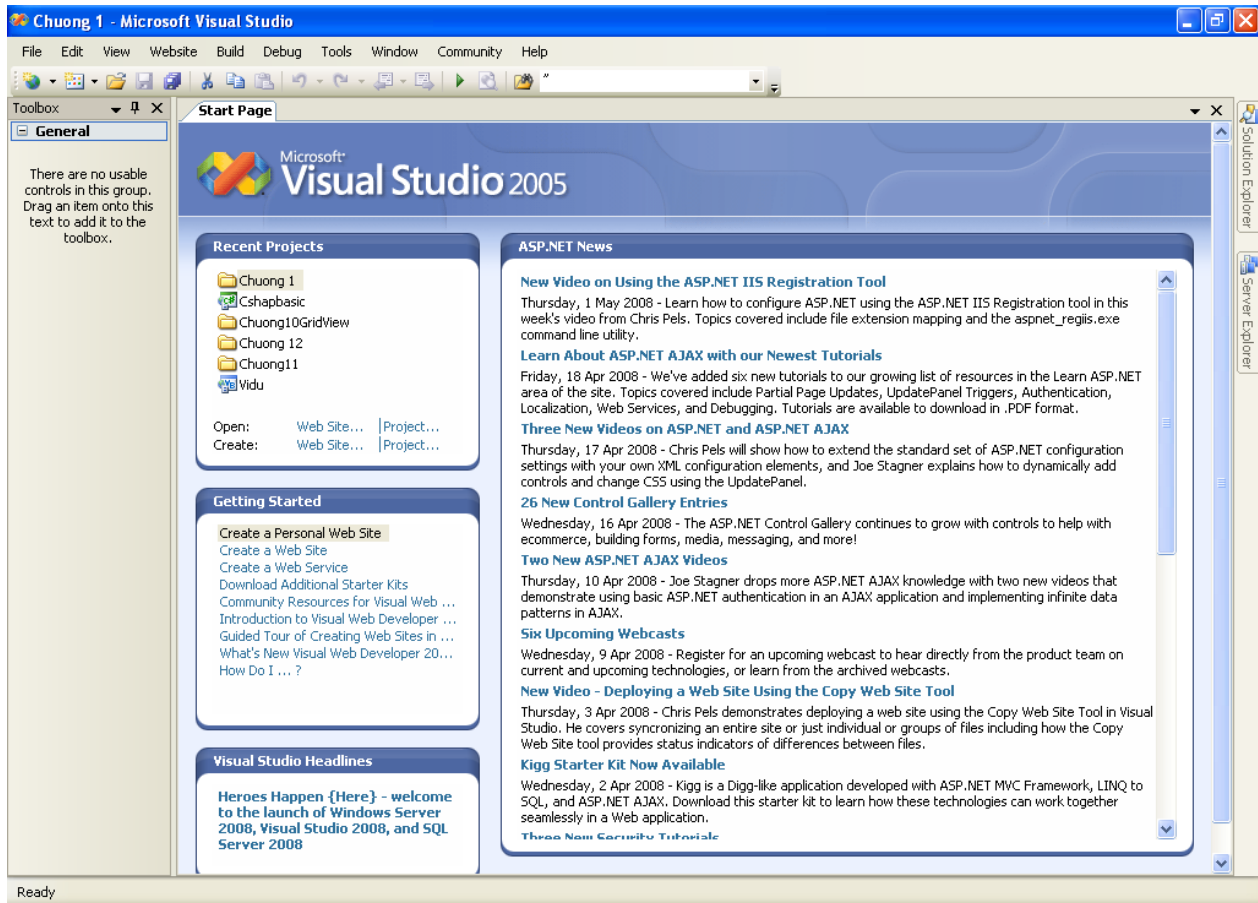


Chương 1. Giới thiệu chung về cấu trúc ASP.NET Framework và cơ bản về C#

I. Giới thiệu chung về ASPNetFramework

Trong giáo trình này chúng ta sẽ học ASP.NET trên IDE VisualStudio2005(Bạn có thể sử dụng Visual Web Developer 2005).

