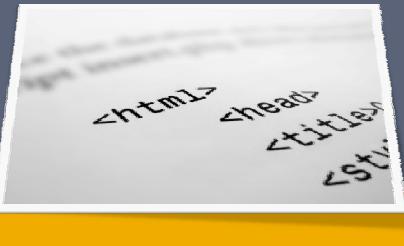


## ASP.NET WITH C# CLASS 1

อ.ณัฐภรณ์ แก้วรัตนภรณ์



## Course's Overview



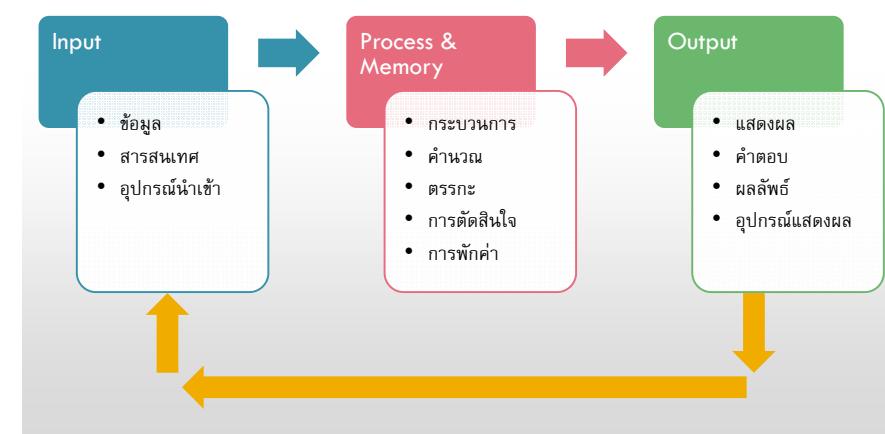
- 45 hours
- ระบบ A2A โดยเป็นระบบ  
แลกเปลี่ยนข้อมูลระหว่าง  
เว็บ tourism และ เว็บ  
Hotel โดยใช้ Web  
Services

## TODAY! Algorithm & Basic Web Programming

- วางแผน ความเข้าใจ ในการบูรณาการผลิตซอฟต์แวร์ หรือ แอ็พพลิเคชัน
- โครงสร้างของเว็บแอ็พพลิเคชัน และ Server Side
- Visual Studio 2005 เปื้องต้น

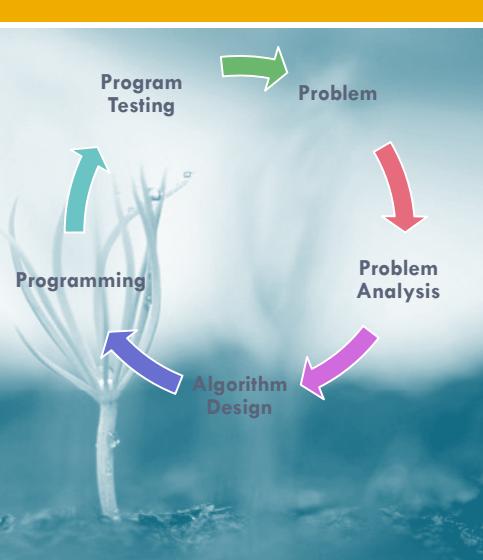


## Computer Organization



# What's Programming

- การเขียนโปรแกรม (Programming) คือ การใช้คำสั่งให้เครื่องคอมพิวเตอร์ทำงานหรือคำนวณตามลำดับคำสั่งอย่างเป็นระบบ โดยการเขียนโปรแกรม จะต้องผ่านกระบวนการวิเคราะห์อีกครึ่งก่อนเสมอ เพื่อแสดงผลลัพธ์ที่มีความถูกต้องและใช้เวลาในการประมวลผลน้อยที่สุด ซึ่งจะทำให้เกิดประสิทธิภาพและประสิทธิผลในผลงาน



## Question

- จงเขียนอักษรที่มีการอ่านน้ำ้ โดยมีอุปกรณ์ดังนี้
    - ผ้าขนหนู
    - ผักบัว
    - สบู่
    - ยาสระผม
    - แปรงสีฟัน
    - ยาสีฟัน



# What's Algorithm

- อัลกอริทึม คือ ขั้นตอนในการแก้ไขปัญหาอย่างมีขั้นมีตอน ซึ่งจะมีเป้าหมายและมีแนวทางในการปฏิบัติอย่างชัดเจน เพื่อวัดดูประสิทธิภาพของการแก้ปัญหาและความต้องการ
  - การวิเคราะห์อัลกอริทึมมีหลายวิธี แต่วิธีที่นิยม คือการเขียน ผังงาน (Flow Chart) และโค๊ดเทียม (Pseudo Code)



## Flowchart : Flow + Chart

- เป็นเครื่องมือในการอธิบายขั้นตอนการประมวลผลโดยใช้สัญลักษณ์ในการแสดงความหมายหรือกำหนดลำดับการทำงาน
  - ข้อดี คือ ทำให้เข้าใจง่าย เป็นระบบที่ชัดเจน
  - ข้อเสีย คือ การใช้รูปสัญลักษณ์อธิบายการทำงานที่ซับซ้อน จะต้องใช้เนื้อที่เยอะ
  - ดังนั้น ผังโปรแกรม (**Flow chart**) จึงเหมาะสมกับผู้เริ่มต้น เรียนโปรแกรม

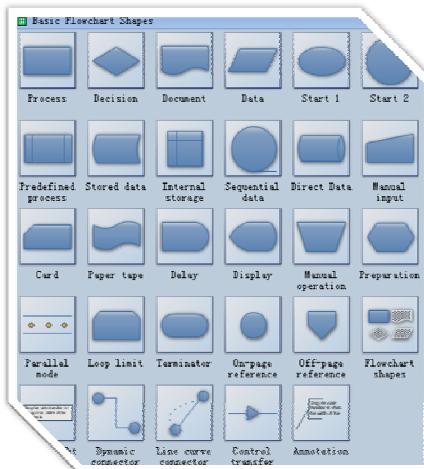
## Flowchart Structure



- **Sequence Structure** เป็นการทำงานแบบเรียงลำดับ (**Sequential**) ตั้งแต่ต้นจนจบ เป็นรูปแบบง่ายๆ ไม่มีการเปรียบเทียบใดๆ มีพื้นที่ทางการให้ของข้อมูลเพียงทางเดียว อาจเป็นแบบนี้ไปล่าง หรือ ซ้ายไปขวา ก็ได้
- **Decision Structure** เป็นการทำงานแบบมีการตัดสินใจหรือเงื่อนไข (**Condition**) เพื่อให้เลือกทำงานต่อ ถ้าหากเลือกได้ก็จะไปทำงานในเงื่อนไขที่เลือก
- **Iteration Structure (Loop)** การทำงานรูปแบบนี้คือการทำงานซ้ำๆ หลายครั้ง หรืออาจทำครั้งเดียว และการทำงานแบบนี้จะต้องมีเงื่อนไขในการทำซ้ำเพื่ออกจาก Loop ด้วย เช่นกัน

## Flowchart Symbols

- สัญลักษณ์ของ Flowchart แบ่งเป็น 3 ประเภท
  - สัญลักษณ์พื้นฐาน (**Basic Symbols**)
  - สัญลักษณ์ระบบ (**System Symbols**)
  - สัญลักษณ์การโปรแกรม (**Programming Symbols**)

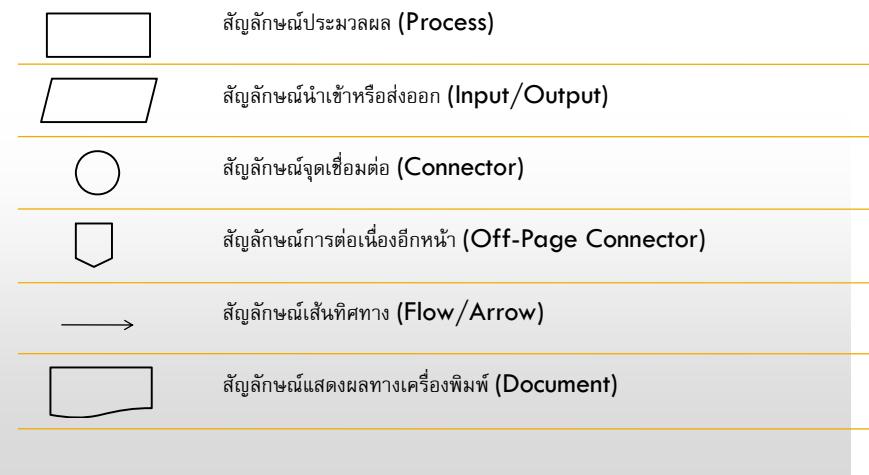


## Why , Algorithm & Flowchart

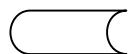


- เพื่อแสดงลำดับขั้นตอนต่างๆ ที่ใช้ในการแก้ปัญหา
- เพื่อให้เราสามารถตรวจสอบความถูกต้องได้ เพื่อให้การแก้ปัญหามีประสิทธิภาพ
- ใช้เป็นสื่อกลางในการบอกต่อวิธีการแก้ปัญหา
  - คิดแก้ปัญหาแต่ไม่ได้ลำดับการทำงานไว้ จะเกิดอะไร?
  - คิดแก้ปัญหานิ่นแล้ว แต่ไม่ได้เขียนบันทึกไว้ อีก 2 อาทิตย์ ลืม ทำไง?
  - คิดแก้ปัญหาไว้ แต่จะบอกไปให้โปรแกรมเมอร์เข้าใจ?
  - คิดแก้ปัญหาไว้ แต่ระบบใหญ่มาก จะทำไง?
  - คิดแก้ปัญหาไว้ แต่ทำไม่คนอื่นลำดับขั้นตอนแก้ยากกว่า?

## Flowchart Symbols Basic Symbols



## Flowchart Symbols System Symbols



สัญลักษณ์แทนสื่อจดเก็บข้อมูลแบบออนไลน์ (Online Storage)



สัญลักษณ์แทนสื่อจดเก็บข้อมูลแบบดิจิตัส (Magnetic Disk)



สัญลักษณ์การรวม (Merge)



สัญลักษณ์การเรียงข้อมูล (Sort)

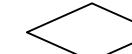


สัญลักษณ์แสดงผลทางจอภาพ (Display)



สัญลักษณ์การนำเข้าด้วยมือผ่านทางแป้นพิมพ์ (Manual Input)

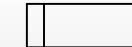
## Flowchart Symbols Programming Symbols



สัญลักษณ์แทนการตัดสินใจ (Decision)



สัญลักษณ์แทนการทำงานเป็นรอบ (Preparation)



สัญลักษณ์แทนกลุ่มโปรแกรมย่อย (Predefine Process)



สัญลักษณ์จุดเริ่มต้นและสิ้นสุด (Initial/Terminator)

## Easy Sample

□ ตัวอย่าง การหาพื้นที่สี่เหลี่ยม

□ ปัญหา คือ อยากทราบพื้นที่สี่เหลี่ยม

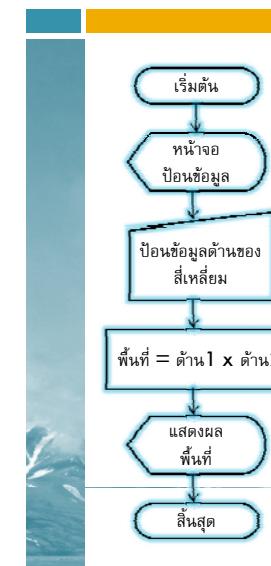
□ วิเคราะห์ หากให้ค่าความยาวของด้านไปคำนวณสูตรจะได้คำตอบไหม?

แล้วหากใส่ไปด้านเดียว ก็จะได้คำตอบแต่ สี่เหลี่ยมจัตุรัส แล้วสี่เหลี่ยมผืนผ้าจะ?

