng đối tượng Command để cập nhật dữ liệu

Lấy dữ liệu mà user vừa cập nhật vào

Tạo kết nối với CSDL

Tạo đối tượng Command để cập nhật dữ liệu

```
Dim cmdUpdate As New OleDbCommand
```

```
cmdUpdate.Connection = Conn
```

```
cmdUpdate.CommandText = "UPDATE BooksAuthors " & _
```

```
"Set ISBN = ?, AuthorID = ?, GioiTinh = ? Where MS=?"
```

Lệnh cập nhật dữ liệu Update

```
cmdUpdate.CommandType = CommandType.Text  
cmdUpdate.Parameters.Add("ISBN", MaSoISBN.Text)  
cmdUpdate.Parameters.Add("AuThorID", IDTacGia.Text)  
cmdUpdate.Parameters.Add("GioiTinh", Phai.Checked)  
cmdUpdate.Parameters.Add("MS", MaSoSach)
```

Truyền tham số cho đối tượng
Command

Chú ý: đối với SQL Server tham số phải có dạng
...Add("@AuThorID", @IDTacGia.Text)

Lệnh cập nhật dữ liệu Update

Cách 1: Dùng đối tượng Command để cập nhật dữ liệu

Lấy dữ liệu mà user vừa cập nhật vào

Tạo kết nối với CSDL

```
Conn.Open()  
cmdUpdate.ExecuteNonQuery()  
Conn.Close()
```

Truyền tham số

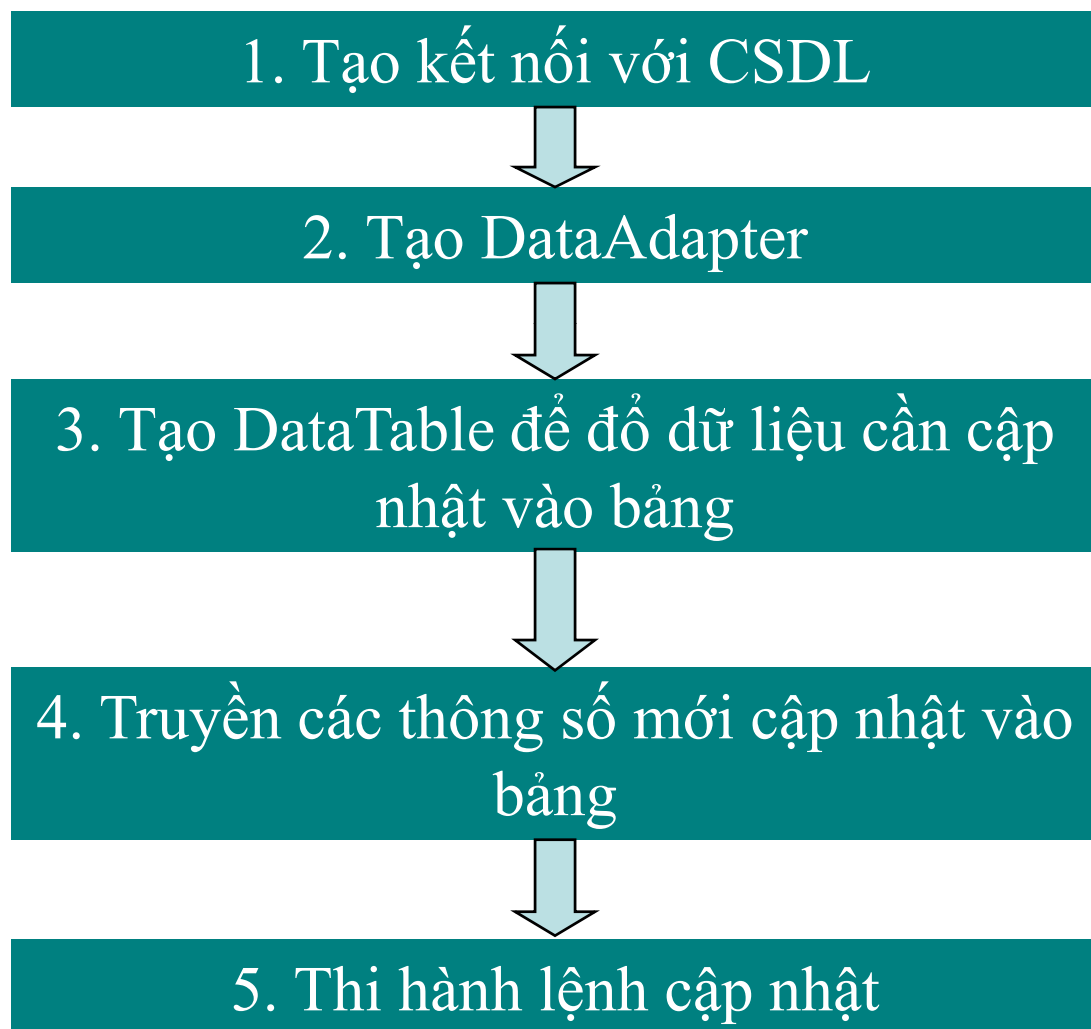
Command

Thi hành lệnh

ASP.NET

Lệnh cập nhật dữ liệu Update

*Cách 2: Dùng đối tượng **DataTable** và **DataAdapter***



Lệnh cập nhật dữ liệu Update

*Cách 2: Dùng đối tượng **DataTable** và **DataAdapter***

1. Tạo kết nối với CSDL

```
Dim ConnString As String
    ConnString = "Provider=Microsoft.Jet.OleDb.4.0; Data
Source=" & Server.MapPath("../dbase/DBHuan.mdb")
Dim Conn As New OleDbConnection(ConnString)
```

4. Truyền các thông số mới cập nhật vào
bảng

5. Thi hành lệnh cập nhật

Lệnh cập nhật dữ liệu Update

*Cách 2: Dùng đối tượng **DataTable** và **DataAdapter***

1. Tạo kết nối với CSDL

2. Tạo DataAdapter

```
Dim IDHinh As Integer =
```

```
masterDataGrid.DataKeys(e.Item.ItemIndex)
```

```
Dim SqlString As String
```

```
SqlString = "SELECT * FROM photo WHERE IDPic=" & IDHinh
```

```
Dim da As New OleDbDataAdapter(SqlString, Conn)
```

3. Thi hành lệnh cập nhật

Lệnh cập nhật dữ liệu Update

*Cách 2: Dùng đối tượng **DataTable** và **DataAdapter***

1. Tạo kết nối với CSDL

2. Tạo DataAdapter

3. Tạo DataTable để đổ dữ liệu cần cập nhật vào bảng

```
Dim PhotoTable As New DataTable  
Dim Com As New OleDbCommandBuilder(da)  
da.Fill(PhotoTable)
```

5. Thi hành lệnh cập nhật

Lệnh cập nhật dữ liệu Update

Cách 2: Dùng đối tượng DataTable và DataAdapter

```
Dim Ngay As TextBox = e.Item.Cells(0).Controls(0)
Dim TenFile As TextBox = e.Item.Cells(1).Controls(0)
Dim TenHinh As TextBox = e.Item.Cells(2).Controls(0)
'-----
PhotoTable.Rows(0)("Date") = Ngay.Text
PhotoTable.Rows(0)("FileName") = TenFile.Text
PhotoTable.Rows(0)("PicName") = TenHinh.Text
```

4. Truyền các thông số mới cập nhật vào
bảng

5. Thi hành lệnh cập nhật

Lệnh cập nhật dữ liệu Update

*Cách 2: Dùng đối tượng **DataTable** và **DataAdapter***

1. Tạo kết nối với CSDL

```
da.Update(PhotoTable)  
PhotoTable.AcceptChanges()  
masterDataGrid.EditItemIndex = -1  
LienKetDuLieu()
```

4. Truyền các thông số mô hình vào
bảng

5. Thi hành lệnh cập nhật

Lệnh xóa dữ liệu - Delete

```
Dim MaSoSach As Integer =  
masterDataGrid.DataKeys(e.Item.ItemIndex)  
Dim Conn As OleDbConnection = TaoKetNoi()  
Dim SqlString As String = "Select * From  
BooksAuthors Where MS=" & MaSoSach  
Dim da As New OleDbDataAdapter(SqlString, Conn)  
Dim Com As New OleDbCommandBuilder(da)  
Dim BangTacGiaSach As New DataTable  
da.Fill(BangTacGiaSach)  
BangTacGiaSach.Rows(0).Delete()  
da.Update(BangTacGiaSach)  
BangTacGiaSach.AcceptChanges()  
LienKetDuLieu()
```

Điều khiển DataList

Thường dùng để hiển thị thông tin một danh sách

Các đặc tính quan trọng của DataList

-RepeatDerection

+ Horizontal



+ Vertical



-RepeatColumns

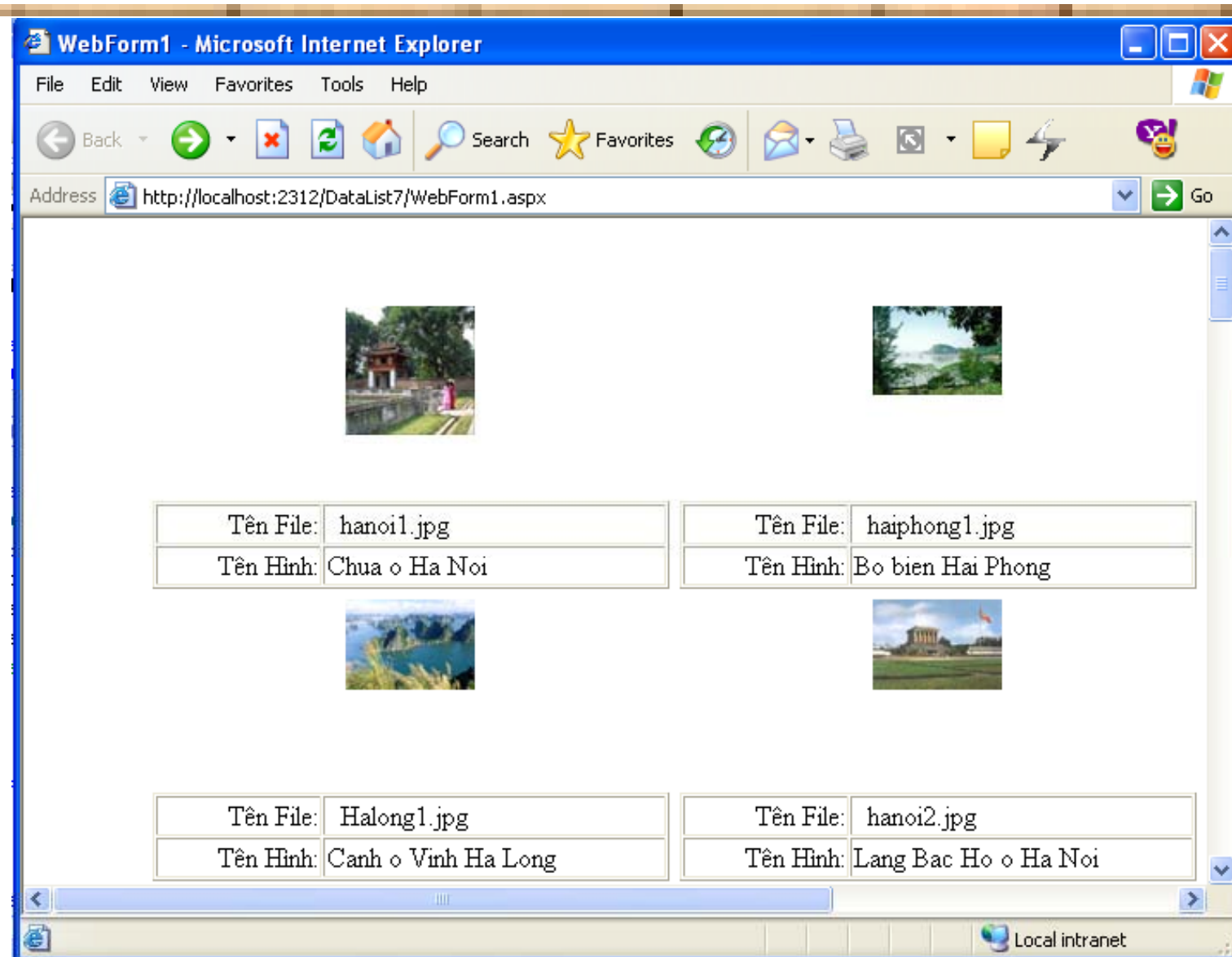
-Layout: Table hoặc Flow

-HorizontalAlign

Điều khiển DataList

*DataList1.DataSource = ds.Tables("photo").DefaultView
DataList1.DataBind()*

```
<asp:DataList id="DataList1" border="0"  
    RepeatDerection="Horizontal" RepeatColumn="3" runat="server">  
<template name="TenTemplate">  
    <table>  
        <tr>  
            <td>  
                <asp: Image ImageUrl='<%#Container.DataItem("FileName") %>'  
                    runat="server" />  
  
            <td>  
                ...  
                <b> <%#Container.DataItem("PicName") %> </b><br>  
                ....  
            </td>  
        </tr>  
    </table>
```



ASP.NET

Điều khiển DataList

Để hiển thị dữ liệu => sự kiện ItemDataBound

Xử lý nhấn nút bấm các button => sự kiện ItemCommand

➤ **Để hiển thị dữ liệu**

```
Dim lblHinh As Label
```

```
    lblHinh = e.Item.FindControl("lblHinh")
```

```
    lblHinh.Text = e.Item.DataItem("PicName")
```

➤ **Hiển thị hình ảnh**

```
Dim hplHinhAnh As HyperLink
```

```
    hplHinhAnh = e.Item.FindControl("hplHinh")
```

```
    hplHinhAnh.ImageUrl = "./Images/" &  
        e.Item.DataItem("FileName")
```

```
    hplHinhAnh.NavigateUrl = "viewphoto.aspx ?FileName=" &  
        lblTenFile.Text
```

Điều khiển Repeater

- HeaderTemplate: Tiêu đề
- ItemTemplate : dữ liệu
- AlternatingItemTemplate: một cách tương đồng của dữ liệu
- SeparatorTemplate: hình thức hiển thị giữa các dòng dữ liệu
- FooterTemplate: tiêu đề dưới

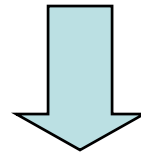
Điều khiển Repeater

Khi kéo Web Control Repeater vào sẽ có dạng:



Repeater - Repeater1

Switch to HTML view to edit the control's templates.



```
<form id="Form1" method="post" runat="server">  
    <asp:Repeater id="Repeater1" runat="server">  
  
    </asp:Repeater>  
</form>
```

Điều khiển Repeater

<HeaderTemplate>

```
<table border="1" width="100%">  
<tr>  
<th>Ma so Vung</th>  
<th>Ten Vung</th>  
</tr>
```