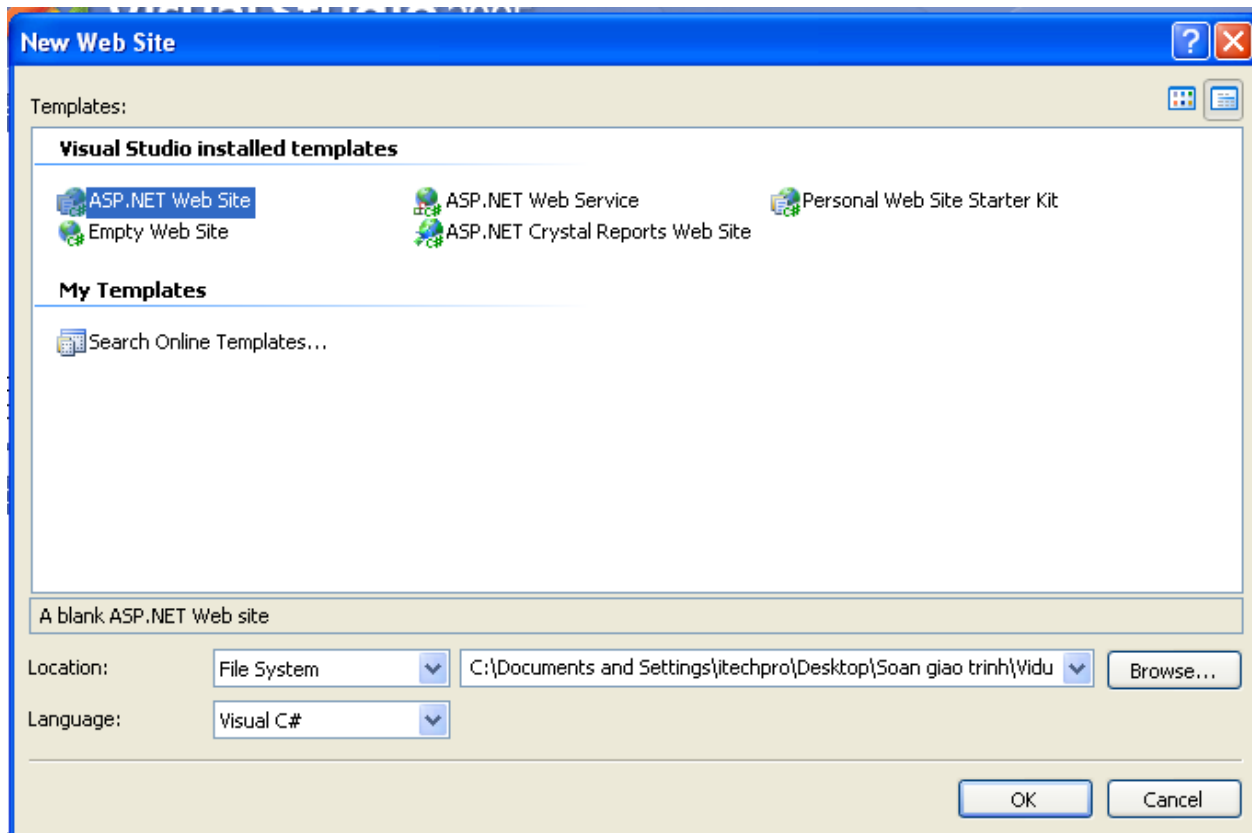
Để tạo một Website mới bạn khởi động VS. giao diện của nó sẽ hiện ra như sau:



Hình 1

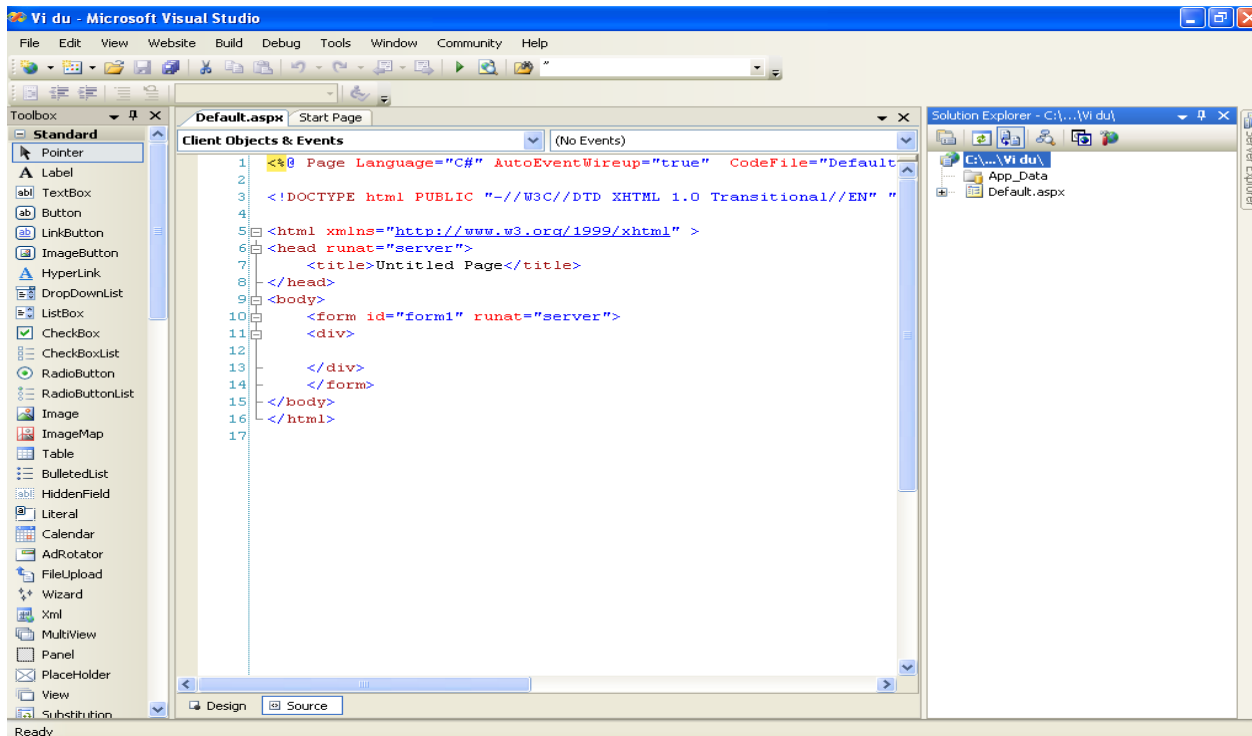
Trong Box Recent Project bạn chọn "Web site..." ở dòng Create

Hộp thoại New Website hiện ra bạn chọn ASP.NET Website rồi chọn thư mục bạn để Website và ngôn ngữ kịch bản để bạn viết Web và nhấn OK



Hình 2

VS sẽ tạo ra một website với tên của Website là tên bạn vừa đặt và mặc định sẽ có một trang Default.aspx như hình sau:



Hình 3

1. ASP.NET và .NET FRAMEWORK

ASP.NET là một phần của .NET Framework Để xây dựng trang asp.NET bạn cần thêm vào các đặc tính của .netframework. NetFramework chứa đựng hai phần Framework Class Library và Common Language Runtime.

1.1 Hiểu về Framework Class Library

Framework chứa đựng hàng nghìn lớp mà bạn có thể sử dụng trong ứng dụng của mình. Ví dụ một vài lớp của .Net Framework

-Lớp File: cho phép bạn tạo file, sửa, xoá hay kiểm tra sự tồn tại của file trên đĩa cứng...

-Lớp Graphics: Cho phép bạn làm việc với nhiều kiểu của ảnh, bạn cũng có thể tạo ra các ảnh từ các phương thức trên lớp này.

-Lớp Smtplib: Cho phép bạn gửi thư.