ถ้าอย่างนั้นให้มีสองตัวแปร แล้วนำมาคูณกัน ก็ได้คำตอบทั้งสี่เหลี่ยมจัตุรัสและผืนผ้า



## Easy Answer

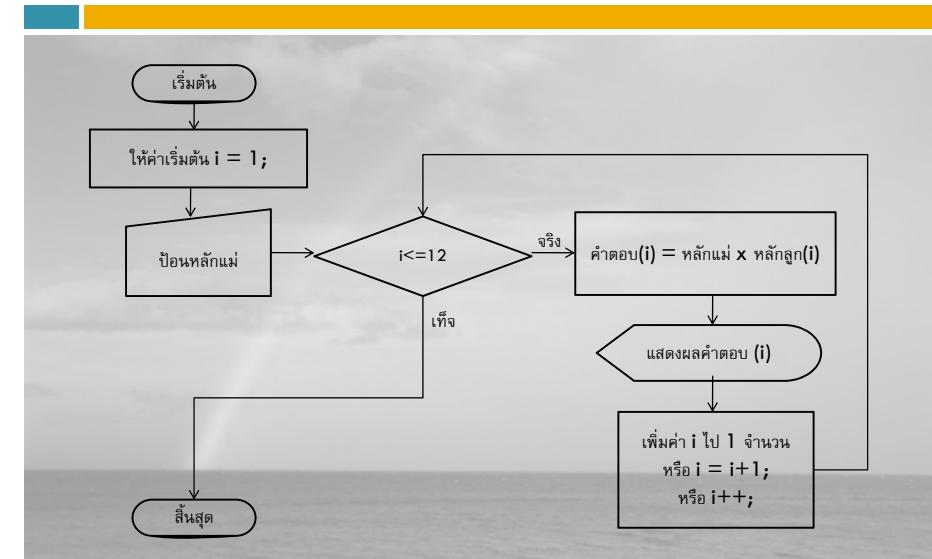


# Easy Quiz 15 minute

- คำถาม อายากได้อัลกอริทึมสำหรับสูตรคูณ
    - ช่วยเขียน Flowchart หน่อยนะ
    - อายากได้แม่สูตรคูณแม่ไหน ก็ได้ รับค่า่อา
    - คูณถึง 12 ก็พอ
    - ช่วยคิดหน่อย!



# Easy Answer



# What's Application

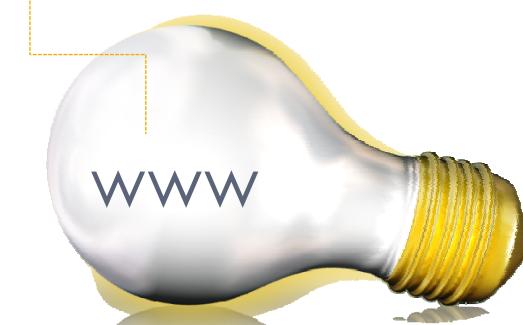
- แอ็ปพลิเคชัน คือ ชุดคำสั่งสำหรับรูปที่ถูกบรรจุการแก้ปัญหาต่างๆ เช่นเดินทางไว้ด้วยกัน เพื่อวัดคุณประสิทธิภาพและเป็นคำตอบอย่างมีประสิทธิภาพและเป็นระบบ ปัจจุบันแบ่งแอ็ปพลิเคชันตามกรอบการทำงานออกเป็น 2 ประเภท

- Client Application
  - Web Application



# Web Application

- เว็บมันดียังไง? ถึงจำเป็นต้องใช้มัน?



## Answer

- ทำงาน 24 ชม. (All day All night)
- แลกเปลี่ยนข้อมูลได้อย่างสะดวกรวดเร็ว
- ข้อมูลส่วนใหญ่เป็นปัจจุบัน
- สามารถแชร์ข้อมูล และแพร่โปรแกรมกันได้
- สามารถเข้าถึงข้อมูลจากที่ใดก็ได้ ที่มี Web Browser
- ไม่ต้องเสียเวลาลงโปรแกรม เพราะใช้ Web Browser
- ร้านค้า ไม่จำเป็นต้องมีหน้าร้าน ลดต้นทุน
- ลูกค้าสามารถสั่งซื้อสินค้า
- สามารถสร้างเครือข่ายได้ง่าย
- ร้านค้า หรือ องค์กร สามารถใช้สื่อต่างๆ นำเสนอได้อย่างอิสระมากขึ้น
- Etc.

## Website



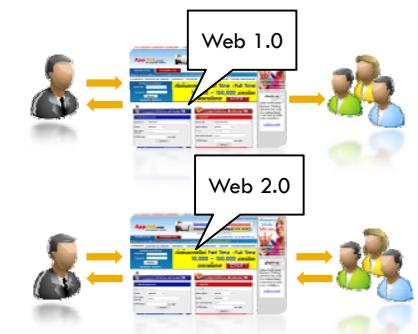
- Homepage Webpage Website แตกต่างกันไหม?
- Static Web กับ Dynamic Web แตกต่างกันไหม?
- Client Side กับ Server Side แตกต่างกันไหม?
- Web 2.0 คือ ?

## Answer

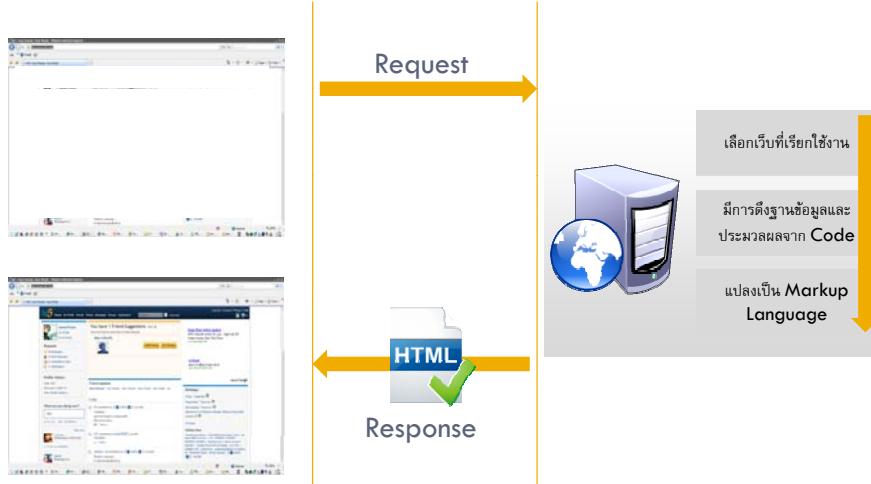
-   Homepage คือ เว็บเพจหน้าแรกของเว็บไซต์
- Webpage คือ หน้าเว็บในแต่ละหน้า
- Website คือ Web ทั้งหมดรวมกัน
- Static Web คือ เว็บที่ไม่มีการทำงาน เน้นการแสดงผลอย่างเดียว
- Dynamic Web คือ เว็บที่มีการทำงาน เน้นการโต้ตอบกับผู้ใช้
- Client Side คือ กระบวนการประมวลผลผู้ใช้ Client เช่น Java Script, CSS etc.
- Server Side คือ กระบวนการประมวลผลผู้ใช้ Server เช่น ASP.NET, JSP, PHP etc.

## Web 2.0

- คำว่า web2.0 ถูกคิดค้นขึ้นในปี 2004 โดย Tim O'Reilly เพื่อใช้อ้างถึง ยุคที่ 2 ของการให้บริการผ่านเครือข่ายอินเทอร์เน็ตที่มุ่งเน้นการให้ความร่วมมือ และแบ่งปันข้อมูลกันระหว่างกลุ่มผู้ใช้ในโลกอินเทอร์เน็ต
- Web 2.0 เป็นพัฒนาการของเว็บที่มีการให้ผู้ใช้ที่เป็น end-User มี การปรับปรุงและโต้ตอบกับเว็บได้มากขึ้น เช่น
  - Web board
  - Weblog / Blog
  - Web Services
  - AJAX
  - Web Community / Society
  - Etc.

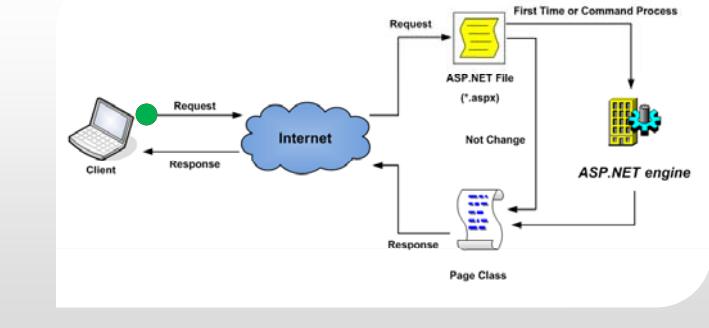


## Dynamic Web & Server Side



## Dynamic Web & Server Side

### On ASP.NET



## What's HTML

- HTML (Hyper-text Markup Language)
  - เป็นภาษาสำหรับการแสดงผลข้อมูลและฟอร์มบน Web Browser เท่านั้น
  - HTML เป็นภาษา Markup คือ เป็นภาษาที่มีการใช้ Tag ในการอธิบายข้อมูลเพื่อแปลงเป็นข้อมูลหรือฟอร์ม
    - รูปแบบของภาษา Markup เช่น <title> LOVE </title>
    - หมายถึง ให้ข้อความภายในได้การที่การที่ลัญลักษณ์ <title></title> ชี้กีดี คำว่า LOVE ให้แสดงที่ Title Bar ของ Web Browser
  - HTML อาจมีการทำงานร่วมกับ CSS (Cascade Style Sheet) เพื่อให้เกิดความสวยงามและสามารถตกแต่งหน้าเว็บได้อย่าง Dynamic มากขึ้น
  - และ HTML ยังสามารถทำงานร่วมกับ Java Script ได้ด้วย



## What's CSS (Cascade Style Sheet)

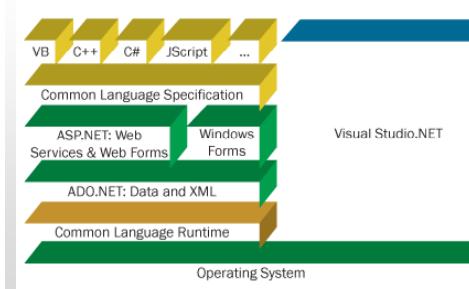
- เป็นคำสั่งหนึ่ง ที่ช่วยในการจัดรูปแบบเอกสาร HTML
- CSS เป็นคำสั่งแบบ Client-Side
- ตัวอย่างของ CSS ง่ายๆ ก็คือการที่เราสามารถปรับปรุง Hi5 ได้
- ตัวอย่างคำสั่ง

```
<style type="text/css">
Body{cursor:
url(http://www.freefeverfiles.com/hi5/cursors/cute/34.cur)}
.friend {width:90px}
.friend-picture div {margin-left: 0px !important}
</style>
```

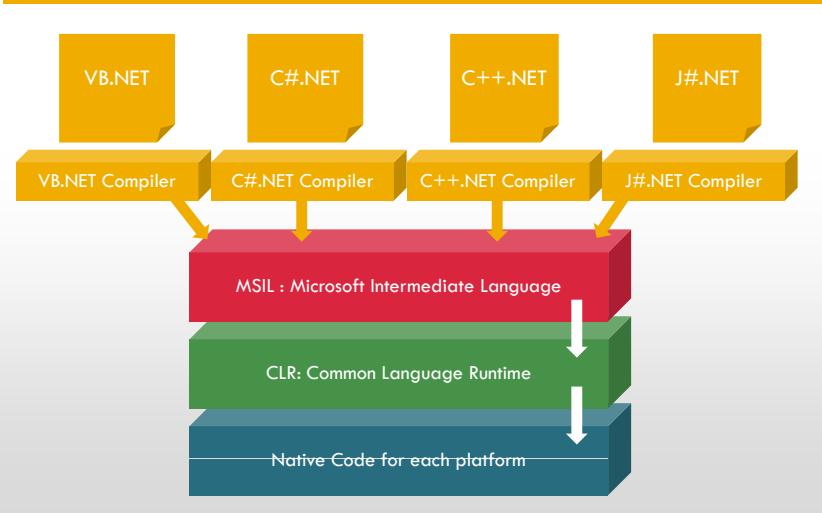
## Dot NET Technology

- Microsoft .NET คือ Platform ของบริษัทไมโครซอฟต์สำหรับการสร้างบริการเว็บ หรือ XML Web Services และ .NET เป็นผลิตภัณฑ์ที่ถือได้ว่าครอบคลุมทุกด้าน โดยสร้างขึ้นเป็นมาตรฐานของอินเตอร์เน็ต มีส่วนของเครื่องมือในการพัฒนา (tools) คือ
  - .NET Framework
  - Visual Studio .NET (IDE: Integrated Development Environment)