</HeaderTemplate>

<ItemTemplate>

```
<tr>  
<td><%#Container.DataItem("IDPlace")>  
<td><%#Container.DataItem("Name")></td>  
</tr>
```

```
dbcomm = New OleDbCommand(sql,  
dbconn)
```

```
dbread = dbcomm.ExecuteReader()
```

```
Repeater1.DataSource = dbread
```

```
Repeater1.DataBind()
```

Demo

- **GridView**
- **DataList**
- **DetailsView**
- **Form View**

Chương 5: Xây dựng và quản lý Website

ASP.NET OBJECTS thông dụng

- Response Object
- Request Object
- Page Object
- Session Object
- HttpCookie Object
- HttpApplication

Tập tin quản lý và cấu hình ứng dụng

Tạo các mẫu trang có khả năng dùng lại - Pagelets

ASP.NET OBJECTS thông dụng

-Response Object

-Request Object

-Page Object

-Session Object

-HttpCookie Object

-HttpApplication

Response Object

Cho phép Server đáp ứng, trả lời hay thông tin
với Client

Phương pháp (Method) Write

Response.Write("Using Write method of Response object")

Response là một Instance của HttpResponse

Buffering Pages

<% Response.Buffer = False %>

Response Object

➤ Để vận dụng hay xử dụng buffer, ta dùng phương pháp (method):

Clear

Flush

End

➤ **Method Redirect**

```
Response.Redirect("http://www.ASP.net")
```

Dùng Response Object cho phép download File

```
Private Sub LinkButton1_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles
```

```
LinkButton1.Click
```

```
Dim DuongDan As String
```

```
DuongDan = Server.MapPath("../download/tailieu.rar")
```

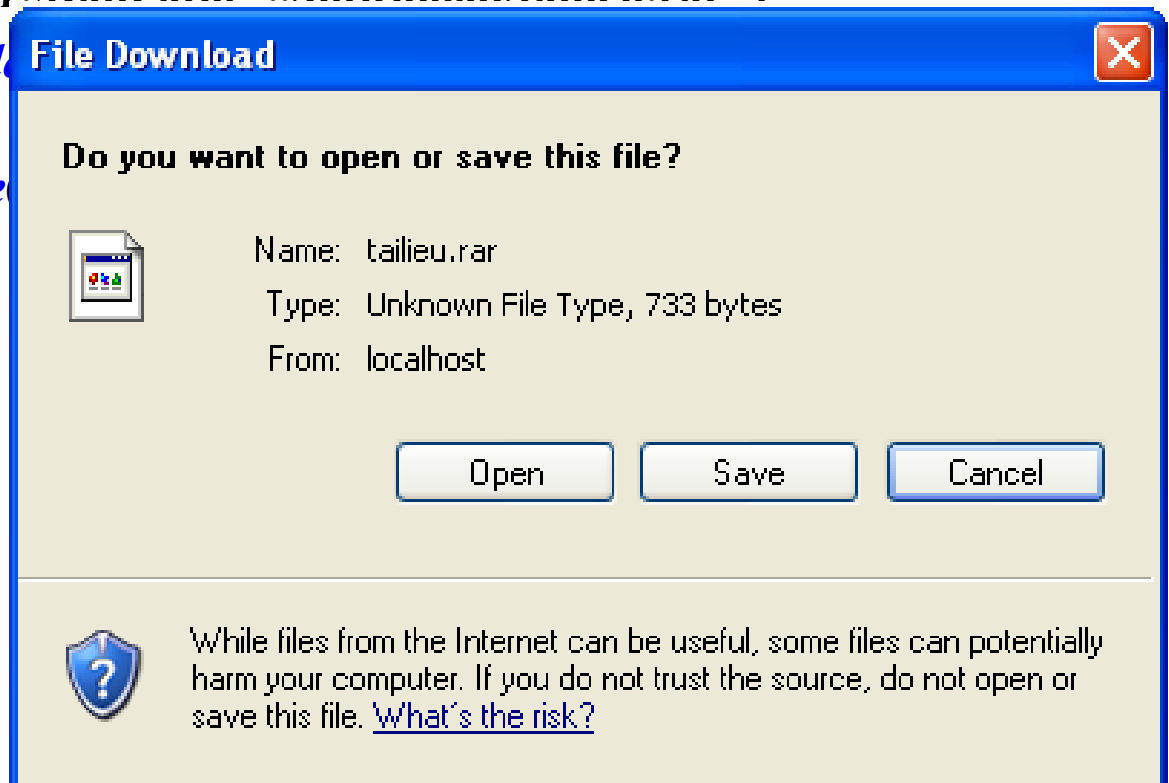
```
Response.AddHeader("Content-Disposition", "attachment; filename=" & DuongDan)
```

```
Response.WriteFile(DuongDan)
```

```
Response.End()
```

```
End Sub
```

LinkButton



Request Object

Dùng để thông tin giữa Server và Client browser

➤ Request.QueryString

VD: `http://www.asp.net?username=vdlung&password=cheery`

‘Để tìm lại toàn bộ thông tin "username=vdlung&password=cheery"

Request.QueryString

‘Để tìm lại giá trị (value) "vdlung"

Request.QueryString("username")

➤ Dưới dạng form:

‘Tìm lại toàn bộ giá trị trong form

Request.Form

‘Tìm lại giá trị value "vdlung"

Request.Form(username)

Dùng **Request** để thu thập thông tin về **ServerVariables** và **Cookies**

Lưu ý:

Thường thì Querystring **chỉ dùng tối đa 255 characters**

Page Object

**Page object có xuất sứ từ Page class
trong .NET Framework**

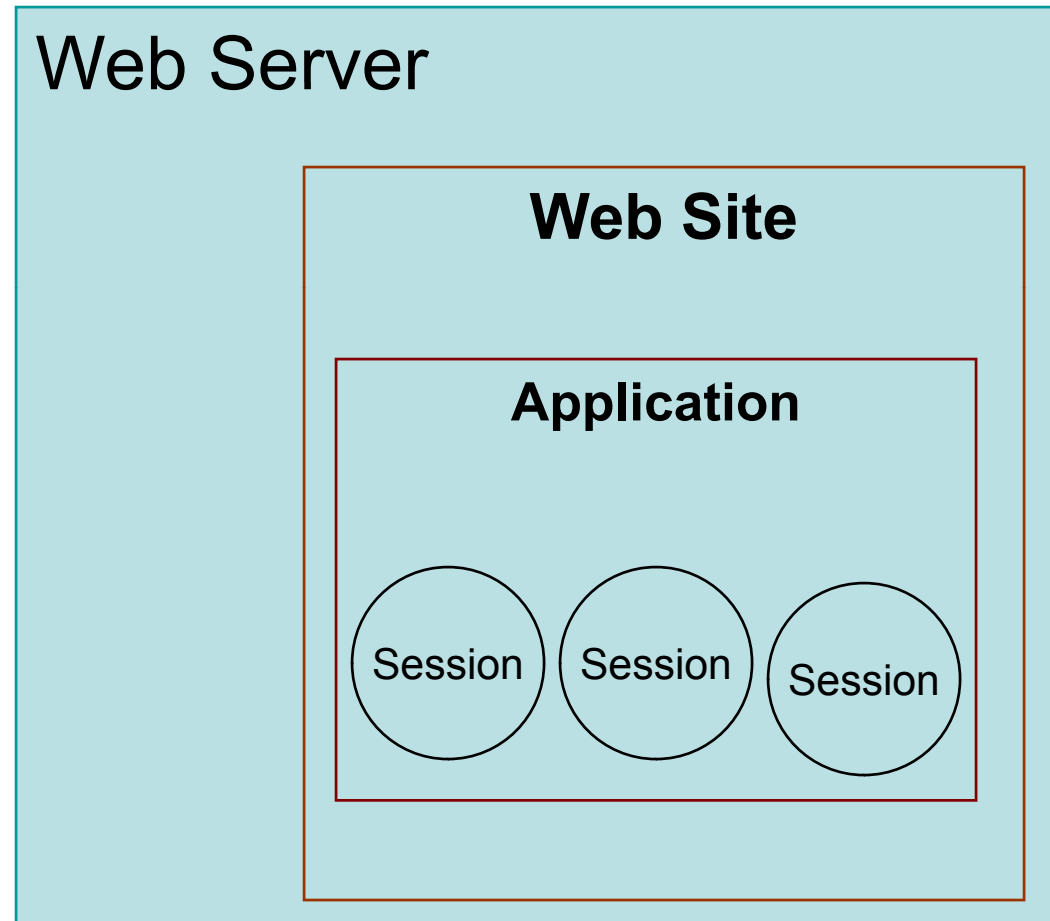
Load: khởi động trang web

IsPostBack: kiểm tra load lần đầu?

Databind: Kết nối với CSDL

Đối tượng Session, Application

- Giúp các trang aspx liên kết và trao đổi dữ liệu cho nhau



Đối tượng Application

- Quản lý tất cả các thông tin của một ứng dụng web
- Thông tin trong Application có thể được xử lý trong bất kỳ trang aspx nào trong suốt chu kỳ sống của ứng dụng

➤ Tạo biến Application:

`Application("tên biến") = <giá trị>`

➤ Lấy giá trị từ biến:

`<biến> = Application("tên biến")`

VD:

Dùng đối tượng này để tính số lượng người truy cập vào website

`Application("SoLanTruyCap") = 1000`

`Application("SoNguoiOnline") = 5`

Khai báo và khởi tạo cho biến Application

Sub `Application_Start`(object sender, EventArgs e)

‘ Số người sử dụng đã truy cập website

`Application("SiteHitCounter") = 1000`

‘ Số người sử dụng đang truy cập website

`Application("CurrentUsers") = 0`

End Sub

Khai báo biến cố `Session_Start`, `Session_End`

Sub `Session_Start`(object sender, EventArgs e)

 Application("SiteHitCounter") = Application("SiteHitCounter") + 1

 Application("CurrentUsers") = Application("CurrentUsers") + 1

End Sub

Sub `Session_End`(object sender, EventArgs e)

 Application("CurrentUsers") = Application("CurrentUsers") - 1

 // sessionstate mode = "InProc"

End Sub

Khair báo lấy giá trị của đối tượng Application

```
protected Sub Page_Load(object sender, EventArgs e)
```

```
Literal1.Text = "Hit Counter: " &  
Application("SiteHitCounter").ToString()
```

```
Literal2.Text = "Current users: " &  
Application("CurrentUsers").ToString()
```

```
End Sub
```

```
<form id="form1" runat="server">  
  <div>  
    <asp:Literal ID="Literal1" runat="server"></asp:Literal><br />  
    <br />  
    <asp:Literal ID="Literal2" runat="server"></asp:Literal>  
  </div>  
</form>
```

Các thuộc tính quan trọng của Application

➤ AllKeys

```
Sub ViewAllKeys()
```

```
    Dim app As String
```

```
    Literal1.Text =
```

```
        "View all keys in Application object by AllKeys property<br>"
```

```
    for each app in Application.AllKeys
```

```
        Literal1.Text &=app & "<br>"
```

```
    Next
```

```
End Sub
```

```
View all keys in Application object by AllKeys property  
SiteHitCounter  
CurrentUsers
```

Các thuộc tính quan trọng của Application

➤ Keys

Sub ViewKeys()

```
Dim app As String
```

```
Literal2.Text = "View all keys in Application object" & _  
                " by Keys property<br>"
```

```
for each ( app in Application.Keys)
```

```
    Literal2.Text &= app & "<br>"
```

```
Next
```

```
End Sub
```

Các thuộc tính quan trọng của Application

➤ Count

Application.Count

```
Sub GetAllValues()
```

```
    Dim i As Integer
```

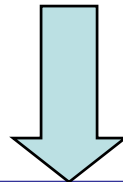
```
    Literal2.Text = "Lấy giá trị của key trong ứng dụng" & _  
                  " nhờ thuộc tính Count<br>"
```

```
    for i = 0 To Application.Count-1
```

```
        Literal2.Text &= Application.GetKeys(i) & "=" & _  
        Application(i).ToString() & "<br>"
```

```
    Next
```

```
End Sub
```



Lấy giá trị của key trong ứng dụng nhờ thuộc tính Count
SiteHitCounter = 1002
CurrentUsers = 5

Các phương thức quan trọng của Application

- **Get(“SitHitCounter”)**
- **Lock**
- **Unlock**
- **Remove(“CurrentUsers”)**
- **RemoveAt(1)**
- **RemoveAll()**

Đối tượng session

- Lưu trữ thông tin của người dùng trong một phiên làm việc
- Thông tin trong Session có thể được xử lý trong bất kỳ trang aspx nào trong suốt phiên làm việc hiện hành

➤ Tạo biến Session:

`Session("tên biến") = <giá trị>`

➤ Lấy giá trị từ biến:

`<biến> = Session("tên biến")`

VD:

Khi chưa đăng nhập vào hệ thống:

`Session("UserID") = 0`

`Session("UserName") = ""`

Sau khi đăng nhập thành công, thông tin trong session sẽ được cập nhật lại như:

`Session("UserID") = 1`

`Session("UserName") = "vdlung"`

Session Object

```
<%@Page Language="VB" %>
```

```
<script runat="server">
```

```
sub Submit_Click(obj as object,e as EventArgs)
```

```
    if tbName.Value <>""
```

```
        Session("Name ")=tbName.Value
```

```
        Response.Write("Hi " & Session("Name ")&"!")
```

```
    else
```

```
        Response.Write("You forgot to enter a name.")
```

```
    end if
```

```
end sub
```

```
</script>
```

Session Object

- Thường dùng session để nắm giữ các thông tin như UserName, email hoặc UserID
- Dựa vào các thông tin này mà phân quyền truy cập cho user
- Ví dụ:

Start Page | WebForm1.aspx | **Login.aspx** | Login.aspx.vb

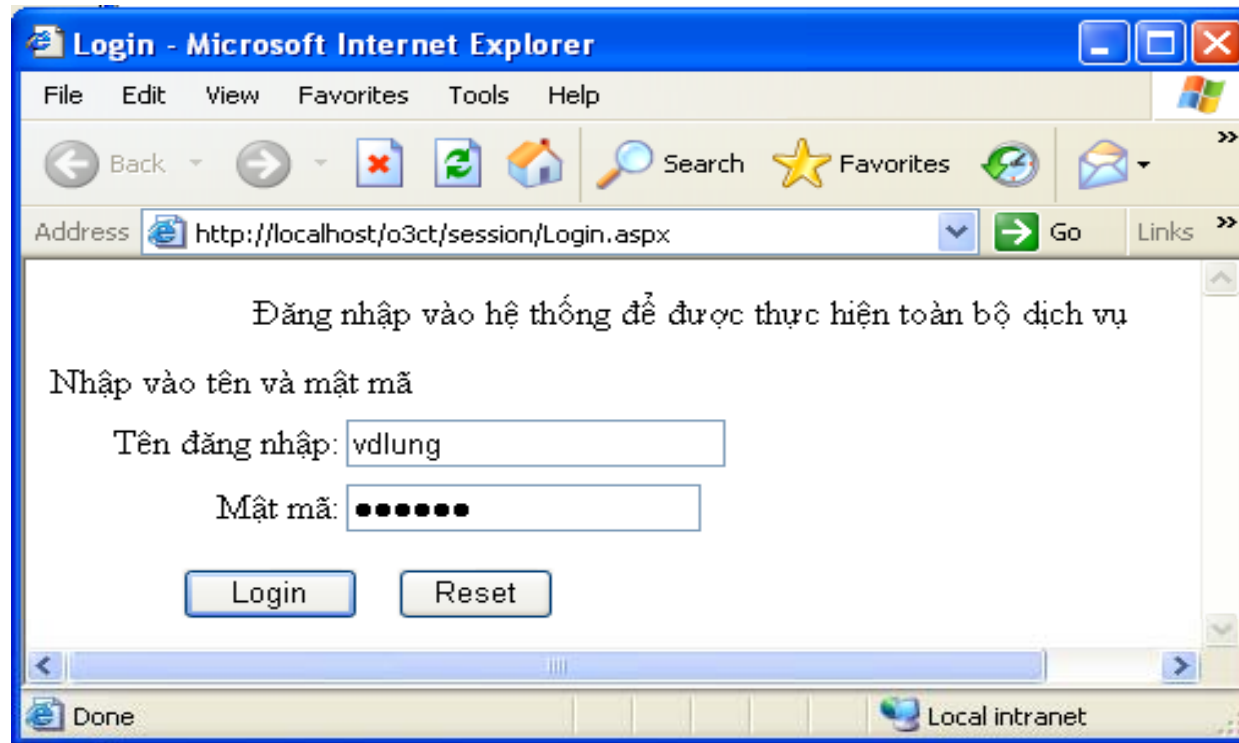
Đăng nhập vào hệ thống để được thực hiện toàn bộ dịch vụ

Nhập vào tên và mật mã

Tên đăng nhập:

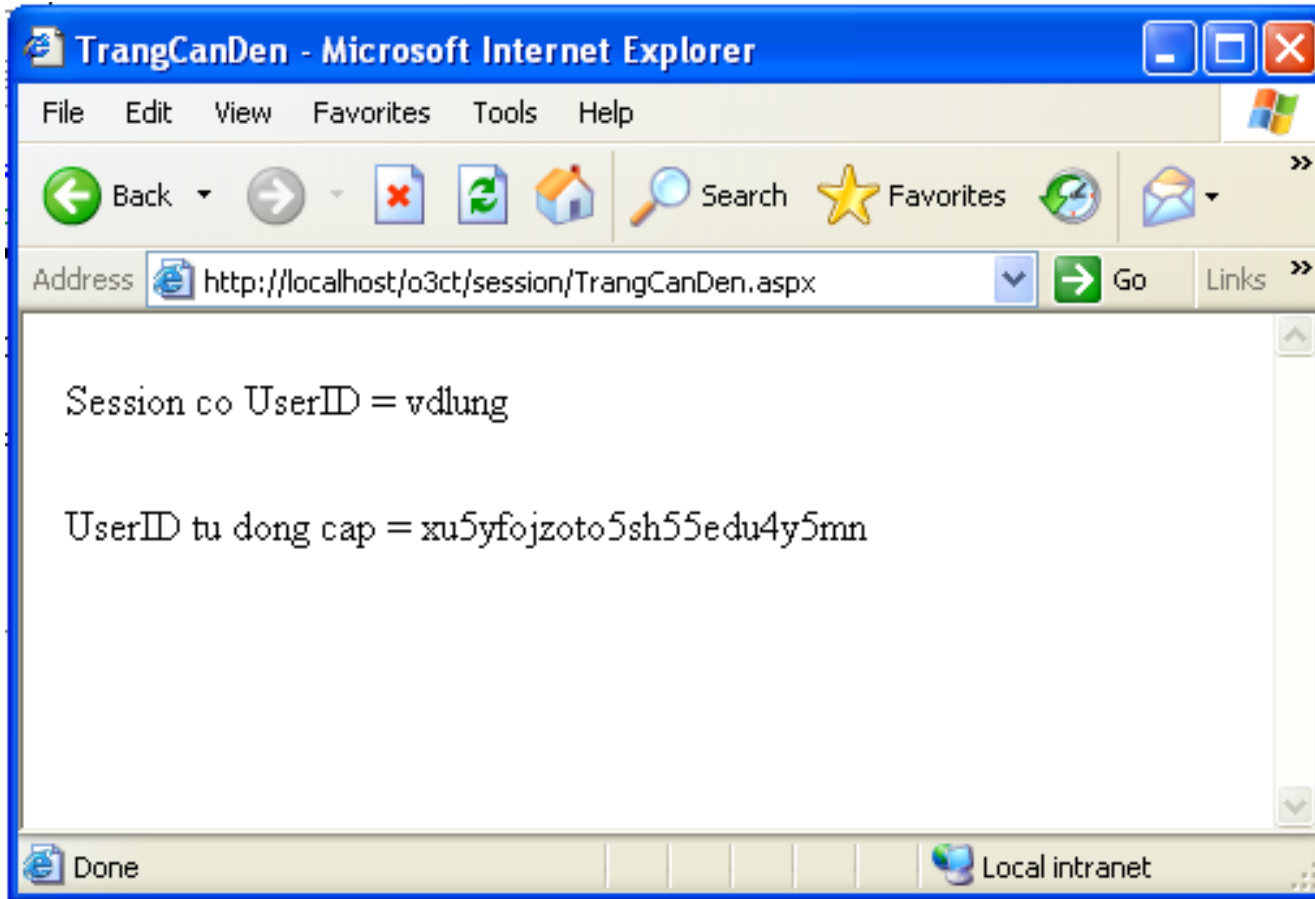
Mật mã:

Session Object



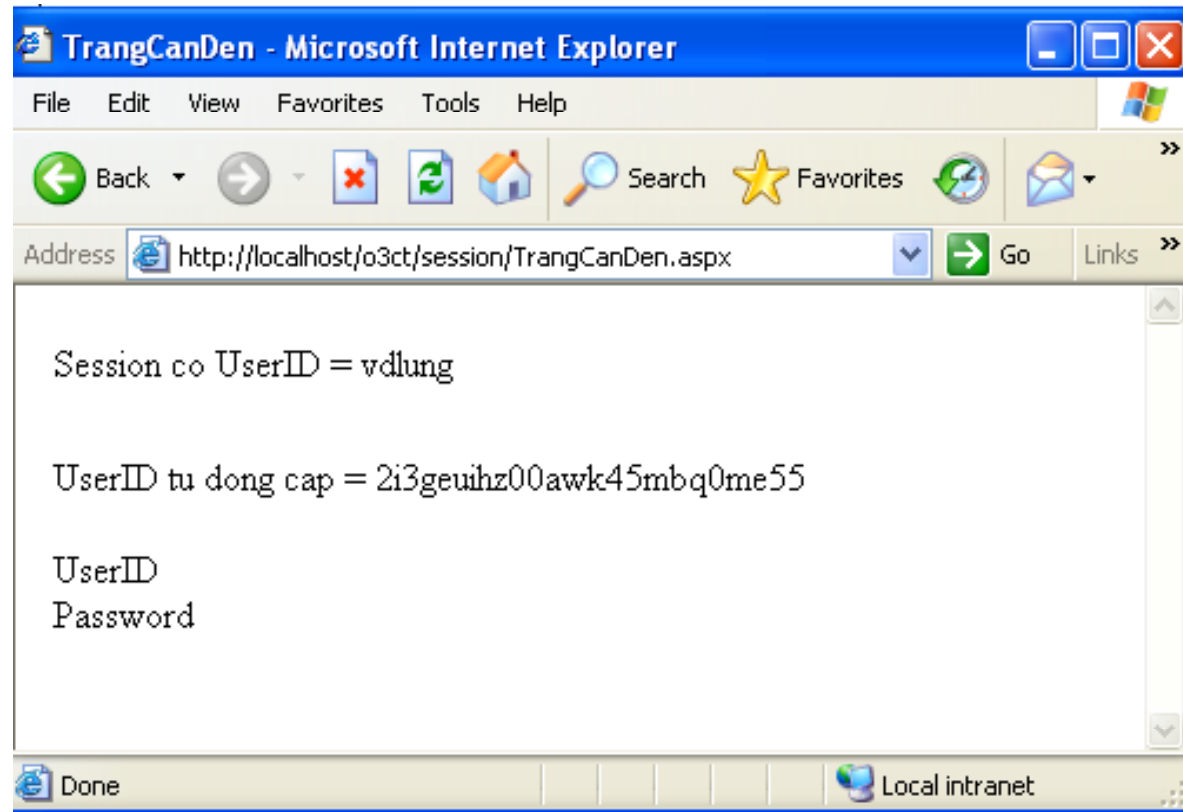
```
Private Sub btnLogin_Click(...) Handles btnLogin.Click
    If (tbUserID.Text <> "") Then
        Session("UserID") = tbUserID.Text
        Response.Redirect("TrangCanDen.aspx")
    End If
End Sub
```

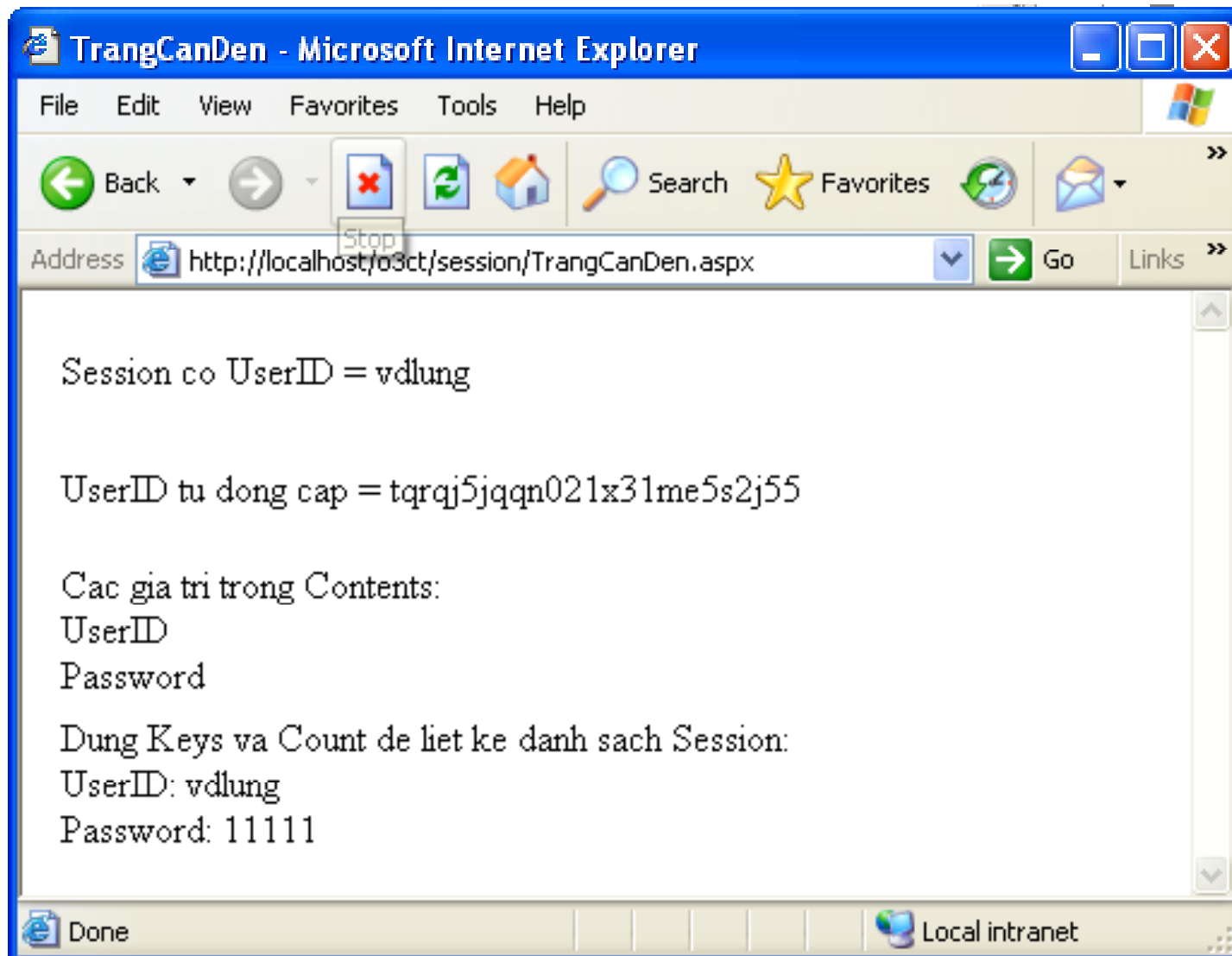
Session Object



Các thuộc tính của Session Object

- *Contents*
- *Keys*
- *Count*





Session Object

Kiểm soát (control) Session Object

Có nhiều cách kiểm soát đối tượng (Object) Session trong trang ASP.NET, cơ bản gồm có:

- **Timeout** : `Session.Timeout = 60` **Remove("UserID")**
- **Abandon** **RemoveAt(1)**
- **IsNewSession** **RemoveAll()**

Abandon dùng để kết thúc 1 session ngay tức khắc. Thí dụ, sau khi user kiểm tra email (Web Email) xong và logout để người khác không thể dùng ... 'ké' lúc họ vắng mặt, ta có thể kết thúc session đó bằng cách:

`Session.Abandon`

Sessions không Cookies

ASP.NET dùng Cookies để lưu trữ Session ID và thông tin về user. Nếu browser của user không yểm trợ cookies hoặc trong nhiều trường hợp, user không muốn (hay chấp nhận) yểm trợ cookies?

ASP.NET dùng **Cookie Munging** => ASP.NET phải mã hoá và giải mã (encode and decode) Session ID cùng với các thông tin liên hệ và đặt vào các trang Web trước khi gửi đi.