Hiểu về Namespaces: hơn 13 nghìn lớp trong Netframework. Đây là một con số rất lớn, Microsoft đã chia các lớp cùng xử lý về một vấn đề gì đó vào các không gian tên chung hay namespaces.

Một Namespace đơn giản là một danh mục, ví dụ tất cả các lớp thao tác với File và thư mục chúng ta đưa vào một namespaces chung gọi là System.IO, hay tất cả các lớp làm việc với SqlServer có thể đưa vào namespace System.Data.SqlClient.

Các namespaces chung nhất trong net:

- . System
- . System.Collections
- . System.Collections.Specialized
- . System.Configuration
- . System.Text
- . System.Text.RegularExpressions
- . System.Web
- . System.Web.Caching
- . System.Web.SessionState
- . System.Web.Security
- . System.Web.Profile
- . System.Web.UI
- . System.Web.UI.WebControls
- . System.Web.UI.WebControls.WebParts

1.2 Hiểu về Assembly:

Một Assembly là một file dll trên đĩa cứng của bạn, nơi mà lưu trữ các lớp của .NET, ví dụ tất cả các lớp trong .ASP.NET Framework đều nằm trong Assembly System.web.dll.

Trước khi sử dụng các lớp trong dll bạn cần tạo một tham chiếu đến file dll này

1.3 Hiểu về Common Language Runtime(CLR)

Phần thứ 2 của NetFramework là CLR chịu trách nhiệm về thực thi mã ứng dụng của bạn.

Khi bạn viết ứng dụng bằng ngôn ngữ C#, VB.NET hay bằng một ngôn ngữ bất kỳ trên nền NetFramework mã của bạn sẽ được không bao giờ biên dịch trực tiếp thành mã máy. Thay vào đó chúng được biên dạng sang ngôn ngữ đặc tả MSIL (Microsoft intermediate Language).

MSIL nhìn rất giống với ngôn ngữ hướng đối tượng Assembly, nhưng không giống kiểu ngôn ngữ Assembly. MSIL là ngôn ngữ bậc thấp và phụ thuộc vào Platform.

Khi ứng dụng của bạn thực thi, mã MSIL là "just in time" biên dịch sang mã máy bởi JITTER(just in time compiler)

Như vậy khi bạn viết các lớp trên .Net bằng bất kỳ ngôn ngữ nào khi bạn biên dịch sang Assembly bạn đều có thể sử dụng Assembly đó cho các ngôn ngữ khác.

1.4 Hiểu về các điều khiển trên Asp.net

Các điều khiển asp.net là phần quan trọng nhất trong ASP.NET Framework. một Control ASP.NET là một lớp mà thực thi trên server và đưa ra nội dung trên trình duyệt. ASP.NET có hơn 70 control mà bạn có thể sử dụng trong xây dựng ứng dụng web của bạn và cơ bản nó chia ra các nhóm control sau:

Standard control: bao gồm các điều khiển đưa ra các thành phần chuẩn của form như: Label, Button, TextBox...

Validator Control: là các control cho phép bạn kiểm tra tính hợp lệ của các control cho phép nhập giá trị trên form.

Rich Control: là những điều khiển như FileUpload, Calendar...

Data Control là các điều khiển cho phép thao tác với dữ liệu

Navigation Control: là những điều khiển giúp bạn dễ dàng di chuyển giữa các trang trong website.

Login control: Là các điều khiển về bảo mật của ứng dụng cho phép bạn đưa ra các form đăng nhập, thay đổi mật khẩu...

HTML Control: cho phép bạn chuyển các điều khiển của HTML thành các điều khiển có thể làm việc trên server.

1.5 hiểu về điều khiển sự kiện trên server

phần lớn các điều khiển của asp.net hỗ trợ 1 hoặc nhiều sự kiện, ví dụ điều khiển ASP.NET Button hỗ trợ sự kiện Click, khi người sử dụng nhấn chuột vào Button một sự kiện sẽ được đưa ra và công việc này được xử lý trên server.