## Microsoft .NET Framework



## CLR (Common Language Runtime)



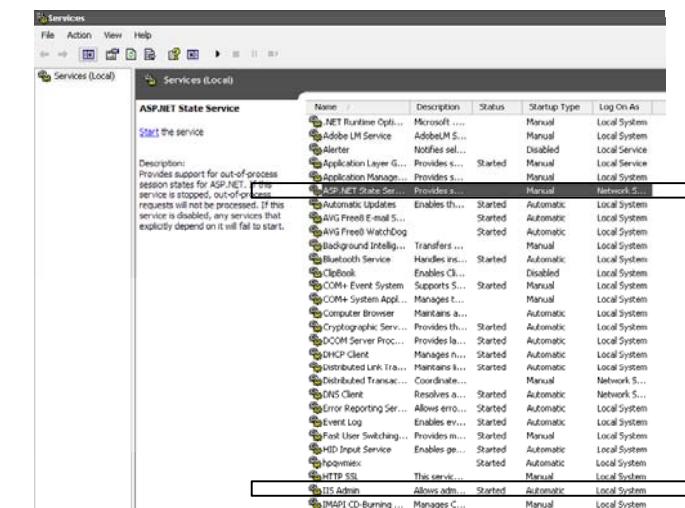
## CLR - Benefits

- จัดการหน่วยความจำโดยอัตโนมัติ
  - ▣ CLR มีการใช้ Garbage Collection ช่วยกำจัด Object ที่ไม่ใช้แล้ว เพื่อป้องกันไฟหหน่วยความจำรั่ว (Memory Leakage)
- สนับสนุนการสร้างแอ็พพลิเคชันด้วยหลากหลายภาษา
  - ▣ มีการอนุญาตให้ใช้ภาษาโปรแกรมที่ .NET Framework สนับสนุนการทำงาน ทำให้โปรแกรมเมอร์มีความหลากหลายมากขึ้น
- การส่งมอบโปรแกรมที่ง่ายขึ้นและปลอดภัย
  - ▣ ไม่ต้องมีการทำ Registration ใน Windows Registry ก่อนใช้ Components นั้นๆ แต่ใช้การ Copy File ไปใช้ได้เลย

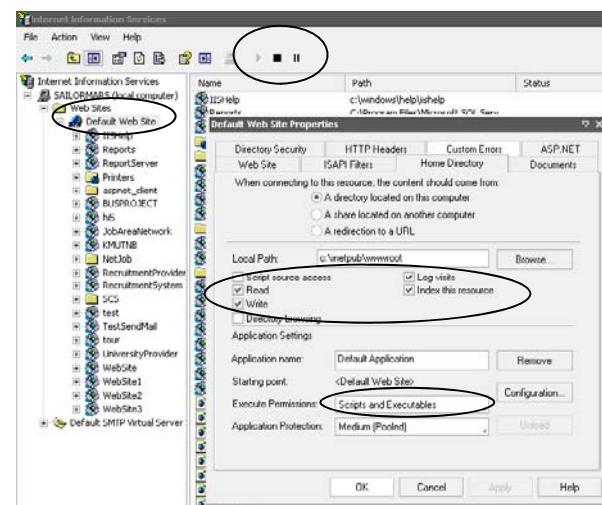
# ASP.NET

- IIS (Internet Information Services) คือ การจำลองเครื่องเราเป็น Server ให้สามารถ RUN คำสั่ง ASP.NET ได้ เพราะ ASP.NET เป็นลักษณะของ Server-Side Programming
- Visual .NET Studio เป็น IDE ที่เหมาะสมที่สุดในการพัฒนา ASP.NET ซึ่งอาจเลือกใช้ Notepad หรือ Dreamweaver ในการพัฒนา ก็ได้ เช่นกัน แต่อาจไม่สนับสนุน ASP.NET ทั้งหมดได้
- WEB Browser สามารถใช้ IE (Internet Explorer) หรือเว็บบราวเซอร์อื่นๆ ที่สนับสนุน ก็ได้
- การ RUN ใช้ IP Call Back นั่นคือ 127.0.0.1 หรือ Localhost

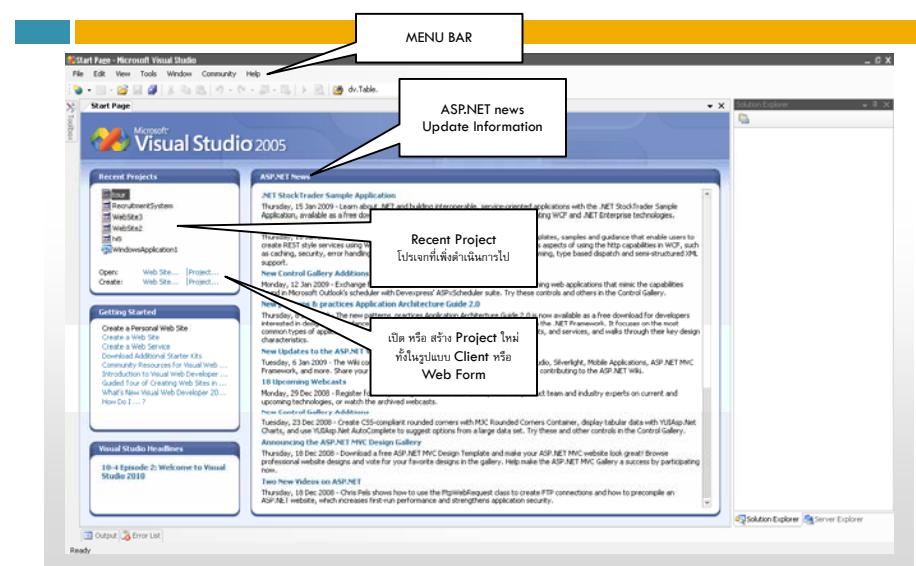
## Services



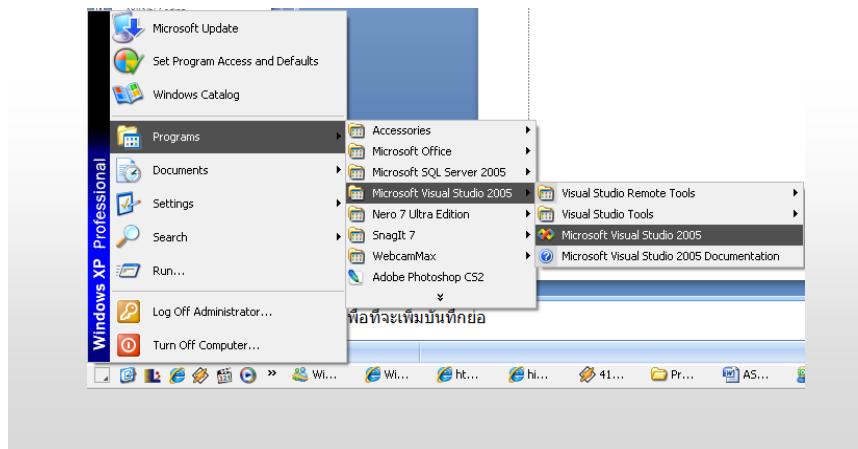
## IIS



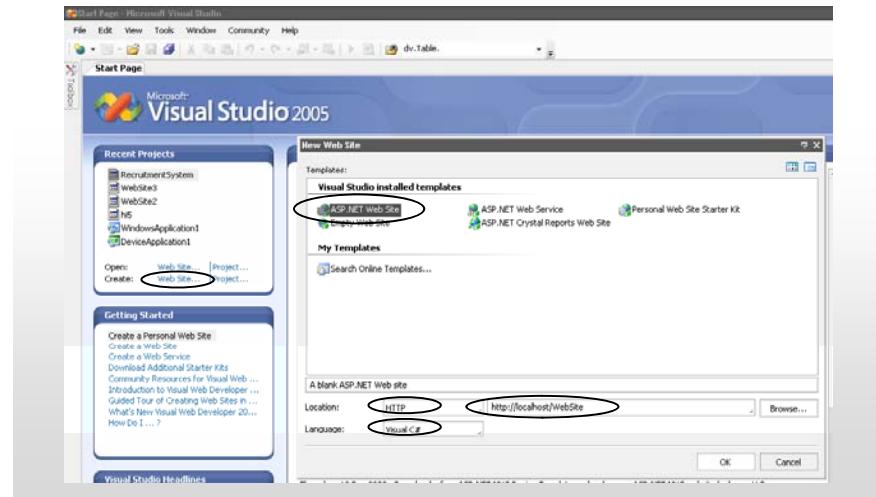
## Visual Studio .NET 2005



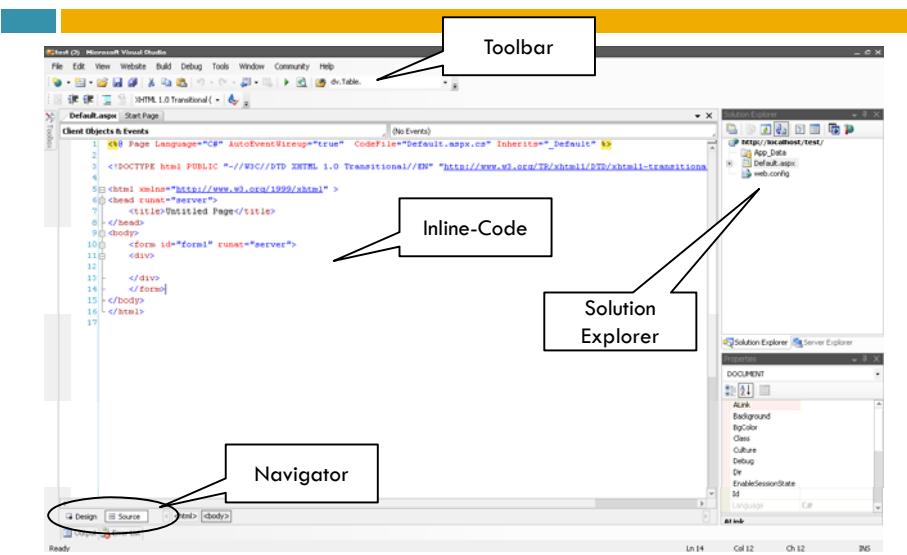
## Visual Studio .NET 2005



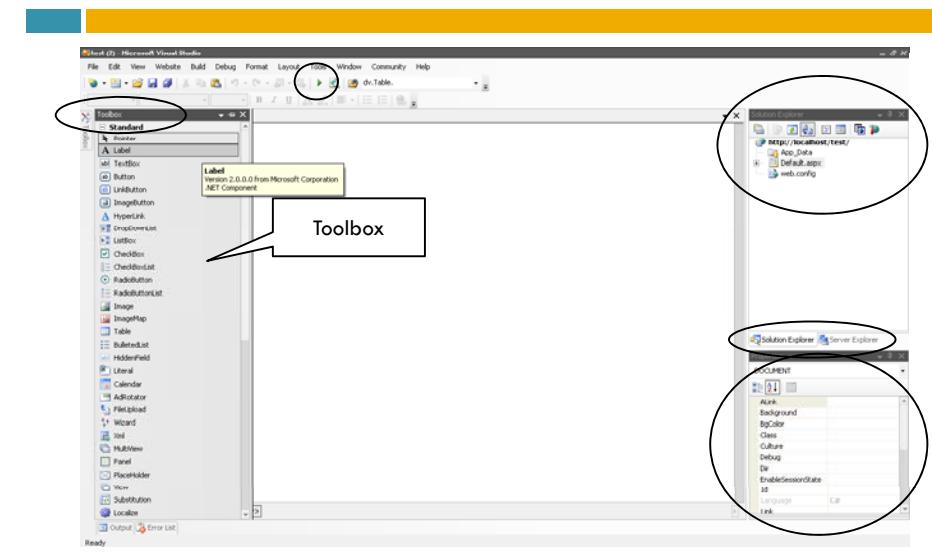
## Guidance VS.NET2005



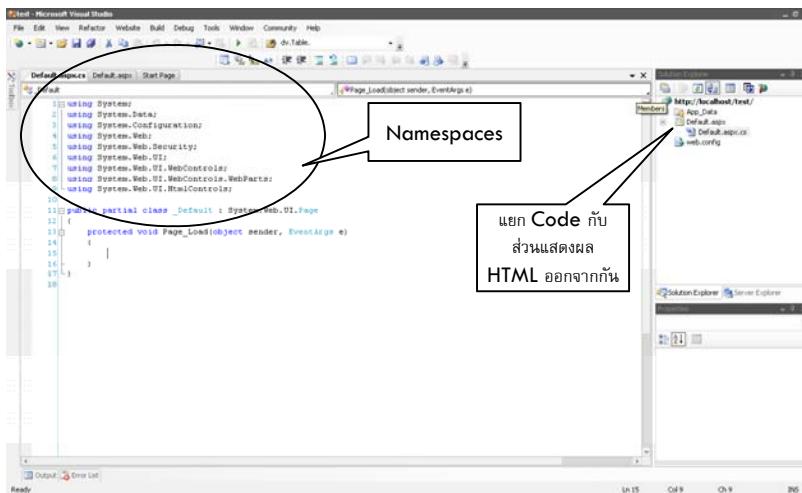
## Guidance VS.NET2005



## Guidance VS.NET2005



## Guidance VS.NET2005



## Test : Hello World

- 1. เปิด/สร้าง โปรเจก test
- 2. Double Click หน้าจอ จะเข้าสู่โหมด Coding
- 3. ใน Event : Page\_Load ให้ป้อนโปรแกรม ดัง Syntax นี้
  - Response.Write("Hello World");
  - รันโปรแกรม