Tiến trình Sessions không Cookies

➤ User yêu cầu hiển thị 1 trang Web

➤ ASP.NET sẽ dò (scan) từng nối (hyperlinks) trong mã HTML (HTML code)

Cứ cuối mỗi nối như vậy, ASP.NET sẽ gắn thêm phần Session ID đã được mã hoá (encode), vd:

```
<a href="sessionwithoutcookies.aspx?SessionID=1ma2hoa3giai4ma5hoai6lam7met8qua9....">
```

➤ Khi user nhập 1 nối, ASP.NET sẽ giải mã (decode) và chuyển phần Session đó tới trang Web user yêu cầu

➤ Trang Web này vận dụng các biến số (variables) của Session và lập lại tiến trình để có thể giữ thông tin giữa các trang Web với nhau

HttpCookie Object

HttpCookie object cung cấp các phương pháp (method) để tạo ra và vận dụng những cookies. Có 2 loại Cookie: Temporary và Persistent

- Để đọc và ghi thông tin vào Cookie dùng *HttpCookie*
- Để truy cập từng phần tử của Cookie dùng *HttpCookieCollection*

Cách dùng:

- Khai báo và khởi tạo:

```
Dim pCookie As HttpCookie = New HttpCookie("Remember",  
"Yes")
```

- Ghi đối tượng HttpCookie xuống máy tính
`Response.Cookies.Add(pCookie)`

HttpCookie Object



HttpCookie Object

Tuy nhiên, phương pháp (method) dùng **Request** và **Response Objects** là các phương pháp (method) thông dụng nhất để vận dụng cookie

Tạo ra Cookies

Ta dùng Response Object để tạo ra Cookies với 2 cách sau:

- Tạo nhiều Cookies, mỗi cookie kèm với một giá trị (gọi là Cookies đa dạng, đơn giá trị)
- Tạo một Cookie với nhiều cặp key/value (gọi là Cookie đơn dạng, đa giá trị)

Ví dụ:

'Kiểu Cookies đa dạng, đơn giá trị

```
Response.Cookies("MyCookie1").Value = "Single Cookie Value 1"
```

```
Response.Cookies("MyCookie2").Value = "Single Cookie Value 2" ‘
```

Kiểu Cookie đơn dạng, đa giá trị

```
Response.Cookies("MyASPNETPage").("Username") = "nguyenvana"
```

```
Response.Cookies("MyASPNETPage").("Password") = "11111111"
```

Liên hệ với Cookies

'Cho kiểu Cookies đa dạng, đơn giá trị

```
Response.Write (Request.Cookies("MyCookie1").Value)
```

```
Response.Write (Request.Cookies("MyCookie2").Value)
```

'Cho kiểu Cookie đơn dạng, đa giá trị

```
Response.Write (Response.Cookies("MyASPNETPage").("Username"))
```

```
Response.Write (Response.Cooies("MyASPNETPage").("Password"))
```

Chú ý!!!!

Không thể trộn 2 kiểu trên với nhau. **Ta không thể nào tạo ra một Cookie vừa có Value vừa có key.**

Đáo hạn (Expires) Cookies

□ Kiểu ngày giờ đáo hạn

```
Response.Cookies("MyASPNETPage").Expires =
```

```
    DateTime.FromString("30/02/2003")
```

'hoặc là

□ Kiểu khoảng thời gian đáo hạn

```
Response.Cookies("MyASPNETPage").Expires =
```

```
    DateTime.Now.AddMonths(1)
```

Các thuộc tính quan trọng của HttpCookie

➤ Domain

➤ Path

➤ HasKeys

➤ Secure

- Domain dùng để giới hạn việc sử dụng cookies ở một domain ta chỉ định, ví dụ như `www.myserver.com`

- Path dùng tương tự như Domain, nhưng giới hạn việc sử dụng cookies ở path chỉ định nào đó trong Server của ta.

- HasKeys báo cho ta biết Cookie dùng kiểu 'đa dạng, đơn giá trị' với các cặp key/value.

- Secure báo cho ASP.NET biết nên chuyển cookie 1 cách an toàn hay không và thường chỉ xảy ra ở trên HTTPS protocol. Giá trị mặc định (default) của Secure là False.

Ví dụ Cookie

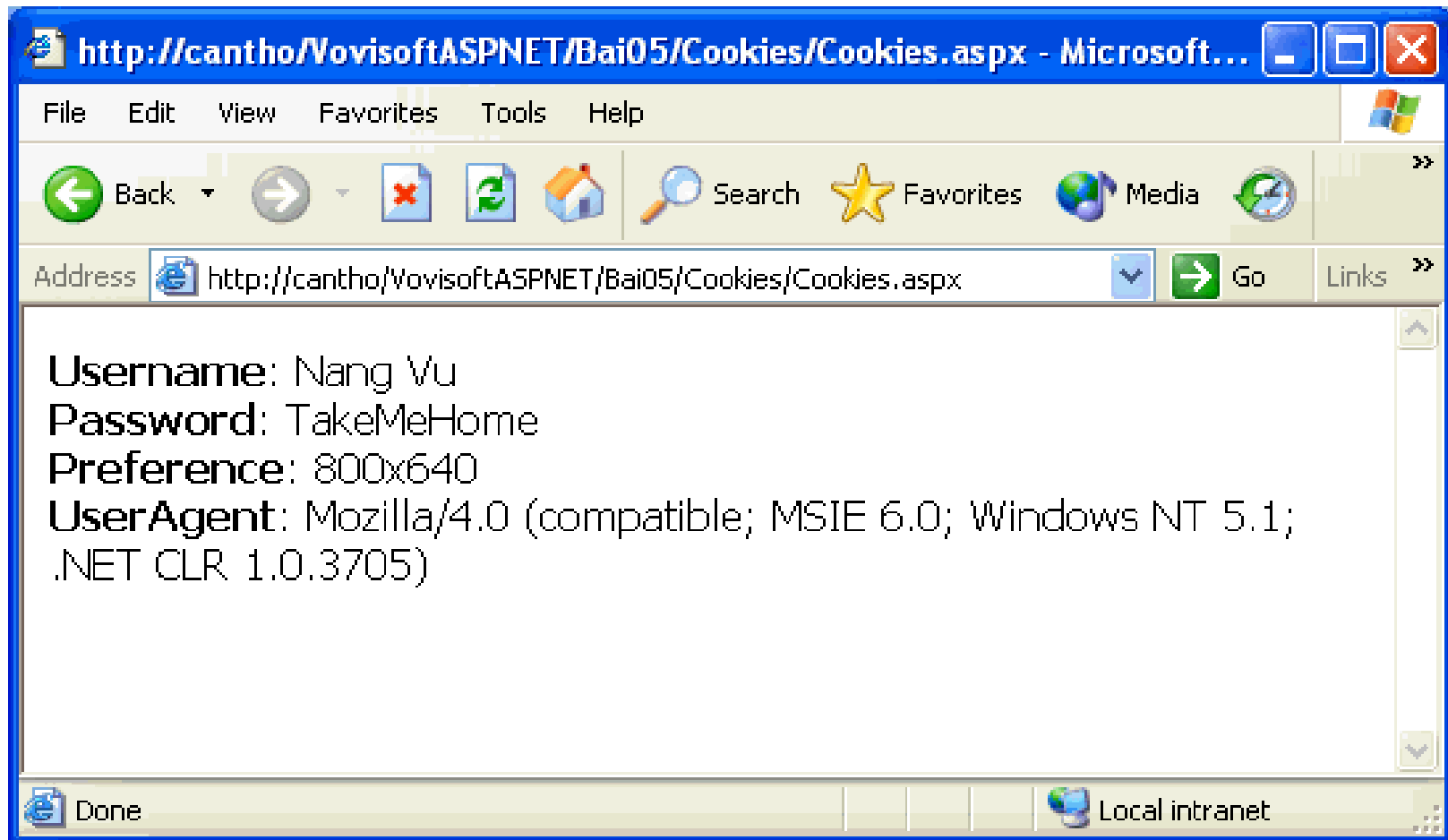
```
sub Page_Load(obj as object, e as EventArgs)
    dim strVariable as string

    'set up some cookie variables
    Response.Cookies("MyASPNETPage")("Username") = "Nang Vu"
    Response.Cookies("MyASPNETPage")("Password") = "TakeMeHome"
    Response.Cookies("MyASPNETPage")("Preference") = "800x640"
    Response.Cookies("MyASPNETPage")("UserAgent") =
        Request.ServerVariables("HTTP_USER_AGENT")

    for each strVariable in Response.Cookies("MyASPNETPage").Values
        lblCookies.Text &= "<b>" & strVariable & "</b>: " &
        Request.Cookies("MyASPNETPage")(strVariable) & "<br>"
    next

end sub
```

Ví dụ Cookie



Tập tin quản lý và cấu hình ứng dụng

a) Tập tin Global.asax

- Khai báo các biến cố => nắm bắt hành động của người sử dụng
- Khai báo và khởi tạo giá trị cho các biến Application, Session
- Viết xử lý cho các sự kiện của hai đối tượng Application và Session
- Tự động sinh ra khi biên dịch ứng dụng

Cấu trúc tập tin Global.asax

```
Public Class Global
    Inherits System.Web.HttpApplication

    Sub Application\_Start(ByVal sender As Object, ByVal e As EventArgs)
        ' Fires when the application is started
    End Sub

    Sub Session\_Start(ByVal sender As Object, ByVal e As EventArgs)
        ' Fires when the session is started
    End Sub

    Sub Application\_BeginRequest(ByVal sender As Object, ByVal e As EventArgs)
        ' Fires at the beginning of each request
    End Sub
```

Tập tin quản lý và cấu hình ứng dụng (tt)

```
Sub Application_AuthenticateRequest(ByVal sender As Object, ByVal e As EventArgs)
```

```
    ' Fires upon attempting to authenticate the user
```

```
End Sub
```

```
Sub Application_Error(ByVal sender As Object, ByVal e As EventArgs)
```

```
    ' Fires when an error occurs
```

```
End Sub
```

```
Sub Session_End(ByVal sender As Object, ByVal e As EventArgs)
```

```
    ' Fires when the session ends
```

```
End Sub
```

```
Sub Application_End(ByVal sender As Object, ByVal e As EventArgs)
```

```
    ' Fires when the application ends
```

```
End Sub
```

```
End Class
```

Khai báo và khởi tạo cho biến Application

```
void Application_Start(object sender, EventArgs e)
{
    Application.Lock()
    ‘ Số người sử dụng đã truy cập website
    Application("SiteHitCounter") = 1000
    ‘Số người sử dụng đang truy cập website
    Application("CurrentUsers") = 0
    Application.Unlock()
}
```

Khai báo biến cố `Session_Start`, `Session_End`

```
void Session_Start(object sender, EventArgs e)
{
    Application("SiteHitCounter") = Application("SiteHitCounter") + 1
    Application("CurrentUsers") = Application("CurrentUsers") + 1
}
```

```
void Session_End(object sender, EventArgs e)
{
    Application("CurrentUsers") = Application("CurrentUsers") - 1
    // sessionstate mode = "InProc"
}
```

Tập tin quản lý và cấu hình ứng dụng (tt)

b) Tập tin web.config

Lưu trữ thông tin cấu hình của ứng dụng

- Tạo ra tự động
- Viết theo định dạng XML

```
<sessionState
```

```
    mode="InProc" ' sqlserver hoặc stateserver
```

```
    stateConnectionString="tcpip=127.0.0.1:42424"
```

```
    sqlConnectionString="data
```

```
source=127.0.0.1;Trusted_Connection=yes"
```

```
    cookieless="false"
```

```
    timeout="20"
```

```
/>
```

Tạo các mẫu trang có khả năng dùng lại - Pagelets

- Tương tự như `#include` trong ASP
- Tên biến: cục bộ
- Có thể truyền tham số cho các mẫu trang pagelets khi triệu gọi hoặc nhúng trang
- Có thể nhúng nhiều lần trong 1 trang
- Mã lệnh trong pagelets có thể viết bằng các ngôn ngữ khác với ngôn ngữ của trang chính

Tạo các mẫu trang có khả năng dùng lại - Pagelets

Cách dùng:

Chuyển phần mã lệnh cần dùng chung vào một file có phần mở rộng là .aspc ví dụ vào file “TrangNhung.aspc”

(!!!Lưu ý: bên trong pagelets không nên chứa các thẻ như <html> hay <body>)

Trong trang chính (ví dụ default.aspx):

```
<%@ Register TagPrefix="myPagelet" TagName="allOfInput"
    src="TrangNhung.aspc"%>
<html>
<title> Thử sử dụng nhúng vào trang này </title>
<!-- Chèn mẫu trang pagelets -->
<myPagelet:allOfInput id="PageletCtrl" runat="server" />
</html>
```

Định nghĩa thuộc tính và phương thức cho Pagelets

```
<script language="VB" runat="server" >
```

```
Public Property ISBN As String
```

```
Get
```

```
    ISBN = txtISBN.Text
```

```
End Get
```

```
Set
```

```
    txtISBN.Text = ISBN
```

```
End Set
```

```
End Property
```

```
Public Function MotHam() As String
```

```
    ....
```

```
    MotHam = <Gia tri tra ve>
```

```
End Function
```

```
    ....
```

```
</script
```


Dùng các thuộc tính

- Truy xuất thuộc tính ISBN

```
strISBN = PageletCtrl.ISBN
```

- Đặt giá trị cho ô nhập liệu

```
PageletCtrl.ISBN = "1-02342-44413"
```

Hoặc ngay khi nhúng:

```
<myPagelet:allOfInput id="PageletCtrl"  
    ISBN="1-02342-44413"  
    runat="server" />
```