Ví dụ: Trang UnderstandEvent.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="UnderstandEvent.aspx.cs" Inherits="UnderstandEvent" %>
<script runat="server">
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = TextBox1.Text;
    }
</script>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Hiểu về sự kiện phía server</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <asp:Button ID="Button1" OnClick="Button1_Click"
        runat="server" Text="Button" />
      <hr />
      <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    </div>
  </form>
</body>
</html>
```

Trong ví dụ trên gồm 3 điều khiển của ASP.NET là TextBox, Label, và Button, mỗi khi người sử dụng nhập dữ liệu vào Textbox và nhấn vào Button sự kiện Button1_Click được đưa ra và điền dữ liệu từ TextBox và Label.

1.6 Hiểu về View State

Giao thức http là giao thức nền móng của WWW, là một giao thức chuẩn thực tế. mỗi lần bạn request một trang từ website, một dữ liệu mới được đưa ra, ASP.NET Framework có thể quản lý được vượt ra ngoài giới hạn của giao thức http, ví dụ bạn điền dữ liệu vào một điều khiển Label với thuộc tính Text của nó, dữ liệu này sẽ được lưu trữ qua nhiều trang web và chỉ thay đổi khi nó được gán lại giá trị.

Ví dụ sau sẽ đưa một một trang asp.net trong đó gồm 2 điều khiển Button và Label(Text của nó hiển thị số đếm), mỗi lần nhấn vào Button thì giá trị của Label tăng lên 1.

Ví dụ trang Understandstate.aspx:

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
  protected void Button1_Click(object sender,EventArgs e)
  {
    Label1.Text = Convert.ToString(int.Parse(Label1.Text) + 1);
  }
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Under Stand State</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="Label1" runat="server" Text="0"></asp:Label>
      <asp:Button ID="Button1" OnClick="Button1_Click" runat="server"
Text="Button" />
    </div>
  </form>
</body>
```

```
</html>
```

Nếu bạn mở View Source code của trang Understandstate.aspx trên bạn sẽ thấy như sau:

```
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwULLTE4OTg4OTc0MjUPZBYCAgQPZBYCAgEPDxYCHgRUZXh0BQEzZGRkz0aTZTJffZRUP11aiDXbPGQGitk=" />
<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWAqKK7LjKBwKM54rGBgzCWqbCIzq33uVXHF19FzfdGPsJ" />
```

Đây là 2 file hidden trong form nó chứa giá trị Text của Label khi trang được postback nó sẽ ghi nhớ text đó và sẽ khởi tạo lại giá trị của Label khi trang Load.

1.7 Hiểu về trang asp.net

Sử dụng Code-Behind

Thay vì sử dụng `<script runat="server">` `</script>` ngay trên trang asp.net. người ta đưa ra thêm một trang gọi là Code-behind chứa các mã lệnh thực hiện trên trang asp.net.

Ví dụ như phần đầu của chương khi ta tạo ra một website thì mặc định sẽ tạo ra một lớp Default.aspx và nó sẽ kèm theo một trang Default.aspx.cs

Trang Default.aspx

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

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Default</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>

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

Trang Default.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
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)  
    {  
  
    }  
}
```

1.8 Điều khiển sự kiện của trang asp.net

Khi chạy trang asp.net thì vòng đời của trang asp.net gồm các sự kiện

1. PreInit
2. Init
3. InitComplete
4. PreLoad
5. Load
6. LoadComplete
7. PreRender
8. PreRenderComplete
9. SaveStateComplete
10. Unload

Sử dụng thuộc tính Page.IsPostBack

Với sự kiện Load của trang thì khi tải trang lên thì có một sự kiện nào đó được đưa ra, nếu có nghĩa mỗi lần load lại trang nó lại thực hiện công việc đó, còn nếu ta đưa thêm vào thuộc tính Page.IsPostBack thì ta có thể điều khiển được sự kiện nào được thực hiện và sự kiện nào không khi trang được tải lại.