## ASP.NET Coding

- ในการใช้ IDE เพื่อพัฒนา ASP.NET นั้น สามารถทำได้ 2 วิธี คือ

### ▀ □ Inline-Code

- คือการแทรก Code ภาษาโปรแกรมไปปะปนกับ HTML ทำให้ไฟล์เต็มวิธี \*.ASPX

### ▀ □ Code-Behind

- คือการแยกไฟล์ Code ออกจากส่วนแสดงผล HTML ซึ่งวิธีนี้เป็นวิธีที่แนะนำ เพราะแยกส่วนของการแสดงผลและส่วนของการประมวลผลออกจากกัน ทำให้การพัฒนาและปรับปรุงสามารถทำได้ง่ายและมีประสิทธิภาพมากกว่า แต่ในบางครั้งเราอาจจะต้อง Inline-Code ได้บางกรณี เช่น กัน

- การทำงานค่าทางในส่วนของการ Code จะอยู่ในไฟล์ \*.ASPX.CS หรือ \*.ASPX.VB ส่วนการแสดงผลอยู่ในไฟล์ \*.ASPX

## 1 Error in ASP.NET and IIS

- กรณีที่ทดสอบ Script แล้วเกิดข้อผิดพลาด ไม่สามารถ Run ได้ ซึ่งอาจเกิดจาก IIS ไม่รู้จัก ASP.NET มาก่อน โดยจะต้องทำการลงทะเบียนให้ IIS รู้จักกับ ASP.NET ก่อน

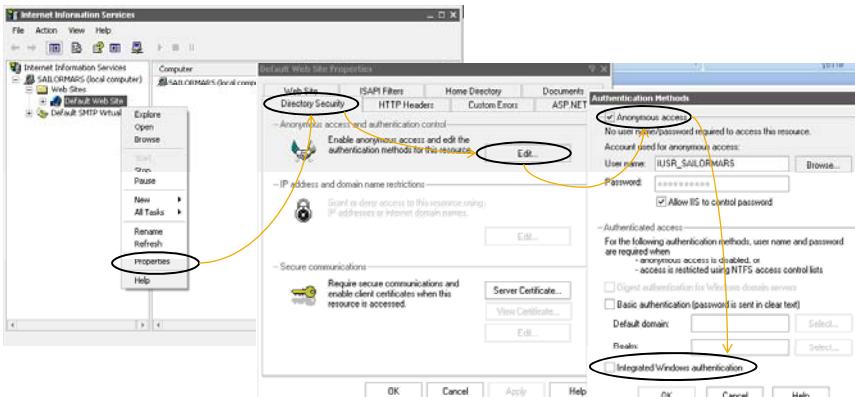
### ▀ 1. ตรวจสอบ .NET framework 2.0 ถูกติดตั้งอยู่ที่ใด เวอร์ชั่นอะไร โดยปกติแล้ว คือ C:\windows\microsoft.net\framework\v2.0.50727

### ▀ 2. เข้า Command Prompt

- cd\
- cd windows\microsoft.net\framework\v2.0.50727
- aspnet\_regiis -i

## 2Error in ASP.NET and IIS

- อาจเกิดข้อผิดพลาดอีกกรณีหนึ่ง คือ ปัญหาการแสดงผลในการเข้าใช้การ Run ASP.NET ซึ่งหน้าจะแจ้ง Debugging Not Enabled



## Test : Hello World

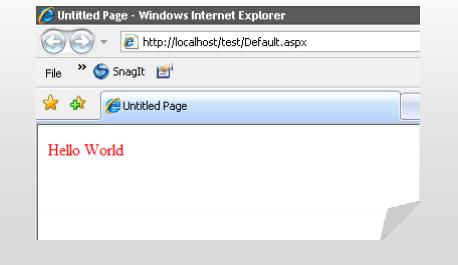
- จะเปลี่ยนสีตัวอักษรทำอย่างไร?
  - Response.Write("<font color='red'>Hello World</font>");

HTML สำหรับ Font

HTML ปิดซองที่มาครับ

- Color ที่ใช้ อาจจะอยู่ในรูปแบบของ เลขฐาน 16 (HexaDecimal)  
หรือ Base-16

Link Color Table



## C#.NET Syntax

- C# เป็น Syntax (ชื่นแท็กซ์) หรือ ภาษาสัมพันธ์ หรือ Grammar หรือ ไวยกรณ์ทางภาษาคอมพิวเตอร์ในกรอบความคิดของ ASP.NET
- C# ยึดหลัก Case-Sensitive หมายความว่า ตัวอักษรที่ทำการ Coding นั้น ตัวเล็ก ตัวใหญ่ มีความสำคัญ
- C# มีความเป็นภาษาโปรแกรมที่สนับสนุนการโปรแกรมเชิงวัตถุ



## C#.NET Variables

Type	Description
byte	-128: the largest possible value is 127. Variables with a byte type can hold 8-bit unsigned integers. Unlike sbyte variables, byte variables are not signed and can only hold positive numbers. The smallest possible value for a byte variable is 0; the largest possible value is 255.
short	Variables with a short type can hold 16-bit signed integers. The smallest possible value for a short variable is -32,768; the largest possible value is 32,767.
ushort	Variables with a ushort type can hold 16-bit unsigned integers. The u in ushort stands for unsigned. The smallest possible value of a ushort variable is 0; the largest possible value is 65,535.
int	Variables with an int type can hold 32-bit signed integers. The smallest possible value of an int variable is -2,147,483,648; the largest possible value is 2,147,483,647.
uint	Variables with a uint type can hold 32-bit unsigned integers. The u in uint stands for unsigned. The smallest possible value of a uint variable is 0; the largest possible value is 4,294,967,295.
long	Variables with a long type can hold 64-bit signed integers. The smallest possible value of a long variable is 9,223,372,036,854,775,808; the largest possible value is 9,223,372,036,854,775,807.
ulong	Variables with a ulong type can hold 64-bit unsigned integers. The u in ulong stands for unsigned. The smallest possible value of a ulong variable is 0; the largest possible value is 18,446,744,073,709,551,615.
char	Variables with a char type can hold 16-bit Unicode characters. The smallest possible value of a char variable is the Unicode character whose value is 0; the largest possible value is the Unicode character whose value is 65,535.

## C#.NET Variables

```
byte MyByteVariable;  
int _Value123;  
decimal AVeryLargeNumber;
```

float	Variables with a float type can hold a 32-bit signed floating-point value. The smallest possible value of a float type is approximately 1.5 times 10 to the 45th; the largest possible value is approximately 3.4 times 10 to the 38th.
double	Variables with a double type can hold a 64-bit signed floating-point value. The smallest possible value of a double is approximately 5 times 10 to the 324th; the largest possible value is approximately 1.7 times 10 to the 308th.
decimal	Variables with a decimal type can hold a 128-bit signed floating-point value. Variables of type decimal are good for financial calculations. The smallest possible value of a decimal type is approximately 1 times 10 to the 28th; the largest possible value is approximately 7.9 times 10 to the 28th.
bool	Variables with a bool type can hold one of two possible values: true or false. The use of the bool type is one of the areas in which C# breaks from its C and C++ heritage. In C and C++, the integer value 0 was

## C#.NET Conditions

- if..else..
- if..else if..
- switch..case..
- Operator สำหรับการเปรียบเทียบเงื่อนไขและตัวราก
  - ▣ == คือ เท่ากับ/เท่ากัน (Equal)
  - ▣ != คือ ไม่เท่ากัน
  - ▣ > คือ มากกว่า และ >= คือ มากกว่าและเท่ากัน
  - ▣ < คือ น้อยกว่า และ <= คือ น้อยกว่าและเท่ากัน
  - ▣ && คือ ตัวรากและ (And Logic)
  - ▣ || คือ ตัวรากหรือ (Or Logic)



## C#.NET Conditions : if..else..

- If .. Else .. เป็นการโปรแกรมเงื่อนไขสำหรับตัดสินใจแบบโครงสร้าง
- ประโยชน์ของ if..else.. คือ กรณีที่หากต้องมีการตัดสินใจหรือกำหนดเงื่อนไขใดๆ ให้แก่การแก้ปัญหาที่สามารถทำได้
- Syntax ของภาษา C# คือ
  - ▣ if(บรรจุเงื่อนไข)
  - ▣ { โปรแกรมอะไรสักอย่างที่จะทำก็ต่อเมื่อเงื่อนไขนั้นเป็นจริง; }

## Test : if

- โปรแกรมตามนี้

```
10  
11 public partial class _Default : System.Web.UI.Page  
12 {  
13     protected void Page_Load(object sender, EventArgs e)  
14     {  
15         int i;  
16         i = 20;  
17         if (i == 20)  
18         {  
19             Response.Write("ค่า i = " & i.ToString());  
20         }  
21     }  
22 }
```

'test' Application.  
ig the compilation of a resource required to service  
9: Operator '&' cannot be applied to operands of ty

Line 17: if (i == 20)  
Line 18: {  
Line 19: Response.Write("ค่า i = " & i.ToString());  
Line 20: }  
Line 21: }  
Source File: c:\inetpub\wwwroot\test\Default.aspx.cs Line: 19

## Test : if

The screenshot shows the Microsoft Visual Studio IDE with an ASP.NET page named Default.aspx. The code in the code-behind file is as follows:

```
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15         int i;
16         i = 20;
17         if (i == 20)
18         {
19             Response.Write("ตัว i = " + i.ToString());
20         }
21     }
22 }
23
```

A callout box points to the line `Response.Write("ตัว i = " + i.ToString());` with the text "ใช้มัตว์แบบกับข้อความใช้ Syntax "+"

The browser window shows the output: ตัว i=20

## Test : if..else..

The screenshot shows the Microsoft Visual Studio IDE with an ASP.NET page named Default.aspx. The code in the code-behind file is as follows:

```
13 protected void Page_Load(object sender, EventArgs e)
14 {
15 }
16 protected void Button1_Click(object sender, EventArgs e)
17 {
18     if (Convert.ToDecimal(textBox1.Text) > 10)
19     {
20         Response.Write("ตัวเลขมากกว่า 10");
21     }
22     else
23     {
24         Response.Write("ตัวเลขน้อยกว่า 10");
25     }
26 }
27
```

## Test : if..else..

The screenshot shows the Microsoft Visual Studio IDE with an ASP.NET page named Default.aspx. The code in the code-behind file is as follows:

```
13 if (Convert.ToDecimal(textBox1.Text) > 10)
14 {
15     Response.Write("ตัวเลขมากกว่า 10");
16 }
```

An error dialog box is displayed, stating: "FormatException was unhandled by user: Input string was not in a correct format." The browser window shows two test cases:

- Case 1: ตัวเลขมากกว่า 10 (Input: 200) - Shows the error dialog.
- Case 2: ตัวเลขน้อยกว่า 10 (Input: 2.56) - Shows the output: ตัวเลขน้อยกว่า 10.

## Test : if-else

The screenshot shows the Microsoft Visual Studio IDE with an ASP.NET page named Default.aspx. The code in the code-behind file is as follows:

```
13 if (Convert.ToInt32(textBox1.Text) > 200 || Convert.ToInt32(textBox1.Text) < 100)
14 {
15     Response.Write("ยักษ์");
16 }
17 else
18 {
19     Response.Write("มนุษย์");
20 }
```

Two input fields are shown: ส่วนสูง: and น้ำหนัก: . A dropdown menu labeled "ค่านอก" is also visible.

Checklist items:

- ทำระบบกรอกส่วนสูงและน้ำหนัก
- หากส่วนสูงมากกว่า 200 และน้ำหนักมากกว่า 100 ให้แสดงผลว่า “ยักษ์”
- หากน้ำหนักมากกว่า หรือ น้ำหนักน้อยกว่า 100 ให้แสดงผลว่า “มนุษย์”

## Test : if-else

The screenshot shows the Microsoft Visual Studio interface. In the top-left, there's a code editor with the following C# code:

```
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15     }
16     protected void Button1_Click(object sender, EventArgs e)
17     {
18         if (Convert.ToDecimal(TextBox1.Text) > 200 && Convert.ToDecimal(TextBox2.Text) > 100)
19         {
20             Response.Write("ขำ");
21         }
22         else
23         {
24             Response.Write("ไม่ขำ");
25         }
26     }
27 }
```

In the bottom-right, a browser window titled "Untitled Page" displays the output of the code. It has two text input fields: "ส่วนสูง: 234" and "น้ำหนัก: 100". Below them is a button labeled "คำนวณ". The page content area shows the text "ไม่ขำ".