Hoặc gọi phương thức của Pagelets:

```
Ketqua = PageletCtrl.MotHam()
```



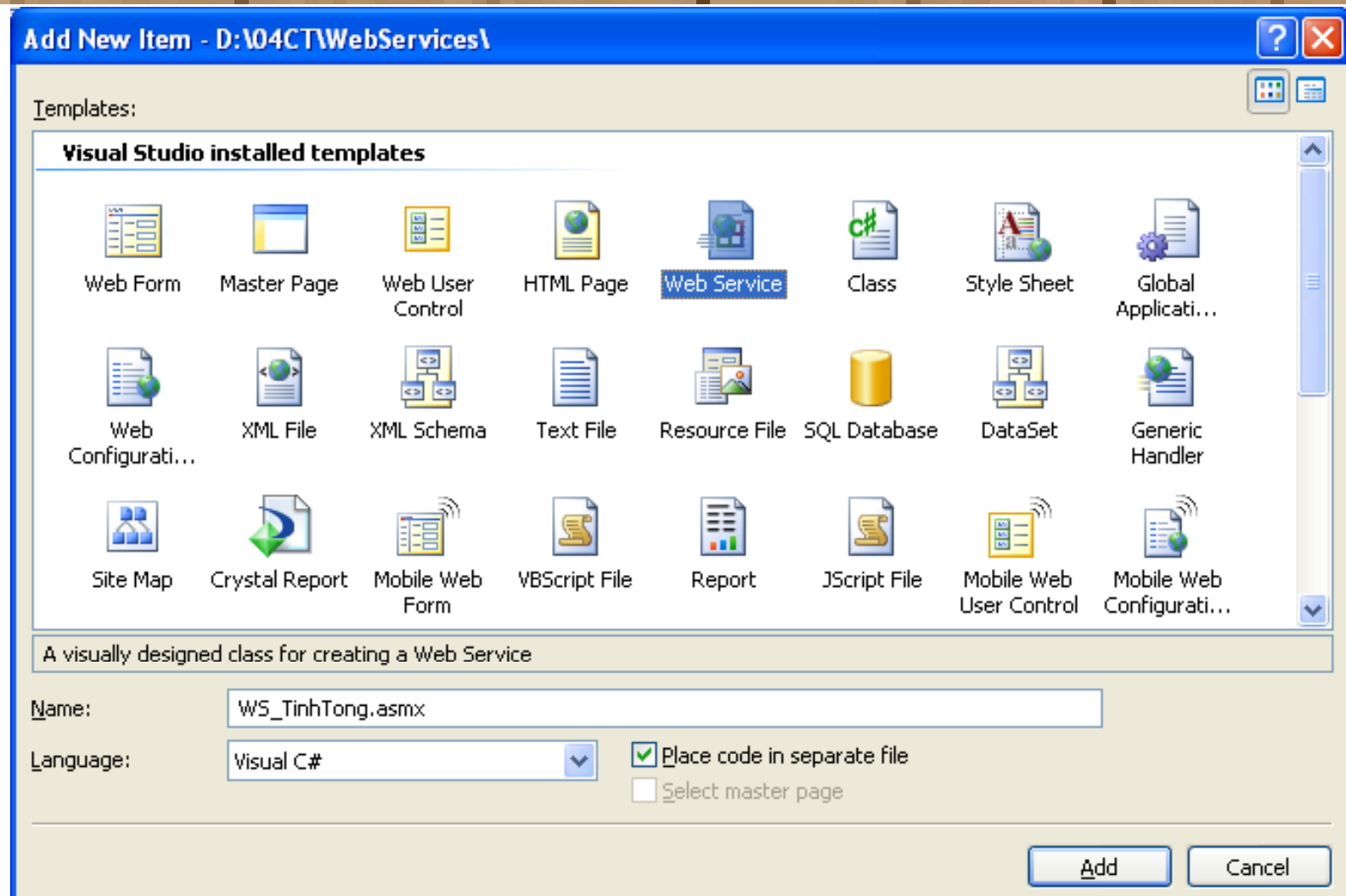
Chương VI: Web services

- Các nhược điểm của DCOM (Distributed Component Object Model):
 - Gửi các thông tin dưới dạng nhị phân
 - Bị chặn bởi Firewall và Proxy Server
 - Truy cập thông qua giao thức Web TCP/IP.
- Webservices xây dựng trên SOAP (Simple Object Access Protocol) => gọi thực hiện và trả về kết quả là dạng TEXT (XML) => có khả năng hoạt động xuyên qua Firewall:
 - Khả năng làm việc trên mọi hệ điều hành
 - Mở rộng khả năng phối hợp giữa các ứng dụng
 - Tăng cường giao tiếp giữa Client/Server
- Ví dụ web services:
 - + Đường sắt Việt Nam cung cấp các web services cho phép đặt mua vé tàu
 - + Các nhà hàng, khách sạn cung cấp các web services cho phép đặt tiệc, đặt phòng

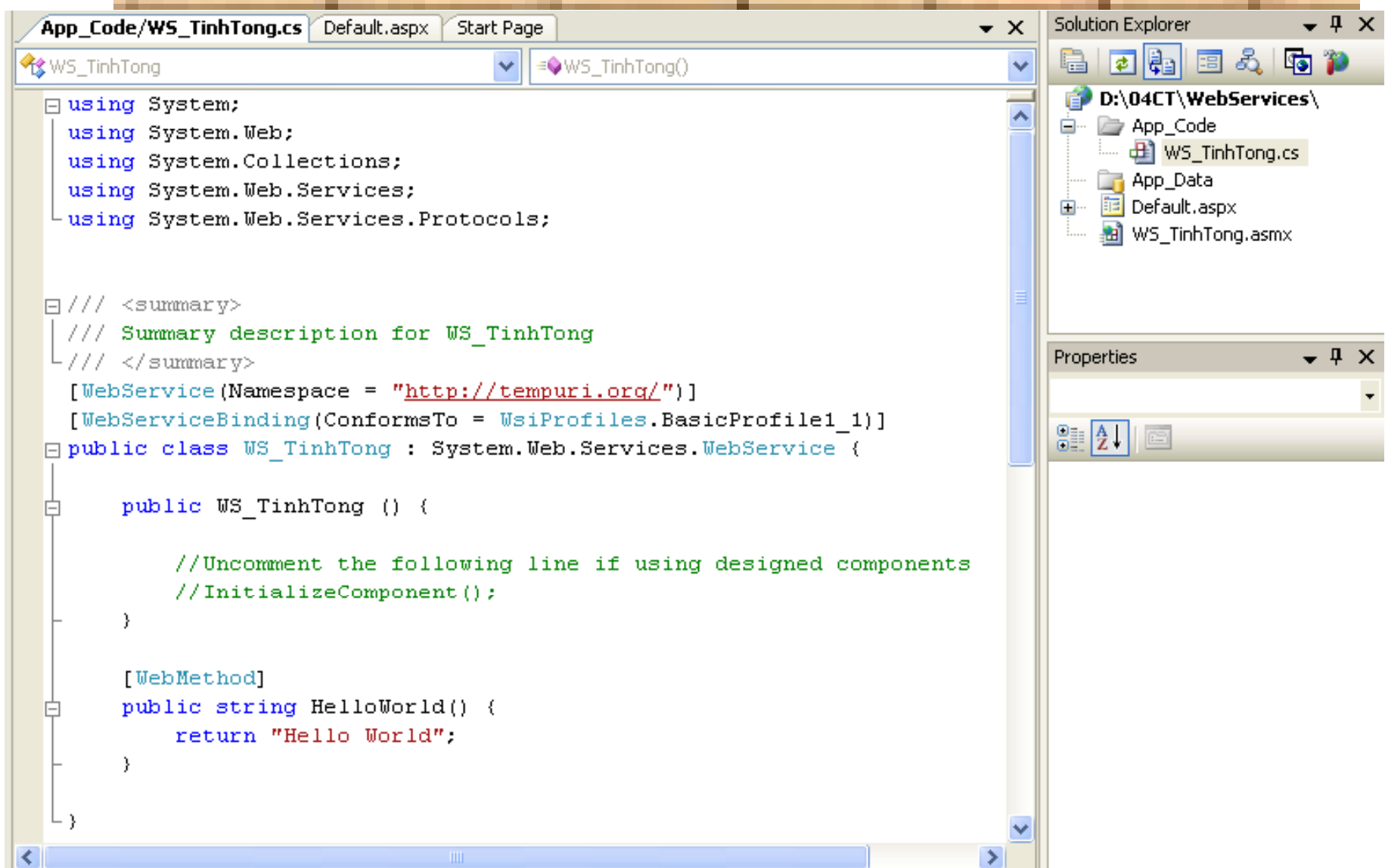
Các bước xây dựng Web services

1. Tạo Web services
Cho khởi động file service.asmx
2. Kiểm tra Web service
3. Sử dụng Web service do mình tạo ra
4. Sử dụng Web service được cung cấp miễn phí trên mạng

Tạo Web services



Tạo Web services (tt)



```
using System;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;

/// <summary>
/// Summary description for WS_TinhTong
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class WS_TinhTong : System.Web.Services.WebService {

    public WS_TinhTong () {

        // Uncomment the following line if using designed components
        // InitializeComponent();

    }

    [WebMethod]
    public string HelloWorld() {
        return "Hello World";
    }
}
```

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the C# code for `WS_TinhTong.cs`. The code includes using statements for `System`, `System.Web`, `System.Collections`, `System.Web.Services`, and `System.Web.Services.Protocols`. It features a summary comment, a `WebService` attribute with namespace `http://tempuri.org/`, and a `WebServiceBinding` attribute. The `WS_TinhTong` class inherits from `System.Web.Services.WebService` and contains a constructor and a `HelloWorld` method that returns `"Hello World"`.
- Solution Explorer:** Shows the project structure for `D:\04CT\WebServices\`, including folders `App_Code` and `App_Data`, and files `Default.aspx` and `WS_TinhTong.asmx`.
- Properties Window:** Is currently empty.

Kiểm tra Web service



WS_TinhTong Web Service - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites RSS Print Mail Stop

Address http://localhost:1118/WebServices/WS_TinhTong.asmx Go

WS_TinhTong

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Cong](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the `WebService` attribute's `Namespace` property. The `WebService` attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "`http://microsoft.com/webservices/`":

```
C#  
[WebService(Namespace="http://microsoft.com/webservices/")]
```

Local intranet

Sử dụng Web service do mình tạo ra

The screenshot displays a Microsoft Internet Explorer window titled "WS_TinhTong Web Service - Microsoft Internet Explorer". The address bar shows the URL: `http://localhost:1118/WebServices/WS_TinhTong.aspx?op=Cong`. The main content area features a dark blue header with the text "WS_TinhTong" and a link: "Click [here](#) for a complete list of operations."

Below the header, there is a section titled "Cong" and a "Test" section. The "Test" section contains the instruction: "To test the operation using the HTTP POST protocol,". A table is provided for entering test parameters:

Parameter	Value
x:	4
y:	2

Below the table is an "Invoke" button. An inset window shows the XML response received from the service:

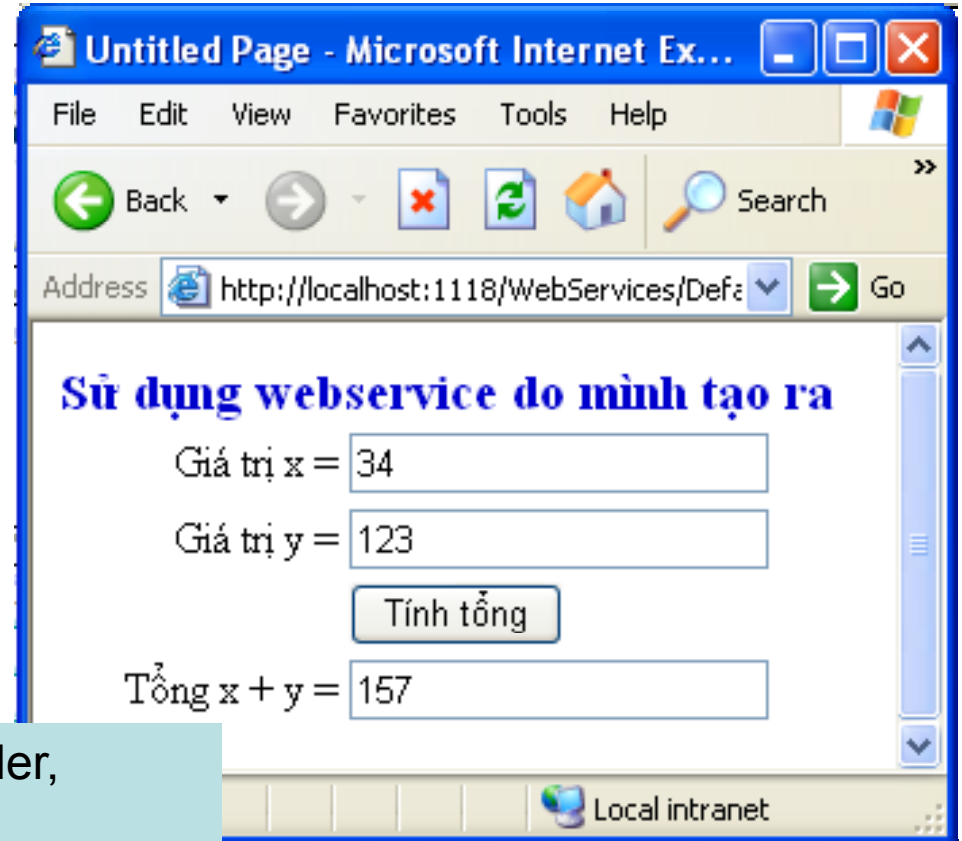
```
<?xml version="1.0" encoding="utf-8" ?>
<int xmlns="http://tempuri.org/">6</int>
```

The status bar at the bottom of the browser indicates "Done" and "Local intranet".

Sử dụng Web service do mình tạo ra

Sử dụng webservice do mình tạo ra

Giá trị x =	<input type="text"/>
Giá trị y =	<input type="text"/>
	<input type="button" value="Tính tổng"/>
Tổng x + y =	<input type="text"/>



```
protected void Button1_Click(object sender,
EventArgs e) {
    WS_TinhTong sum = new WS_TinhTong();
    int x, y;
    x = Convert.ToInt32(txt_X.Text.ToString());
    y = Convert.ToInt32(txt_Y.Text.ToString());
    txtTong.Text = sum.Cong(x,y).ToString();
}
```

Sử dụng Web service được cung cấp miễn phí trên mạng

Microsoft Internet Explorer window showing the website **webserviceX.NET**. The address bar displays `http://www.webservicex.net/WCF/Default.aspx`.

The website content includes:

- Navigation: Home, Web Services, Tools, Support, Contact
- Top Web Services:
 - Stock Quote
 - Currency Convertor
 - Global Weather
 - London Gold And Silver Fixing
 - SendsSMSWorld
 - OFAC SDN and Blocked Persons
 - Translation Engine
 - ABA Lookup
- Welcome to webserviceX.NET:

Web services enable to quickly integrate applications across multiple platforms, systems and even across businesses. Emerging web services standards such as SOAP, WSDL and UDDI will enable system-to-system communication that is easier and cheaper than ever before.
- New Web Services:
 - LloydsRiskCodeService
 - North American Industry Classification System
 - United Nations Standard Products and Services Code
 - FedACH
 - Medi Care Supplier
 - Barcode Generator
 - USA Weather Forecast
 - GeoIPService

The status bar at the bottom shows `http://www.webservicex.net/WCF/ServiceDetails.aspx?SID=46` and Internet.

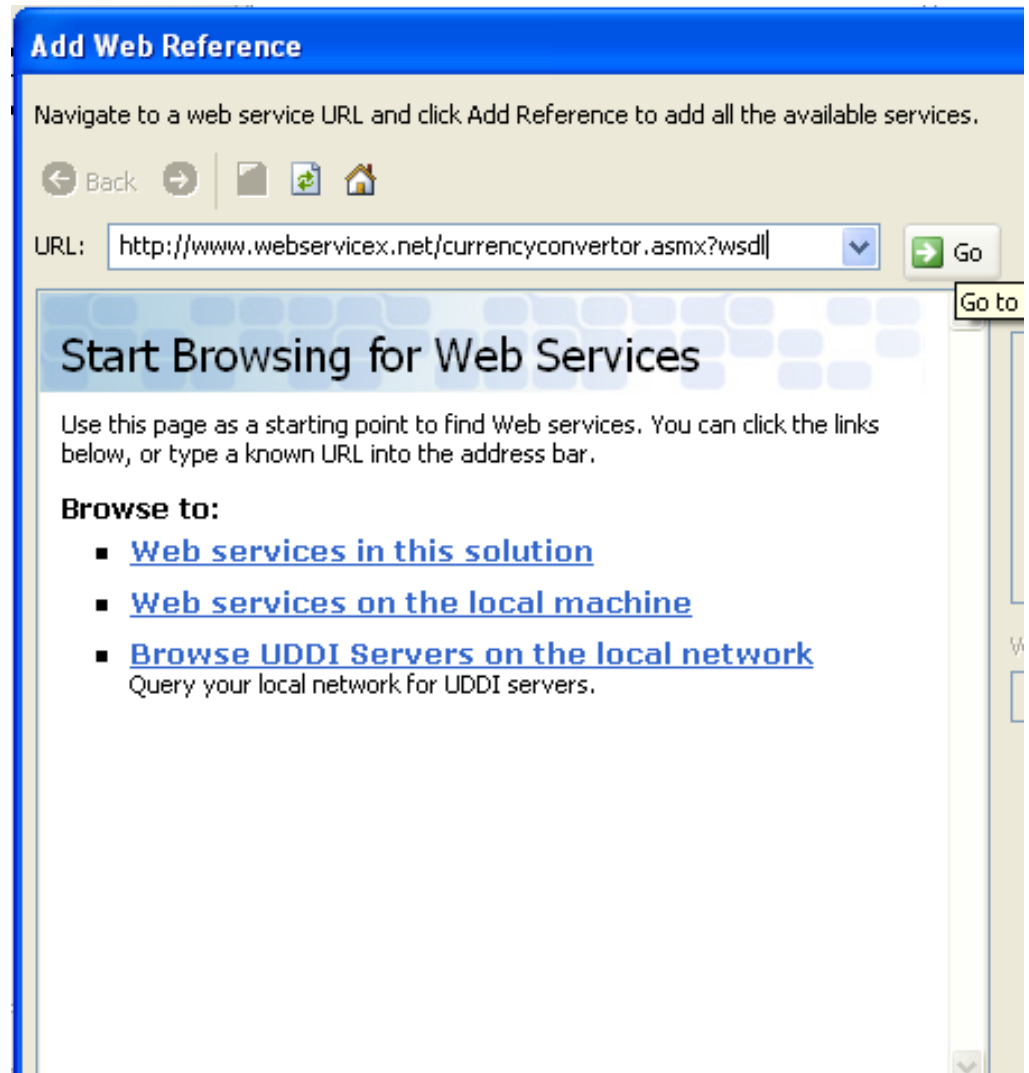
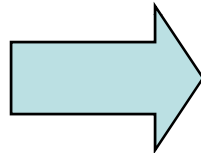
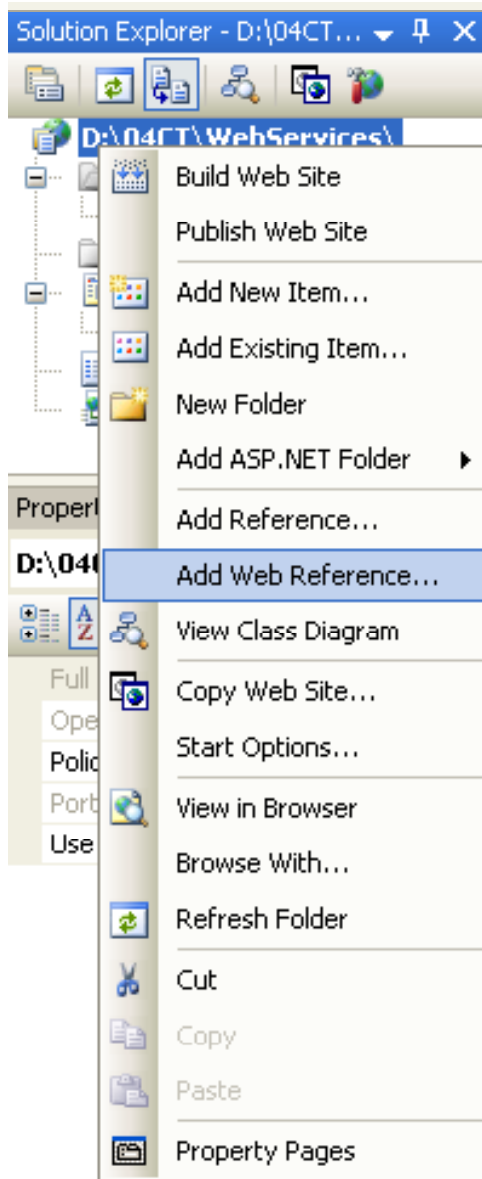
Các webservicex thông dụng

<http://www.websvcicex.net/currencyconvertor.asmx?wsdl>

<http://www.websvcicex.net/globalweather.asmx?wsdl>

<http://www.websvcicex.net/country.asmx?wsdl>

Các bước sử dụng webservice



Các bước sử dụng webservice (tt)

Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available services.

Back

URL: Go

"CurrencyConvertor" Description

Methods

- **ConversionRate** (*FromCurrency As Currency* , *ToCurrency As Currency*) As *double*

Get conversion rate from one currency to another currency

<p><u>Different currency Code and Names around the world</u></p><blockquote><p>AFA-Afghanistan Afghani
ALL-Albanian Lek
DZD-Algerian Dinar
ARS-Argentine Peso
AWG-Aruba Florin
AUD-Australian Dollar
BSD-Bahamian Dollar
BHD-Bahraini Dinar
BDT-Bangladesh Taka
BBD-Barbados Dollar
BZD-Belize Dollar
BMD-Bermuda Dollar
BTN-Bhutan Ngultrum
BOB-Bolivian Boliviano
BWP-Botswana Pula
BRL-Brazilian Real
GBP-British Pound
BND-Brunei Dollar
BIF-Burundi Franc
XOF-CFA Franc (BCEAO)
XAF-CFA Franc (BEAC)

KHR-Cambodia Riel
CAD-Canadian Dollar
CVE-Cape Verde Escudo
KYD-Cayman Islands Dollar
CLP-Chilean Peso
CNY-Chinese Yuan
COP-Colombian Peso
KMF-Comoros Franc
CRC-Costa Rica Colon
HRK-Croatian Kuna
CUP-Cuban Peso
CYP-Cyprus Pound
CZK-Czech Koruna
DKK-Danish Krone
DJF-Djibouti Franc
DOP-Dominican Peso
XCD-East Caribbean Dollar
EGP-Egyptian Pound
SVC-El Salvador Colon
EEK-Estonian Kroon
ETB-Ethiopian

Web services found at this URL:

1 Service Found:

- currencyconvertor

Web reference name:

Add Reference

Cancel

Các bước sử dụng webservice (tt)

<http://www.websvcx.net/country.asmx?wsdl>

Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available services.

Back Home Refresh

URL: Go

"country" Description

Methods

- **GetCountries** () As *string*
Get all countries with ISO code
- **GetCountryByCountryCode** (*CountryCode* As *string*) As *string*
Get country name by country code
- **GetCountryByCurrencyCode** (*CurrencyCode* As *string*) As *string*
Get country by currency code
- **GetCurrencies** () As *string*
Get all currency,currency code for all countries
- **GetCurrencyByCountry** (*CountryName* As *string*) As *string*
Get currency by country name
- **GetCurrencyCode** () As *string*
Get all currencycode for each country
- **GetCurrencyCodeByCurrencyName** (*CurrencyName* As *string*) As *string*
Get currency by currency name
- **GetGMTbyCountry** (*CountryName* As *string*) As *string*
Get greenwich mean time(GMT) by country name
- **GetISD** (*CountryName* As *string*) As *string*

Web services found at this URL:

1 Service Found:
- country

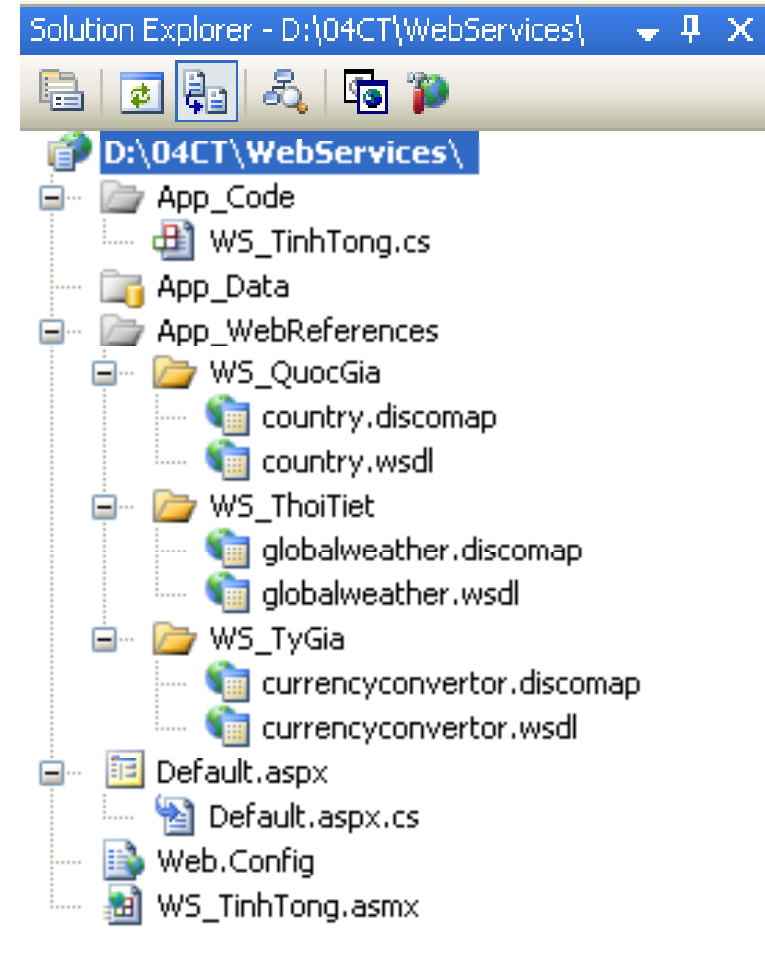
Web reference name:

Add Reference

Cancel

Các bước sử dụng webservice (tt)

Sau khi thêm vào các dịch vụ



Thể hiện các thông tin của webservice

Sử dụng Webservice miễn phí

Tỷ giá ngoại tệ

1 USD = [lbUSD] VNĐ

1 EUR = [lbEUR] VNĐ

1 GBP = [lbGBP] VNĐ

Thời tiết thành phố Hồ Chí Minh

[lbHCM]

Thời tiết thành phố Hà Nội

[lbHaNoi]

Các thành phố trọng điểm của Việt Nam

[lbCities]

Các nước trên thế giới và đơn vị tiền tương ứng

[lbQuocGia]

[lbTien]

Kết nối thông tin hiển thị

```
protected void Page_Load(object sender, EventArgs e)
{
// Hiển thị tỷ giá ngoại tệ
    WS_TyGia.CurrencyConvertor TyGia = new WS_TyGia.CurrencyConvertor();
    lblUSD.Text = TyGia.ConversionRate(WS_TyGia.Currency.USD,
                                        WS_TyGia.Currency.VND).ToString();
    lblEUR.Text = TyGia.ConversionRate(WS_TyGia.Currency.EUR,
                                        WS_TyGia.Currency.VND).ToString();
    lblGBP.Text = TyGia.ConversionRate(WS_TyGia.Currency.GBP,
                                        WS_TyGia.Currency.VND).ToString();

// Hiển thị thông tin thời tiết
    WS_ThoiTiet.GlobalWeather thoitiet = new WS_ThoiTiet.GlobalWeather();
    lblHCM.Text = thoitiet.GetWeather("Ho Chi Minh", "Viet Nam");
    lblHaNoi.Text = thoitiet.GetWeather("Ha Noi", "Viet Nam");

// Hiển thị các thành phố Việt Nam
    lblCities.Text = thoitiet.GetCitiesByCountry("Viet Nam");

// Hiển thị các nước trên thế giới và đơn vị tiền tương ứng
    WS_QuocGia.country quocgia = new WS_QuocGia.country();
    lblQuocGia.Text = quocgia.GetCountries();
    lblTien.Text = quocgia.GetCurrencies();
}
```

Sử dụng Webservice miễn phí

Tỷ giá ngoại tệ

1 USD = 16130 VNĐ

1 EUR = 24896.6621 VNĐ

1 GBP = 31876.2734 VNĐ

Thời tiết thành phố Hồ Chí Minh

Ho Chi Minh, Vietnam (VVTS) 10-49N 106-40E 19M May 02, 2008 - 11:00 AM EDT / 2008.05.02 1500 UTC Variable at 1 MPH (1 KT):0 greater than 7 mile(s):0 mostly cloudy 75 F (24 C) 75 F (24 C) 100% 29.71 in. Hg (1006 hPa) Success

Thời tiết thành phố Hà Nội

Ha Noi, Vietnam (VVNB) 21-01N 105-48E 6M May 02, 2008 - 11:00 AM EDT / 2008.05.02 1500 UTC from the ESE (120 degrees) at 13 MPH (11 KT):0 greater than 7 mile(s):0 partly cloudy 78 F (26 C) 75 F (24 C) 88% 29.65 in. Hg (1004 hPa) Success

Các thành phố trọng điểm của Việt Nam

Viet Nam Da Nang

Viet Nam Ha Noi

Viet Nam Nha Trang

Viet Nam Hue

Viet Nam Pleiku City

Viet Nam Quy Nhon

Viet Nam Ho Chi Minh

Viet Nam Vinh

Các nước trên thế giới và đơn vị tiền tương ứng

Afghanistan, Islamic State of	Afghanistan, Islamic State of af Afghani AFA
Albania	Albania al Lek ALL
Algeria	Algeria dz Dinar DZD
American Samoa	American Samoa as
Andorra, Principality of	Andorra, Principality of ad Franc ADF

Kiểm tra File Format & File Size

```
protected void btnUpload_Click(object sender, EventArgs e)
{
    string[] tokens = FileUploadImage.FileName.Split('.');
    string extension = tokens[tokens.Length - 1].ToLower();
    double size = (double)FileUploadImage.FileBytes.Length / (1024*1024);

    // check whether the uploaded file is an image and file size less than 1MB
    if (size > 1)
        lblMessage.Text = "File size must be less than 1MB";
    else if (extension == "jpg" || extension == "gif")
    {
        // generate relative & absolute path to save the uploaded file
        string strRelativePath = "images/" + FileUploadImage.FileName;
        string strAbsolutePath = Server.MapPath(strRelativePath);
        // save uploaded file
        FileUploadImage.SaveAs(strAbsolutePath);
        lblMessage.Text = strAbsolutePath + " has been saved.";
    }
    else
        lblMessage.Text = "File extension must be jpg or gif.";
}
```