## C#.NET Conditions : if..else if

- If .. Else If เป็นการโปรแกรมเงื่อนไขสำหรับตัดสินใจแบบโครงสร้าง
- ประโยชน์ของ if..else If คือ กรณีที่หากต้องมีการตัดสินใจหรือกำหนดเงื่อนไขใดๆ หลายกรณี
- Syntax ของภาษา C# คือ
  - if(บรรจุเงื่อนไข1)
    - { โปรแกรมอะไรสักอย่างที่จะทำก็ต่อเมื่อเงื่อนไขที่ 1 เป็นจริง; }
  - else if(บรรจุเงื่อนไขที่2)
    - { โปรแกรมอะไรสักอย่างที่จะทำก็ต่อเมื่อเงื่อนไขที่ 2 เป็นจริง; }
  - ... จะเพิ่ม else if ต่อได้เรื่อยๆ และอาจจบด้วย else กรณีที่ไม่ตรงกับเงื่อนไขใดๆ ทั้งล้วน หรือที่เรียกว่า Default

## Test : if-else if

The screenshot shows the Microsoft Visual Studio interface. In the top-left, there's a code editor with the following C# code:

```
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15     }
16     protected void Button1_Click(object sender, EventArgs e)
17     {
18         if (Convert.ToDecimal(TextBox1.Text) <= 50 )
19         {
20             Response.Write("Grade E");
21         }
22         else if(Convert.ToDecimal(TextBox1.Text) <= 60)
23         {
24             Response.Write("Grade D");
25         }
26         else if (Convert.ToDecimal(TextBox1.Text) <= 70)
27         {
28             Response.Write("Grade C");
29         }
30         else if (Convert.ToDecimal(TextBox1.Text) <= 80)
31         {
32             Response.Write("Grade B");
33         }
34         else
35         {
36             Response.Write("Grade A");
37         }
38     }
39 }
```

In the bottom-right, a browser window titled "Untitled Page" displays the output of the code. It has one text input field with the value "คะแนน: 100". Below it is a button labeled "คำนวณ". The page content area shows the text "Grade E".

## Test : if-else if

The screenshot shows the Microsoft Visual Studio interface. In the top-left, there's a code editor with the following C# code:

```
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15     }
16     protected void Button1_Click(object sender, EventArgs e)
17     {
18         if (Convert.ToDecimal(TextBox1.Text) <= 50 )
19         {
20             Response.Write("Grade E");
21         }
22         else if(Convert.ToDecimal(TextBox1.Text) <= 60)
23         {
24             Response.Write("Grade D");
25         }
26         else if (Convert.ToDecimal(TextBox1.Text) <= 70)
27         {
28             Response.Write("Grade C");
29         }
30         else if (Convert.ToDecimal(TextBox1.Text) <= 80)
31         {
32             Response.Write("Grade B");
33         }
34         else
35         {
36             Response.Write("Grade A");
37         }
38     }
39 }
```

In the bottom-right, a browser window titled "Untitled Page" displays the output of the code. It has three text input fields:

- Grade A: คะแนน: 100
- Grade E: คะแนน: 34
- Grade B: คะแนน: 79

Each has a "คำนวณ" button below it. The Grade A section shows the text "Grade E".

## Switch..Case..

- ใช้ในการนี่ที่ต้องมีการตรวจสอบเงื่อนไขจำนวนมาก

- มี Syntax ดังนี้

- switch (ตัวแปรที่ต้องการตรวจสอบ)

```
{
```

```
case "ตัวแปรที่มีความลับเฉพาะใน switch":
```

```
    โปรแกรมที่ต้องการเมื่อ case นี้เป็นจริง;
```

```
    break;
```

```
default:
```

```
    โปรแกรมนอกเหนือจาก case ต่างๆ
```

```
    break;
```

```
}
```

## Test : Switch..Case..

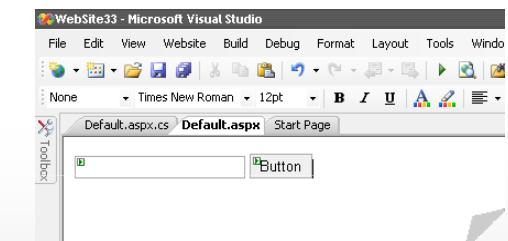
- ให้เขียนโปรแกรมแปลคำ

- ป้อน Fish แสดง ปลา

- ป้อน bird แสดง นก

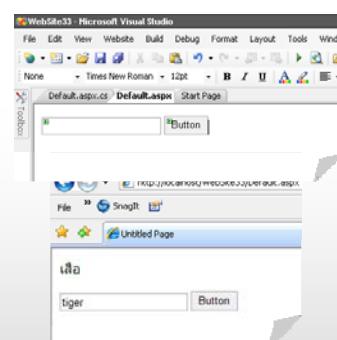
- ป้อน tiger แสดง เสือ

- อื่นๆ แสดง ไม่มีคำแปล



## Test : Switch..Case..

```
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15     }
16     protected void Button1_Click(object sender, EventArgs e)
17     {
18
19         switch (TextBox1.Text.ToString())
20         {
21             case "fish":
22                 Response.Write("ปลา");
23                 break;
24             case "bird":
25                 Response.Write("นก");
26                 break;
27             case "tiger":
28                 Response.Write("เสือ");
29                 break;
30             default :
31                 Response.Write("ไม่มีคำแปล");
32                 break;
33         }
34     }
35 }
```

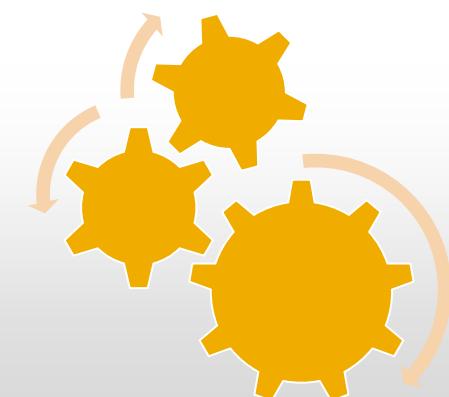


## C#.NET Iterations

- For..

- While..

- Do.. While..



## C#.NET Iterations : For

- For เทมาะกับการวนลูป ที่ทราบจำนวนรอบ หรือคาดหมายจำนวนรอบไว้

- รูปแบบของ Syntax คือ

```
for(ตัวแปรเริ่มต้น;เงื่อนไข;การเปลี่ยนแปลงตัวแปร)
{
    โปรแกรม;
}
```

\*การวนลูปจะเกิดการวนก็ต่อเมื่อเงื่อนไขเป็นจริง

## C#.NET Iterations : For : Sample

```
1U
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15         for (int i = 1; i <= 10; i++)
16         {
17             Response.Write(i.ToString() + "<br>");
18         }
19     }
20 }
21
22
```



## While..

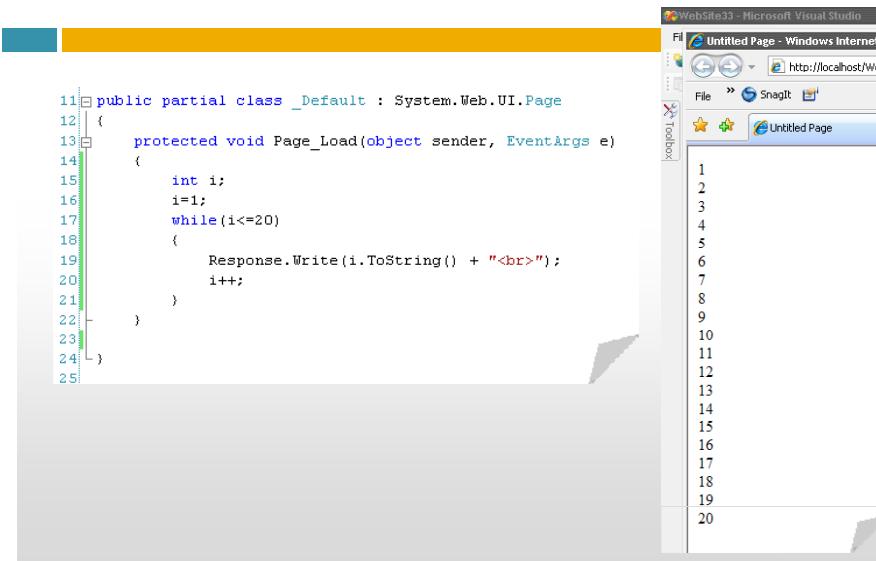
- เป็นการวนลูป แต่ใช้กรณีที่ไม่ทราบจำนวนรอบที่แน่นอน

```
while(เงื่อนไข)
{
    โปรแกรม;
}
```

\*จะออกจาก Loop ก็ต่อเมื่อเงื่อนไขเป็นเท็จ

## Sample : While

```
1I
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15         int i;
16         i=1;
17         while(i<=20)
18         {
19             Response.Write(i.ToString() + "<br>");
20             i++;
21         }
22     }
23 }
24
25
```



## Sample : Do..While..

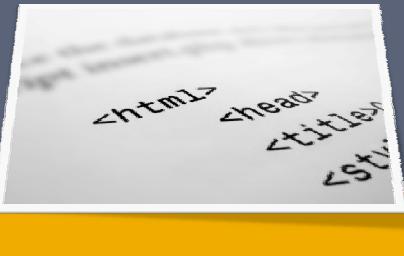
- เป็นการวนลูปโดยจะทำที่ **do** ก่อน และจึงตรวจสอบเงื่อนไข หากเงื่อนไขเป็นจริงจะกลับขึ้นไปวนลูปใหม่จนกว่าเงื่อนไขจะเป็นเท็จ

- **Syntax** ของ Do..While

```
do
{
    โปรแกรม;
}
while (เงื่อนไข)
```

## ASP.NET WITH C# CLASS 2

ณัฐภพ แก้วรัตนภพ



### Comment in C#

- การเขียน Comment ในภาษา C#.NET ใช้เครื่องหมาย “//” และ ข้อความหลังจากเครื่องหมายนี้ตัว Complier จะข้ามการแปลคำสั่งไป บรรทัดนั้น
- ในภาษา C# ยังมีตัวทำ Comment อีก คือ “/\* และ \*/” ซึ่ง สามารถทำ Comment แบบหลายบรรทัดได้โดยใช้สัญลักษณ์นี้ไปครอบ (markup) ข้อความนั้นๆที่ต้องการทำเป็น Comment
- ข้อดีของการทำ Comment คือ ทำให้เราเข้าใจและทราบว่าโปรแกรม ณ จุดนี้จะทำงานอย่างไร หรือมีการแทรกคำที่ใช้สำหรับเตือนความจำเกี่ยวกับ ตัวแปรได้ เป็นต้น

### [ ] vs ( ) in ASP.NET

- เครื่องหมาย [ ] หรือ Block Quote ใช้ในการอ้างอิงถึงแหล่งข้อมูล
- เครื่องหมาย ( ) ใช้ในการเรียกใช้ Function เท่านั้น
- ภาษา C# จะไม่ປະກາດใช้เครื่องหมาย () และ [] เข้าด้วยกัน ทำให้ C# เป็นภาษาที่มีรูปแบบที่ดีและเหมาะสม ง่ายต่อการทำความเข้าใจ มากกว่า VB.NET

### Server Controls ที่สำคัญ

- |                |                     |
|----------------|---------------------|
| ▪ Text Box     | ▪ Table             |
| ▪ Label        | ▪ Calenda           |
| ▪ Button       | ▪ File Upload       |
| ▪ DropDownList | ▪ Multiview         |
| ▪ ListBox      | ▪ View              |
| ▪ CheckBox     | ▪ <u>Validation</u> |
| ▪ RadioButton  | ▪ <u>Navigation</u> |
| ▪ Image        | ▪ <u>Data</u>       |

## การรับส่งค่าแบบ GET

- แบบ GET จะมีข้อมูลที่ส่งติดไปกับ URL
  - ▣ ใช้ HyperLink และแก้ไขที่ NavigateURL หรือ ใช้ Response.Redirect(); ทำการเปลี่ยนหน้าเว็บไปยังหน้าเว็บที่ต้องการ พร้อมการส่งค่า
  - ▣ รูปแบบการต่อท้ายของ HyperLink  
`mylink.aspx?name=john&surname=mayer`
  - ▣ รูปแบบการใช้ Redirect  
`response.redirect("mylink.aspx?name=john&surname=mayer");`
  - ▣ การรับค่าใช้ `request.QueryString["name"];`

## Test1 : GET

- สร้างเพจใหม่ ใช้ชื่อว่า Default2.aspx
  - ▣ โปรแกรมสำหรับรับค่าที่เหตุการณ์ Page\_Load
  - ▣ `Response.write("START:" + request.QueryString["start"] + "</BR>");`
  - ▣ `Response.write("STOP:" + request.QueryString["stop"] + "</BR>");`
- ในเพจ Default1.aspx ใช้ Control ชื่อ HyperLink
- แก้ไข Properties ที่ NavigateURL โดยเปลี่ยนเป็น  
`Default2.aspx?start=10&stop=50`

## Test2 : GET

- สร้างเพจ default3.aspx
- สร้าง textbox 4 ชิ้น โดยให้ชิ้นแรกกรอกข้อมูลชื่อ ชิ้นที่สองกรอกข้อมูล นามสกุล ชิ้นที่สามกรอกข้อมูลอายุ ชิ้นที่สี่กรอกข้อมูลเบอร์โทรศัพท์
- สร้างปุ่มสำหรับส่งไปยังเพจ default4.aspx โดยในเพจ default4.aspx ให้รับค่าและแสดงค่าข้อมูลໄล่ลงมา