Gửi mail sử dụng Client Mail

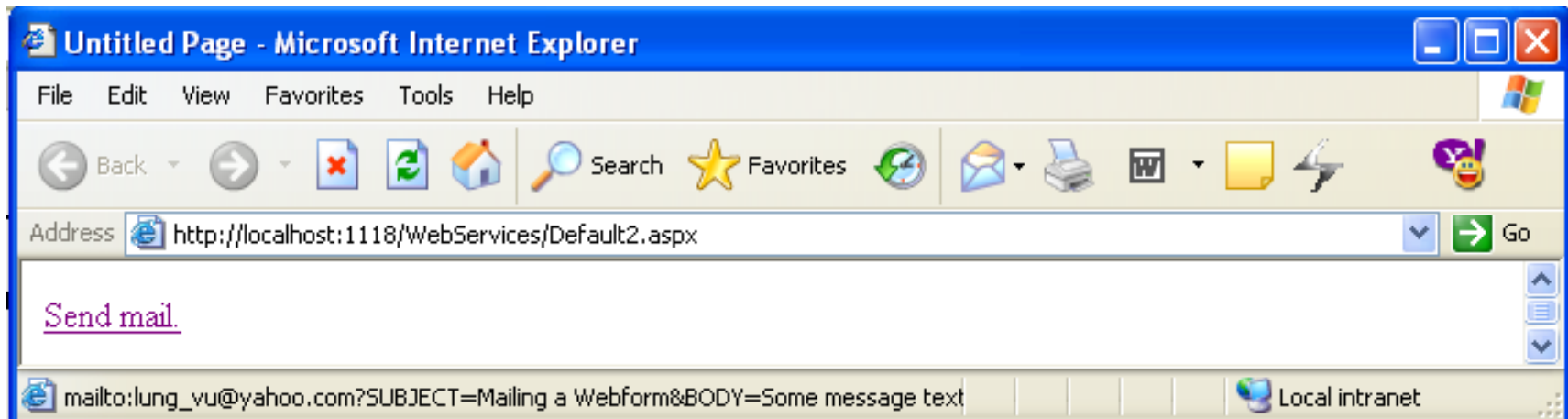
- Use mailto protocol

```
<A href="mailto:abc@yahoo.com?SUBJECT=Question&BODY=Answer">
```

- Use Hyperlink server control

```
<asp:HyperLink ID="hypMail" NavigateUrl=  
    "mailto:lung_vu@yahoo.com?SUBJECT=Mailing a Webform&BODY=Some  
    message text." Runat="server"> Send mail.
```

```
</asp:HyperLink>
```



Gửi mail sử dụng Client Mail (tt)

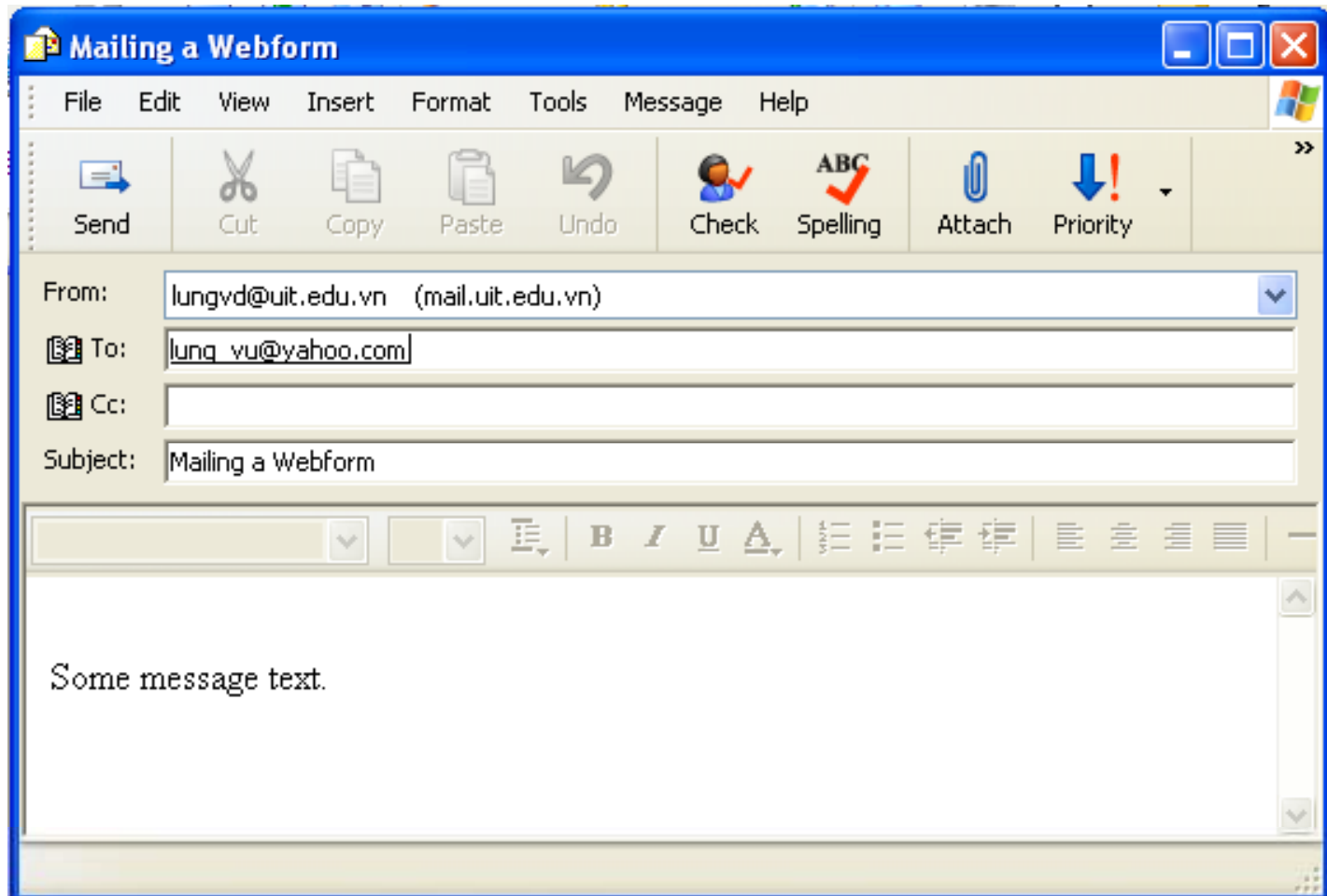
- Format:

mailto:address[?key=setting][&key=setting] ...

<i>Key</i>	<i>ý nghĩa</i>
SUBJECT	Tiêu đề Email
CC	Danh sách địa chỉ Email người nhận, mỗi địa chỉ cách nhau bằng dấu chấm phẩy (;).
BCC	BCC Danh sách địa chỉ Email người nhận được ẩn mỗi địa chỉ cách nhau bằng dấu chấm phẩy (;).
BODY	Nội dung Email.

Gửi mail sử dụng Client Mail (tt)

- ❑ Kết quả sẽ đưa vào Client Mail default



Gửi Email sử dụng Remote Server

- ❑ Soạn nội dung email sử dụng MailMessage
 - From, To
 - Subject, Body
- ❑ Cấu hình Remote Server sử dụng SmtpClient
 - Host, Port
 - Credentials
 - ...
- ❑ Gửi email sử dụng phương thức SmtpClient.Send

Gửi Email sử dụng Remote Server (tt)

```
using System.Net.Mail;
protected void btnSendMail_Click(object sender, EventArgs e) {
    // Create the message.
    MailMessage mailNew = new MailMessage();
    mailNew.From = new MailAddress("someone@microsoft.com", "name");
    mailNew.To.Add("jesse@contoso.com");
    mailNew.Subject = "This is the subject text.";
    mailNew.Body = "This is the message text.";
    Attachment atcltem = new Attachment(Server.MapPath("joey.jpg"));
    mailNew.Attachments.Add(atcltem);

    // Create SMTP Client.
    SmtplibClient smtp = new SmtplibClient();
    smtp.Host = "smtp.gmail.com";
    smtp.Port = 465;
    smtp.EnableSsl = true;
    smtp.Timeout = 100;
    smtp.Credentials = new System.Net.NetworkCredential("username", "password");

    // Send the message.
    smtp.Send(mailNew);
}
```


Bài báo cáo

- ❑ Nộp bài và báo cáo vào tuần 15
- ❑ Báo cáo giấy: 10-20 trang
 - Giới thiệu nhóm
 - Giới thiệu mục đích của web site
 - Giới thiệu các trang web
 - Mô tả CSDL, mối liên kết giữa các bảng,...
 - Quản lý web site
 - Kết luận
- ❑ Báo cáo Seminar
 - Thời gian báo cáo: mỗi nhóm báo cáo + hỏi khoảng 10'-15'
 - Chạy demo web site
 - Giới thiệu các trang và các kỹ thuật đã sử dụng

Thi

- Hình thức: Thi trắc nghiệm
- Thời gian : 60'
- Số câu 60
- Thi trực tiếp trên máy

Chương VII: THIẾT KẾ TỔNG QUAN

- ❑ Phân tích các thành phần của website
- ❑ Thiết kế cấu trúc website
- ❑ Xây dựng các phần dùng chung:
 - Namespace
 - Class
 - User Control
 - JavaScript Function
 - Masterpage
- ❑ Xây dựng các trang đơn lẻ
- ❑ Quản lý và bảo hành web

Phân tích các phần của website

1. Phân tích ý tưởng, mục tiêu => Hoạch định cấu trúc website
2. Thiết kế CSDL
3. Xây dựng các phần chung của website:
 - lớp xử lý (namespace, class),
 - Các đối tượng thể hiện giao diện (user control)
 - Các định dạng css dùng chung
 - Các hàm xử lý phía client (Javascript)
 - Các mẫu trang masterpage
4. Xây dựng các trang đơn lẻ

Ví dụ cấu trúc website

Address: <http://localhost:3828/CECE/Default.aspx> Go Links

KHOA KỸ THUẬT MÁY TÍNH
Faculty of Computer Engineering
University of Information Technology

Khu phố 6, phường Linh Trung - Q. Thủ Đức
34 Trương Định - Quận 3 - TP. HCM
Tel: (08) 8974500 - Fax: (08) 9304900
Email: Ce@uit.edu.vn

Thứ Bảy Ngày 9 Tháng 5 Năm 2009 | Trang chủ | Diễn Đàn | Liên hệ | Sign-in

- Trang Chủ
- Giới thiệu
- Tổ Chức
- Thông tin-Thông báo
- Chương trình đào tạo
- Nghiên Cứu Khoa học
- Hợp Tác

Thư viện điện tử

- Phần mềm
- Tiện ích (2)
- Duyệt Virus (0)
- Sách điện tử
- Kỹ thuật máy tính (0)
- Sách lập trình (0)
- Mạng - Internet (0)
- Cơ sở dữ liệu (0)
- Mã nguồn
- DotNet (0)
- Visual basic (0)
- PHP - MySQL (0)
- C, C++ (0)

Lượt truy cập
5400

- Giới Thiệu
- Thông báo
- Lịch thi
- Sản phẩm mới
- Kinh nghiệm
- Hacker & Virut
- Games - Giải trí

Thông Báo Thứ Nghiệm (05/09/09)
Thứ nghiệm website hoạt động...

Chu De Thong Bao (05/09/09)
chu de thong bao chu de thong bao chu de thong bao chu de thong bao chu de thong bao chu de thong ...

Thông Báo Website Hoat Dong (05/08/09)
thông báo website hoat dong thông báo website hoat dong thông báo website hoat dong thông báo websi...

Thông báo mới nhất

- Thông báo thử nghiệm (05/09/09)
- chu de thong bao (05/09/09)
- thông báo website hoat dong (05/08/09)

Hỗ trợ trực tuyến

Not online
Not online

Liên kết

DANH BẠ PHẦN MỀM VIỆT NAM
www.phanmemvietnam.com

edu.Net
MẠNG GIÁO DỤC

Nhận xét cho một website

❑ Đối với mỗi trang web hầu như đều cần:

- Có giao diện giống nhau,
- Có chuỗi kết nối đến CSDL,
- Thực hiện đọc bảng dữ liệu,
- Các xử lý liên kết dữ liệu,
- Các xử lý khác...

❑ Số lượng xử lý các thao tác tăng => số lượng dòng lệnh càng nhiều

Khi để các đoạn code rải rác trong website gây khó khăn:

- Xây dựng ứng dụng web
- Phát triển ứng dụng web
- Bảo trì ứng dụng

=> Cần có các lớp xử lý và đối tượng thể hiện dữ liệu chung

Không gian tên miền, các lớp chung

□ Định dạng chung

namespace TênNameSpase

{

#region “--- Tên phân vùng 1 ---”

// Các lớp, biến, thuộc tính, hàm

#endregion

#region “--- Tên phân vùng 2---”

// Các lớp, biến, thuộc tính, hàm

#endregion

.....