## การรับส่งค่าแบบ POST

- การส่งค่าแบบ POST เป็นการอัพโหลดข้อมูล โดยไม่แสดงค่าตัวแปรที่ท้าย URL ซึ่งใน ASP.NET จะใช้ `Server.Transfer` เพื่อสะดวกในการส่งค่า หรือใช้ ตัวแปรประเภท `Session` ในการส่งได้เช่นกัน
- การใช้ `Server.Transfer` ให้ระบุชื่อไฟล์ปลายทางที่ต้องการส่ง
  - ▣ `Server.Transfer("default2.aspx");`
  - ▣ ในการรับข้อมูลใช้ `Request.Form["TextBox1"]` โดยสามารถเข้าถึงข้อมูลภายใน `Controls` ของหน้าเพจที่ส่งได้อย่างสะดวกและง่าย

## Test1 : Server.Transfer

- สร้างเพจใหม่ ใช้ชื่อว่า Default5.aspx
  - ▣ โปรแกรมสำหรับรับค่าที่เหตุการณ์ Page\_Load
  - ▣ Response.write("START:" + request.form["textbox1"] + "</BR>");
  - ▣ Response.write("STOP:" + request.form["textbox2"] + "</BR>");
- ในเพจ Default6.aspx สร้าง TextBox 2 ชิ้น โดยชิ้นแรกกรอกข้อมูล START ชิ้นที่สองกรอกข้อมูล STOP และสร้างปุ่มสำหรับส่งข้อมูลไปยังเพจ Default5.aspx

## Test2 : Sending Value

- สร้างเพจสำหรับแสดงข้อมูลสูตรคูณ โดยในหน้าแรก ให้ทำ textbox กรอกข้อมูลและสร้างปุ่มสำหรับส่งข้อมูลไปหน้าที่สอง โดยในหน้าที่สองต้องทำการรับค่าหลักแล้ว และทำการคำนวนพร้อมแสดงผลแม่สูตรคูณถึง 12
  - ▣ ส่งแม่ 2 ได้ค่าดังนี้
    - $2 \times 1 = 2$
    - $2 \times 2 = 4$
    - ....เรื่อยๆ....
    - $2 \times 12 = 24$

## Session Variable

- การรับส่งข้อมูลโดยใช้ Session นั้นมีความสำคัญในการจำพรางข้อมูลเพื่อส่งไปยังปลายทาง ซึ่งสรุปลักษณะของตัวแปรแบบ Session ที่สำคัญได้ 2 กรณี คือ
  - ▣ ตัวแปร Session มีขอบเขตการทำงานกับผู้ใช้แต่ละคนเท่านั้น (Unique)
  - ▣ ตัวแปร Session เป็นการใช้ทรัพยากรฝั่ง Server โดยการใช้งาน Session จำเป็นต้องใช้อายุร่วมด้วย เพราะอาจทำให้ Server ทำงานหนักและอาจจะให้บริการกับ Traffic ที่มีขนาดใหญ่ได้ไม่เต็มประสิทธิภาพ ดังนั้นหากเลิกใช้งานหรือถ่ายตัวแปรแล้ว ให้ทำการทำลาย Session นั้นทั้งทันที
  - ▣ ตัวแปรแบบ Session เหมาะกับการส่งข้อมูลที่มีความสำคัญและต้องการความปลอดภัย เช่น Username และ Password แต่ในปัจจุบันนิยมใช้ส่งข้อมูลกัน เพราะมีความปลอดภัยกว่าการใช้ HTML form ส่ง

## Session Variable

- โดย Syntax ของตัวแปรแบบ Session มีดังนี้
  - ▣ Session["ชื่อตัวแปรเชลชั่น"]
  - ▣ ตัวอย่างการกำหนดค่า Session["name"] = "nutthapat";
  - ▣ การแสดงผลตัวแปรสามารถใช้ Session["name"].ToString();
  - ▣ และสามารถถ่ายค่าให้กับตัวแปรอื่นได้ name = Session["name"]
  - ▣ สามารถกำหนดเวลาได้โดยหน่วยเป็น นาที Session.Timeout = 15
  - ▣ สามารถสร้างหมายเลขเชลชั่นได้ Session.SessionID

## Session Variable

- การทำลายตัวแปร Session
  - ทำลายเฉพาะตัวแปร Session.Remove("ชื่อตัวแปรเชสชั้นที่ต้องการลบ");
  - ทำลายทุกตัวแปร Session ใช้ Session.RemoveAll();
  - เมื่อทำลายตัวแปรเชสชั้นแล้วข้อมูลจะหายไป และ Server จะทำงานน้อยลง

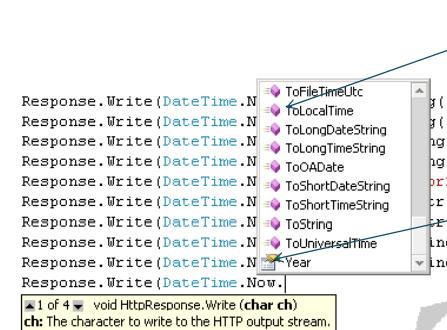
## Test : Session Variable

- สร้างระบบเช็ค Username และ Password
- สร้างเพจและสร้าง Textbox สำหรับกรอกข้อมูล Username และ Password
- สร้างปุ่มลับข้อมูลไปตรวจสอบอีกเพจ โดยให้ใช้ตัวแปร Session
- ในเพจที่ใช้ตรวจสอบจะต้องกำหนดเงื่อนไข
  - หาก Username คือ ชื่อของนักศึกษา
  - และ Password คือ รหัสนักศึกษา
  - ให้ทำการแสดงคำว่า “Username & Password Correct”
  - หากไม่ถูกต้องให้แสดงคำว่า “Username & Password Incorrect”
  - หากไม่ทำการกรอกข้อมูลจะต้องแจ้งเตือนว่า “Please fill data”
  - และแสดงหมายเลข Session ออกบนหน้าจอตัวย

## Date Time

- พิ้งก์ชั้นประเภท Date Time มีสำคัญเข่นกันในการเขียนโปรแกรม  
เนื่องจากในปัจจุบันในการแก้ไขปัญหาอาจมีเรื่องของวันที่และเวลาเกี่ยวข้อง  
กับการแก้ไขปัญหาเสมอ
- โดย Syntax เกี่ยวกับวันที่มีดังนี้
  - Datetime.now สำหรับแสดงวันที่หรือเวลาปัจจุบัน
  - ตัวอย่าง
    - DateTime.Now.ToString(); แสดงวันแบบยาว
    - DateTime.Now.ToShortDateString(); แสดงเวลาแบบยาว
    - DateTime.Now.ToShortTimeString(); แสดงวันแบบสั้น
    - DateTime.Now.ToString("yyyy-MM-dd"); แสดงเวลาแบบสั้น

## Tip!



- สังเกต icon กอล์ฟสีม่วง  
เหมือนเคลื่อนไหว คือ พิ้งก์ชั้น  
ซึ่งจะต้องใส่เครื่องหมาย  
สำหรับพิ้งก์ชั้น "(" ต่อท้าย  
เสมอ
- สังเกต icon หัวมือชี้สีเหลือง  
คือ Properties ของตัวแปร  
นั้นๆ สามารถนำมาแสดงหรือ  
ปรับเปลี่ยนได้ หากໃห้ในการ  
แสดงจะต้องเสริมด้วยพิ้งก์ชั้น  
tostring() เสมอ

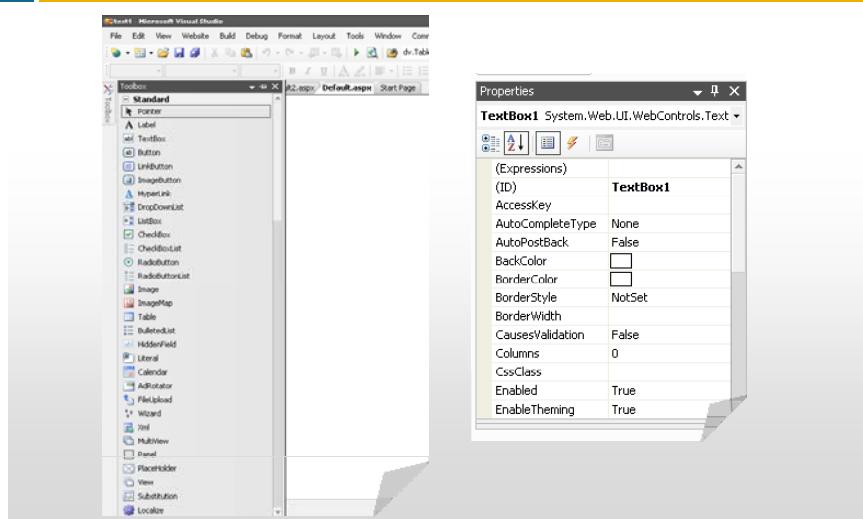
## Test : Date-Time

- จงแสดงว่า อีก 10 วันเป็นวันที่เท่าไหร่? เช่น (วันนี้ 10/2/2552 อีก 10 วันต้องเป็นวันที่ 12/2/2552 เป็นต้น)
- จงแสดงว่า อีก 5 ชั่วโมงถัดไป เป็นเวลาเท่าไหร่?
- วันนี้เป็นวันที่เท่าไหร่นับตั้งแต่ต้นปี (เช่น วันที่ 40 ของปี)
- จงแสดงวันที่วันนี้ เป็นวันอะไรเป็นภาษาไทย? เช่น (จันทร์, อังคาร ...)

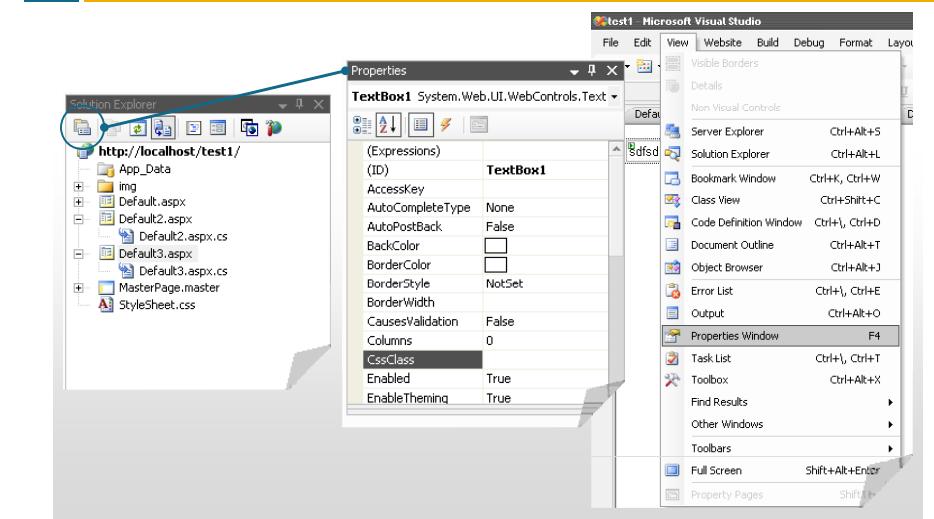
## Answer

```
Response.Write(DateTime.Now.ToString() + "<br>");  
Response.Write(DateTime.Now.ToString("yyyy-MM-dd") + "<br>");  
Response.Write(DateTime.Now.ToString("dd/MM/yyyy") + "<br>");  
Response.Write(DateTime.Now.ToString("HH:mm:ss") + "<br>");  
Response.Write(DateTime.Now.ToString("HH:mm") + "<br>");  
Response.Write(DateTime.Now.AddDays(10).ToString() + "<br>");  
Response.Write(DateTime.Now.AddHours(5).ToString() + "<br>");  
Response.Write(DateTime.Now.DayOfWeek.ToString() + " : วันอะไร? SwitchCase วันภาษาไทย<br>");  
Response.Write(DateTime.Now.DayOfYear.ToString() + " : ตั้งแต่ต้นปีเป็นวันที่เท่าไหร่ซองปี? <br>");
```

## Server Controls

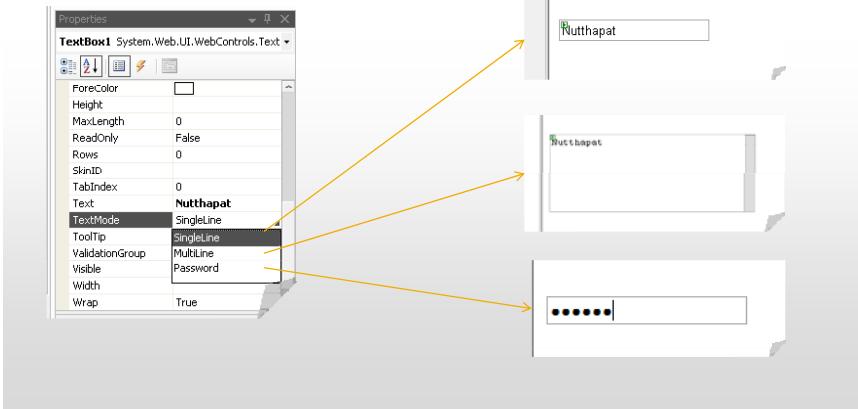


## การปรับแต่ง Properties ของ Controls



# Server Controls ที่สำคัญ

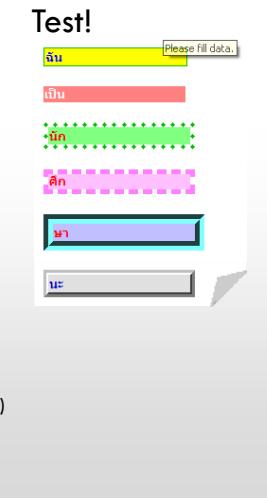
## □ Sample : TextBox



# Server Controls ที่สำคัญ

## □ TEXTBOX หรือ กล่องข้อความ

- Properties ที่สำคัญได้แก่
  - BackColor คือ สีของพื้น Control ตัวนี้
  - BorderColor คือ สีของขอบ
  - BorderStyle คือ รูปแบบของขอบ
  - BorderWidth หน่วยเป็น Pixel
  - CausesValidation คือ การตรวจสอบข้อมูล
  - CSS Class คือ การตั้งค่ารูปแบบด้วย CSS
  - Enabled คือ การกำหนดค่าของตัวอักษร
  - Font คือ สีของตัวอักษร
  - ForeColor คือ สีของตัวอักษร
  - MaxLength คือ จำนวนตัวอักษรที่เป็นไปได้ของข้อมูล
  - ReadOnly คือ สำหรับอ่านอย่างเดียวห้ามทำการป้อนข้อมูล
  - TabIndex คือ ลำดับการเข้าถึง Control ตัวนี้
  - Text คือ ข้อมูลที่หั้นแสดงใน Control
  - TextMode คือ รูปแบบของ TextBox (SingleLine, MultiLine, Password)
  - ToolTip คือ การแสดงคำแนะนำเมื่อมouse ไปวางบน Control
  - Visible คือ การแสดงหรือไม่แสดง Control นี้



# Server Controls ที่สำคัญ

## □ ให้นักศึกษาลองปรับแต่งลักษณะของ Controls ดังต่อไปนี้

- Label
- Button
- DropDownList (การแทรก items)
- CheckBox (Checked)
- RadioButton
- Image
- Table

# การเขียนโปรแกรมกับ Controls

## □ Controls แต่ละตัวสามารถเขียนโปรแกรมควบคุมลักษณะได้

ซึ่งไม่จำเป็นต้องปรับแต่งในเครื่องมือ Properties แต่สามารถกระทำได้เมื่อเราได้ทำการโปรแกรมไว้ ข้อตีคือ เพิ่มความยืดหยุ่นในการแสดงผลและเพิ่มความหลากหลายในการโปรแกรม

ตัวอย่างเช่น

```
TextBox.BackColor = Color.Blue;
```

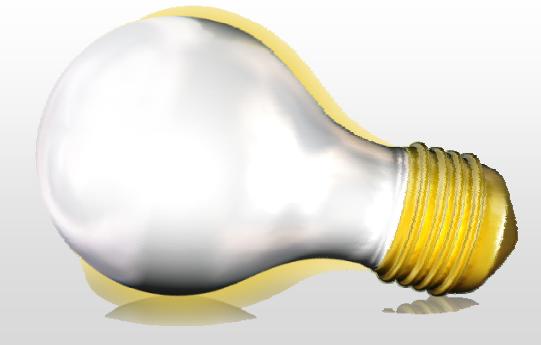
## การเขียนโปรแกรมกับ Controls

- ชี้หากต้องการเขียนโปรแกรมและมีการกำหนดสิ่งให้แก่ **Control** จำเป็นต้องประกาศ **NameSpace** ที่เกี่ยวกับการรับภาพและสี ชื่อ **Namespace** นี้มีการประกาศดังนี้

```
1  using System;
2  using System.Data;
3  using System.Configuration;
4  using System.Collections;
5  using System.Web;
6  using System.Web.Security;
7  using System.Web.UI;
8  using System.Web.UI.WebControls;
9  using System.Web.UI.WebControls.WebParts;
10 using System.Web.UI.HtmlControls;
11 using System.Drawing;
12
```

## Test

- ให้นักศึกษาลองปรับเปลี่ยนรูปแบบและสีล้านของ **Controls** แต่ละตัว โดยใช้การเขียนโปรแกรมแบบ **Code-Behind**
- กลุ่มละ 1 **Control** พร้อมออกแบบมาทำเสนอ

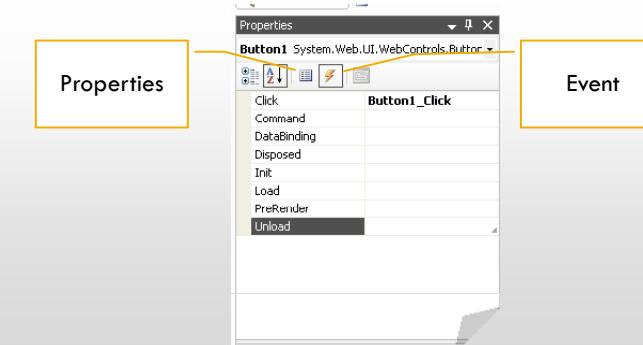


## การเขียนโปรแกรมกับ Controls

- การเขียนโปรแกรมร่วมกับ **Controls** จำเป็นต้องเข้าใจในเรื่องของ เหตุการณ์พาไป หรือ **Event Handles** ซึ่งมีความสำคัญกับการเขียน โปรแกรมแบบ **Visual**
- โดย **Event Handles** เป็นตัวดักจับเหตุการณ์ต่างๆ ที่เกิดขึ้น ในการ เขียนโปรแกรมเราจำเป็นต้องทราบธรรมชาติของ **Controls** แต่ละตัวว่า สามารถเกิดเหตุการณ์ใดขึ้นได้บ้าง เพื่อสามารถทำการโปรแกรมผูกกับ **Controls** ต่างๆ ได้อย่างถูกต้องและใช้ได้อย่างเต็มประสิทธิภาพ ก่อให้เกิดความหลากหลายในการเขียนโปรแกรมและแก้ปัญหา

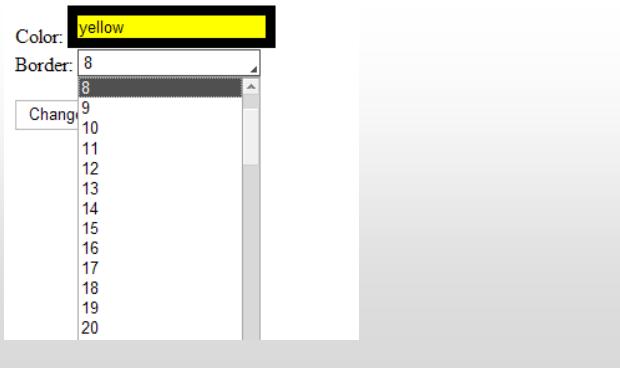
## การเขียนโปรแกรมกับ Controls

- Event** ที่น่าสนใจ เช่น
  - Page\_Load** คือ เหตุการณ์โหลดหน้าเพจนั้นขึ้น
  - Button1\_Click** คือ เหตุการณ์เมื่อคลิก **Control** ที่ชื่อ **Button1**



## รู้จัก DropDownList

- DropDownList ถูกสร้างโดยมีวัตถุประสงค์เพื่อสร้างตัวเลือกแบบหลายตัวเลือก โดยข้อดีของตัวเลือกลักษณะนี้คือสามารถสร้างตัวเลือกได้จำนวนมาก



## การโปรแกรม DropDownList

- DropDownList เป็น Control ที่ความสามารถป้อนตัวเลือก หรือที่เรียกว่า item เข้าไปใน Control ได้ โดยมี 2 วิธี คือ
  - วิธีแรก แบบ Static คือการเพิ่ม item โดยใช้ PropertiesTools โดยวิธีการนี้จะทราบข้อมูลแน่ชัดว่าควรจะใช้ข้อมูลช่วงใดเป็นตัวเลือก เช่น เพศ(ชาย,หญิง) หรือ สถานะภาพ(โสด,สมรส,หย่าร้าง) เป็นต้น
  - วิธีที่สอง แบบ Dynamic คือการเพิ่มโดยการโปรแกรม ซึ่งใช้วิธีการ iteration หรือ Loop ในการป้อนตัวเลือกเข้าไปใน Control โดยวิธีการนี้ใช้ในการนับที่ไม่ทราบแน่ชัดว่าข้อมูลตัวเลือกจะมีขอบเขตเท่าใด เช่น ปี พ.ศ. หรือ ปี ค.ศ. ตั้งแต่ อดีตจนถึงปัจจุบันกาก กรณีที่มีข้อมูลเรียงลำดับกันจำนวนมาก เช่น เรียงข้อมูลให้เลือก 1-1000 ข้อมูล เป็นต้น

## การโปรแกรม DropDownList

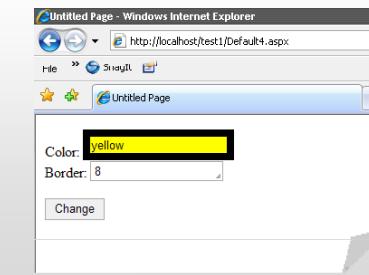
- Syntax ที่สำคัญของ DropDownList มีดังนี้

```
DropDownList1.Items.Add(new ListItem(i.ToString(), i.ToString()));
```

```
Convert.ToInt16(DropDownList1.SelectedItem.Value.ToString());
```

## Test : Event & Properties

- ให้นักศึกษาสร้าง TextBox รับค่าชื่อสี (yellow,red,blue,green) หากรับชื่อสีใดมา ให้ TextBox นั้นเปลี่ยนสีตามข้อมูลที่ป้อน หากพิมพ์ชื่อสีมาไม่ตรงให้เปลี่ยนสี TextBox เป็นสีขาว และสร้าง DropDownList เพื่อเลือกขนาดของ Border (ขอบ)



## Answer : Event & Properties

The screenshot shows the Microsoft Visual Studio IDE with the title bar "test - Microsoft Visual Studio". The code editor displays the Default4.aspx.cs file. The code defines a class Default4 that inherits from System.Web.UI.Page. It contains a Page\_Load event handler that adds items to a dropdown list. A tooltip for the last line of the code is visible: "Appends the specified System.Web.UI.WebControls.ListItem to the end of the collection." The code is as follows:

```
1 using System;
2 using System.Data;
3 using System.Configuration;
4 using System.Collections;
5 using System.Web;
6 using System.Web.Security;
7 using System.Web.UI;
8 using System.Web.UI.WebControls;
9 using System.Web.UI.WebControls.WebParts;
10 using System.Web.UI.HtmlControls;
11 using System.Drawing;
12 public partial class Default4 : System.Web.UI.Page
13 {
14     protected void Page_Load(object sender, EventArgs e)
15     {
16         for (int i = 1; i <= 100; i++)
17         {
18             DropDownList1.Items.Add(new ListItem(i.ToString(), i.ToString()));
19         }
20     }
}
```

## Answer : Event & Properties

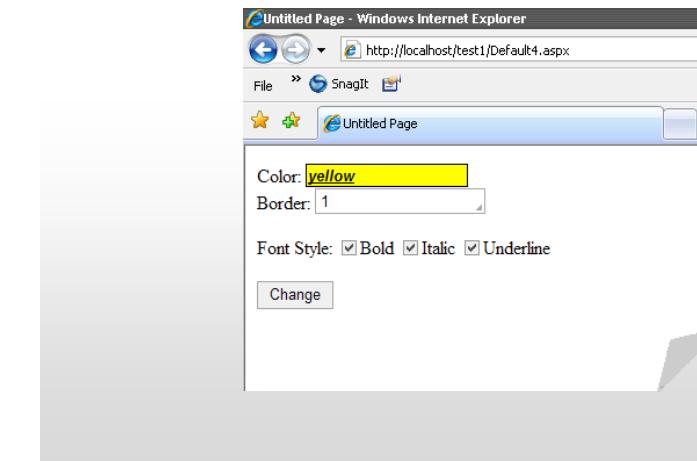
The screenshot shows the Microsoft Visual Studio IDE with the title bar "test - Microsoft Visual Studio". The code editor displays the Default4.aspx.cs file. The code defines a class Default4 that inherits from System.Web.UI.Page. It contains a Button1\_Click event handler that changes the background color of a TextBox based on its text value. The code is as follows:

```
1 protected void Button1_Click(object sender, EventArgs e)
2 {
3     if (TextBox1.Text.ToString() == "red")
4     {
5         TextBox1.BackColor = Color.Red;
6     }
7     else if (TextBox1.Text.ToString() == "blue")
8     {
9         TextBox1.BackColor = Color.Blue;
10    }
11    else if (TextBox1.Text.ToString() == "yellow")
12    {
13        TextBox1.BackColor = Color.Yellow;
14    }
15    else if (TextBox1.Text.ToString() == "green")
16    {
17        TextBox1.BackColor = Color.Green;
18    }
19    else
20    {
21        TextBox1.BackColor = Color.White;
22    }
23    TextBox1.BorderWidth = Convert.ToInt16(DropDownList1.SelectedItem.Value.ToString());
24}
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
```