}

```
using System;  
using System.Data;  
using System.Data.SqlClient;  
using TênNameSpace;
```

Ví dụ không gian tên miền

```
namespace DaoTao.Data
{
    public class SqlData
    {
        | "-----Common-----"
        "==== Chuon trinh trac nghiem ====="
        "==== thu vien ====="
        "==== News ====="
        "-----Username-----"
        "-----Thu hoc vien -----"
        "====Lop hoc===="
        "====Mon hoc===="
        "====Phong hoc===="
        "====Giaovien===="
        "====Thoi Khoa Bieu ====="
        "==== Thong ke website ====="

        }//end class
    }//end Namespace
```


Ví dụ Class chứa các hàm dùng chung

```
namespace DaoTao.Components {  
    public class Globals  
    {  
        static public String GetSkinPath()...  
        static public String GetUrlNewsImages()...  
        static public String GetUrlImages()...  
        static public String GetUrlImages_IconFormPost()...  
        static public String GetLangPath()...  
        static public string ApplicationPath ...  
    }  
}
```

Ví dụ một số hàm dùng chung

```
static public String GetUrlImages()
{
    try
    {
        return ApplicationPath + "\\Themes\\Images\\";
    }
    catch { return ""; }
}
static public string ApplicationPath
{
    get {
        string applicationPath =
        HttpContext.Current.Request.ApplicationPath;

        if (applicationPath == "/") { return string.Empty; }
        else { return applicationPath; }
    }
}
```

Cách sử dụng biến global và hàm chung

- ❑ Link đến một trang web nằm ở ngoài cùng của thư mục chứa website

```
<a id="menuleft"  
href="<%=Globals.ApplicationPath%>/Default.aspx">  
Trang Chủ</a>
```

- ❑ Đường dẫn đến file hình ảnh

```

```

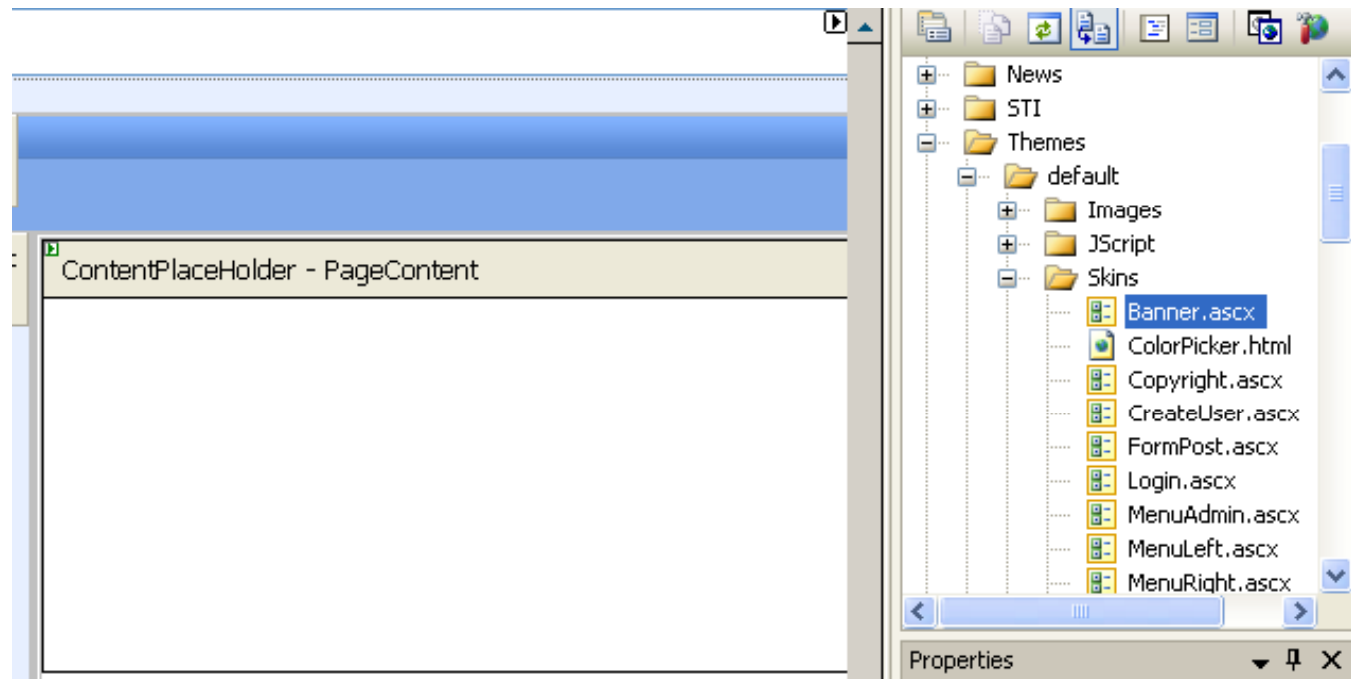
Các đối tượng thể hiện giao diện (user control)

```
<%@ Control Language="C#" %>  
<%@ Import Namespace="DaoTao.Components" %>  
<table style="border-color:#6699cc" width="100%" cellpadding="0" cellspacing="0"  
  class="borderbanner" align="center" border="0">  
<tr>  
  <td align="left" bgcolor="white" style="padding-left:10pt;"></td>  
  <td align="right" bgcolor="white" style="width: 582px"></td>  
</tr>  
</table>
```



Sử dụng User control

```
<%@ Register Src="default/Skins/Banner.ascx" TagName="Banner"
TagPrefix="uc1" %>
<tr>
  <td>
    <uc1:Banner ID="Banner1" runat="server" />
  </td>
</tr>
```

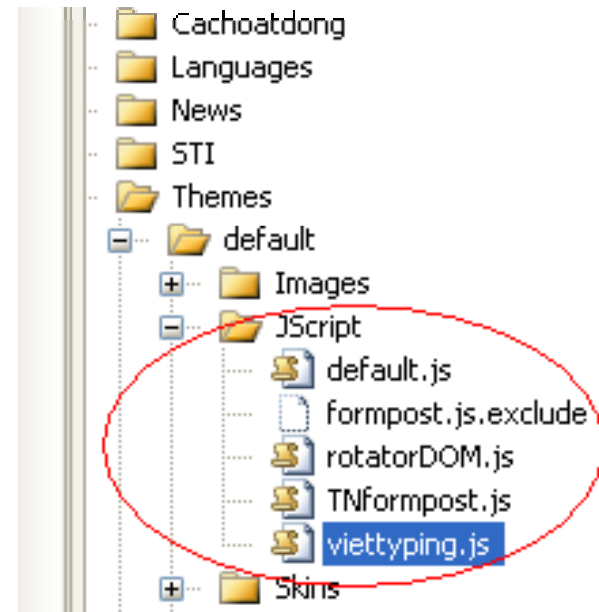


Sử dụng User control

The screenshot shows a web browser window displaying a website for the Faculty of Computer Engineering at the University of Information Technology. The browser's address bar shows the URL `http://localhost:3828/CECE/Default.aspx`. The website header includes the faculty name in Vietnamese and English, along with contact information. A navigation menu on the left lists various categories, with 'Thư viện điện tử' (Digital Library) highlighted. The main content area features several sections: 'Giới Thiệu' (Introduction), 'Thông báo' (Announcements) with sub-sections like 'Thông Báo Thử Nghiệm' and 'Chu De Thong Bao', 'Lịch thi' (Exam Schedule) with 'Thong Bao Mon TKM', and 'Sản phẩm mới' (New Products). A right sidebar contains 'Thông báo mới nhất' (Latest Announcements), 'Hỗ trợ trực tuyến' (Live Support) with 'Not online' buttons, and 'Liên kết' (Links) to external sites like 'DANH BẠ PHẦN MỀM VIỆT NAM' and 'edu.Net MẠNG GIÁO DỤC'. Five red circles are overlaid on the page: (1) on the header, (2) on the navigation menu, (3) on the 'Tổ Chức' (Organization) link, (4) on the 'Lịch thi' section, and (5) on the 'Thông báo mới nhất' section.

Các hàm xử lý phía client (Javascript)

- ✓ Các hàm PopUp
- ✓ Các hàm xử lý kiểm tra phía Client
- ✓ Các hàm xử lý gõ tiếng Việt
- ✓



Ví dụ xây dựng lớp xử lý dữ liệu chung

```
9 | using System.Web.UI.HtmlControls;
10 | namespace Data.TableProcess
11 | {
12 |     /// <summary>
13 |     /// Summary description for DataProcessing
14 |     /// </summary>
15 |     public class DataProcessing:DataTable
16 |     {
17 |         #region "---- Khai bao bien thanh vien ----"
18 |
19 |         #endregion
20 |
21 |         #region "---- Danh sach cac thuoc tinh ----"
22 |
23 |         #endregion
24 |
25 |         "---- Nhóm hàm khởi tạo đối tượng ----"
33 |
34 |         "---- Nhóm hàm cung cấp thông tin ----"
37 |         "---- Nhóm hàm xử lý tính toán ----"
40 |
41 |         "---- Xử lý sự kiện ----"
44 |
45 |     } // End class "DataProcessing"
46 | } // End namespace "Data.TableProcess"
```


Khai báo biến thành viên

```
18 #region "---- Khai bao bien thanh vien -----"
19 // Choi ket noi den CSDL
20     private string mConnStr = ""; // "User ID;pwd=";
21 // Chuoi lenh SQL
22     private string mSQL_String = "";
23 // Ten Bang muon truy van
24     private string mTableName;
25 // Doi tuong ket noi SqlConnection
26     SqlConnection mConn;
27 //Doi tuong truy xuat cap nhat du lieu Adapter
28     SqlDataAdapter mBoDocGhi;
29 // Nguon du lieu
30     private string mDataPath="";
31     private string errorMessage = "";
32     private int errorCode = 0;
33 #endregion
34
35 "---- Danh sach cac thuc tinh -----"
--
```

Danh sách các thuộc tính

```
18 | "---- Khai bao bien thanh vien -----"  
34 |  
35 | #region "---- Danh sach cac thuoc tinh -----"  
36 | public string ConnectionString  
37 | {  
38 |     set { mConnStr = value; }  
39 |     get { return mConnStr; }  
40 | }  
41 | public string ErrorMessage  
42 | {  
43 |     set { errorMessage = value; }  
44 |     get { return errorMessage; }  
45 | }  
46 | public int ErrorCode  
47 | {  
48 |     set { errorCode = value; }  
49 |     get { return errorCode; }  
50 | }  
51 | #endregion  
52 |  
53 | "---- Nhóm hàm khởi tạo đối tượng -----"  
71 |  
72 | "---- Nhóm hàm cung cấp thông tin -----"
```

Nhóm hàm khởi tạo đối tượng

```
#region "---- Nhóm hàm khởi tạo đối tượng ----"
public DataProcessing()
{
    //
    // TODO: Add constructor logic here
    //
    mConnStr = " ";           // "server=(local);"database=
                               // "Integrated Security=SSPI;

    mSQL_String = "";
    mTableName = "";
    mDataPath = "";

}
public DataProcessing(string connStr)
{
    mConnStr = connStr;
}
#endregion

"---- Nhóm hàm cung cấp thông tin ----"
```

Nhóm hàm cung cấp thông tin

```
#region "---- Nhóm hàm cung cấp thông tin ----"
public DataTable DocBang(string SQL_String)
{
    DataTable bang = new DataTable();
    OpenConnection();
    mBoDocGhi = new SqlDataAdapter(SQL_String,mConn);
    mBoDocGhi.Fill(bang);
    CloseConnection();
    return bang;
}
#endregion
```

Một số hàm xử lý

```
#region "---- Nhóm hàm xử lý ----"
void OpenConnection()
{
    mConn = new SqlConnection(mConnStr);
    mConn.ConnectionString = mConnStr;
    mConn.Open();
}
void CloseConnection()
{
    if (mConn.State != ConnectionState.Open)
    {
        mConn.Close();
    }
    mConn.Dispose();
}
#endregion
```

Sử dụng hàm dùng chung

➤ File default.aspx

```
9 | <body>
10 |     <form id="form1" runat="server">|
11 |     <table>
12 |         <asp:Repeater ID="rptThuNghiem" runat="server">
13 |         <ItemTemplate>
14 |             <tr>
15 |                 <td> <%# Eval("UserId") %></td>
16 |                 <td> <%# Eval("UserName") %></td>
17 |             </tr>
18 |
19 |         </ItemTemplate>
20 |     </asp:Repeater>
21 | </table>
22 |
23 | </form>
24 | </body>
25 | </html>
```

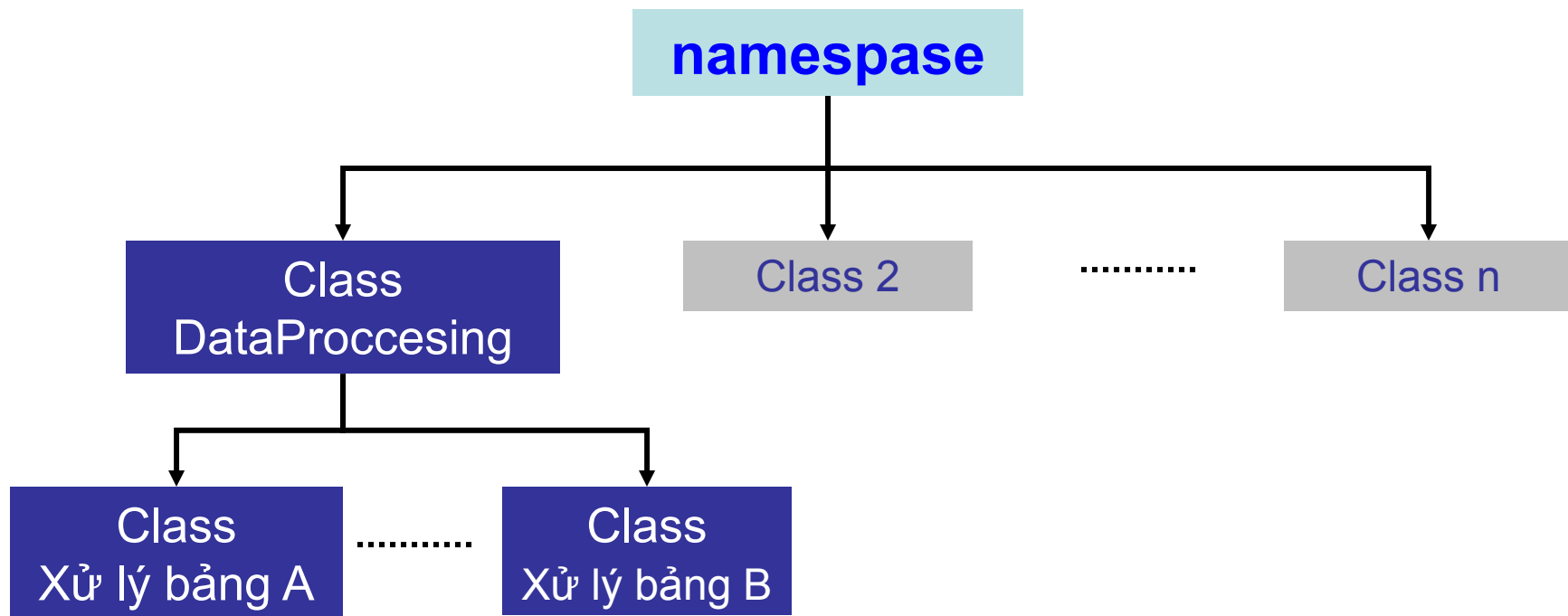
Sử dụng hàm dùng chung

➤ File default.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=.\SQLEXPRESS;" +
        "AttachDbFilename=|DataDirectory|"
    + "\\ASPNETDB.MDF;Integrated Security=True;User Instance=True";
    string SqlStr = "SELECT [UserId], [UserName] FROM [vw_aspnet_Users]";
    DataTable tblData = new DataTable();
    if (!IsPostBack)
    {
        DataProcessing mDataProc = new DataProcessing(ConnectionString);
        tblData = mDataProc.DocBang(SqlStr);
        rptThuNghiem.DataSource = mDataProc.DocBang(SqlStr);
        rptThuNghiem.DataBind();
    }
}
```

Xây dựng các lớp khác

- ❑ Các lớp xử lý nghiệp vụ cho từng bảng riêng được kế thừa từ lớp xử lý dữ liệu chung



THI

- Hình thức: Thi trắc nghiệm
- Thời gian : 60'
- Số câu 60
- Thi trực tiếp trên máy