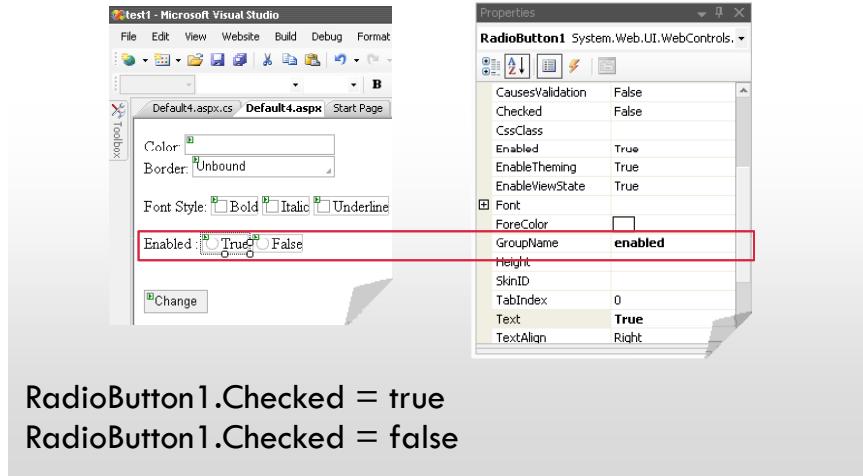
## รู้จักกับ CheckBox

- CheckBox เป็น Control สำหรับเลือกแบบหลายตัวเลือก ข้อดีของ CheckBox คือ เราสามารถเลือกตัวเลือกได้มากกว่า 1 ใช้ในกรณีต้องแบบสอบถามบางอย่างที่สามารถให้เลือกตอบได้มากกว่า 1 ข้อคำตอบ
- Syntax ที่สำคัญของ CheckBox มีดังนี้
  - CheckBox1.Checked = true
  - CheckBox1.Checked = false

## Test : CheckBox

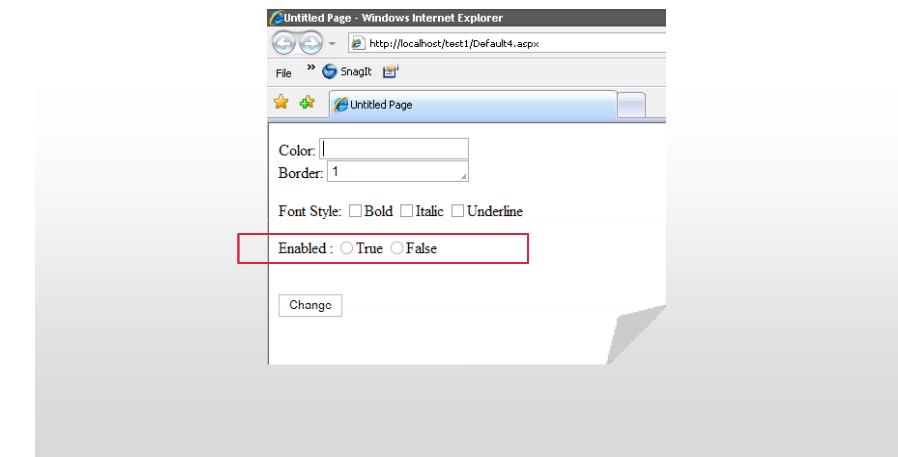


## รู้จักกับ RadioButton



RadioButton1.Checked = true  
RadioButton1.Checked = false

## Test : RadioButton



## รู้จัก Class String

- String คือ ข้อความ โดยใน ASP.NET ตัวแปร String ถือเป็นคลาสที่ มีความสามารถที่หลากหลายและโปรแกรมเมอร์จำเป็นต้องใช้ Class นี้ในการทำงานบ่อยครั้ง เช่น การหาจำนวนของอักษรภายในข้อความ การ ค้นหาคำในข้อความ หรือการตัดคำ เป็นต้น
- String คือ Array ชนิดหนึ่งของ Char
- String เริ่มลำดับอักษรที่ตำแหน่ง 0



## รู้จัก Class String

- ตัวอย่าง Syntax บางตัว
  - การประกาศตัวแปรแบบ string " string temp1; "
  - การประกาศตัวแปรอาร์เรย์แบบ string " string[] temp1; "
  - การหาความยาวของข้อความ  
`Convert.ToString(TextBox1.Text.ToString()).Length.ToString();`

## Demo : Class String

The screenshot shows a Windows application window with a title bar 'Demo : Class String'. Inside, there is a list of String methods with their descriptions and examples. The methods listed are:

- Length หานานดของ String
- IndexOf ค้นหาคำใน String คำตอบจะปรากฎสุดที่ของตัวอักษรแรกที่พบ
- SubString ตัดช่วงมาจ่าตัวนี้ไป โดยใส่เสิร์ฟต้น
- SubString ตัดช่วงมาเรียบฐานนวนอย่าง เริ่มต้น [ ] คำนึงมาที่ต้องการที่สิ้น [ ]
- CompareTo ใช้ในการเปรียบเทียบความแตกต่าง (ไม่เค็มต่าง,- เนตต์ต่าง) [ ]
- Equals ใช้ในการเปรียบเทียบความแตกต่าง กกรณี CaseSensitive (True ไม่เค็มต่าง, False เค็มต่าง) [ ]
- ToLower แปลงเป็นตัวอักษรตัวเล็ก
- ToUpper แปลงเป็นตัวอักษรตัวใหญ่
- TrimStart ใช้ตัดอักษรตัวนำตัวท้าย [ ]
- TrimEnd ใช้ตัดอักษรตัวนำตัวท้าย [ ]
- StartsWith ตรวจสอบว่าตัวหนึ่งต้นท้ายลงกันหรือไม่ [ ]
- EndsWith ตรวจสอบว่าตัวหนึ่งต้นท้ายลงกันหรือไม่ [ ]
- LastIndexOf ตรวจสอบว่าตัวหนึ่งลงอักษรที่ตัวอักษรตัวท้าย หากมีข้ามสือกตัวอักษรตัวท้ายนั้น [ ]
- PadLeft แทรกอักษรระหว่างตัวหน้า ว่างช่อง [ ] และ ให้ข้อความทั้งหมดจานวนครับ [ ] ตัว
- Insert ภารเพิ่มข้อความแทรกเข้าไปใน String แทรกตัวหนึ่งที่ [ ] และ ข้อความที่แทรก [ ] ตัว
- Remove ลบตัวอักษรใน String โดยก้านคุณเอียงตัวแทนที่ต้องการลบ เริ่ม [ ] และ ลบต่อไปจำนวน [ ] อักษร
- Replace การแทนที่ String ด้วยข้อความที่กำหนดใหม่ ข้อความที่จะถูกแทนที่ [ ] และ ข้อความที่แทนที่ [ ]
- Split ตัวการแยก String ออกเป็นส่วนๆ แยกด้วยรอยต่อ [ ]

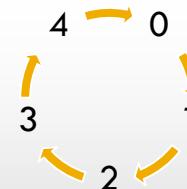
At the bottom left is a 'Submit' button.

## Demo : Class String

```
18 protected void Button1_Click(object sender, EventArgs e)
19 {
20     if (RadioButton1.Checked == true)
21     {
22         TextBox2.Text = Convert.ToString(textBox1.Text.ToString().Length.ToString());
23     }
24     else if (RadioButton2.Checked == true)
25     {
26         TextBox2.Text = Convert.ToString(textBox1.Text.ToString()).IndexOf(textBox3.Text.ToString());
27     }
28     else if (RadioButton3.Checked == true)
29     {
30         TextBox2.Text = Convert.ToString(textBox1.Text.ToString()).Substring(Convert.ToInt16(textBox4.Text));
31     }
32     else if (RadioButton4.Checked == true)
33     {
34         TextBox2.Text = Convert.ToString(textBox1.Text.ToString()).CompareTo(textBox5.Text).ToString();
35     }
36     else if (RadioButton5.Checked == true)
37     {
38         TextBox2.Text = Convert.ToString(textBox1.Text.ToString()).Equals(textBox6.Text).ToString();
39     }
40     else if (RadioButton6.Checked == true)
41     {
42         TextBox2.Text = Convert.ToString(textBox1.Text.ToString()).ToLower().ToString();
43     }
44     else if (RadioButton7.Checked == true)
45     {
46         TextBox2.Text = Convert.ToString(textBox1.Text.ToString()).ToUpper().ToString();
47     }
}
```

## Demo : Class String

```
string values = "TechRepublic.com, CNET.com,  
News.com, Builder.com, GameSpot.com";  
string[] sites = values.Split(',');  
foreach (string s in sites)  
{  
    Response.Write(s.ToString());  
}
```

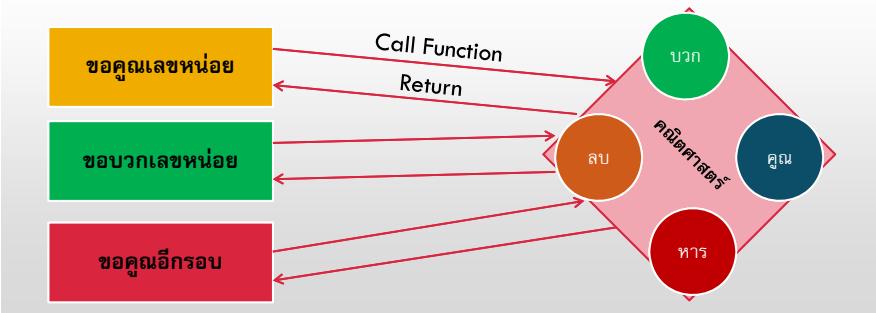


## Test : Class String

- 1. ทำการ Copy บทความภาษาอังกฤษจากเว็บได้ แล้วเขียนโปรแกรมให้ว่ามีตัวอักษร A อยู่ในบทความนั้นเกี่ยวกับ A
- 2. ทำการ Copy บทความภาษาอังกฤษ โดยเขียนโปรแกรมแยกคำทุกตัว หากแต่ละคำอยู่ห่างกัน 1 Space และแสดงผลคำทุกคำที่แยกและนับว่ามีกี่คำ

## การสร้าง Functions

- พังก์ชัน Functions คือ กระบวนการทำงานอย่างหนึ่งในการเขียนโปรแกรม โดยข้อดีของการทำพังก์ชันคือการเรียกใช้ซ้ำได้ ทำให้การเขียนโปรแกรมสั้นลงและในอนาคตหากมีการแก้ไขหรือปรับปรุงกระบวนการ ก็สามารถทำได้สะดวกและรวดเร็วกว่าการเขียนโปรแกรมแบบทั่วไป



## การสร้าง Functions

- Syntax สำหรับสร้าง Function มีดังนี้

```
protected decimal add1(decimal a,decimal b)
```

```
{
```

```
    decimal temp = a + b;
```

```
    return temp;
```

```
}
```

Arguments

Parameters

วิธีการเรียกใช้

```
decimal Var1 = add1(12,8);
```

## Test1 : Function

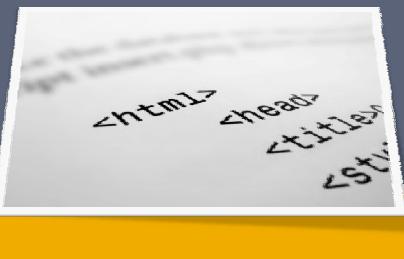
- ให้เขียนโปรแกรมคำนวนหาอายุโดยใช้ ปี ค.ศ. ของพนักงานบริษัทแห่งหนึ่ง จำนวน 10 คน และแสดงผลพร้อมกัน
- ให้เขียนโปรแกรมคำนวนหาพื้นที่สามเหลี่ยมโดยให้ฐานเพิ่มที่ละ 5 ซม และส่วนสูงเพิ่มที่ละ 8 ซม จำนวน 10 รูป
- ให้เขียนโปรแกรมคำนวนอุณหภูมิของอากาศแบบองศาเซลเซียสของประเทศไทยต่างๆ จำนวน 10 ประเทศ หากอุณหภูมิอยู่ในช่วง 0-10 องศาอากาศหนาว แสดงสีม่วง , 11-20 องศาอากาศเย็น แสดงสีชมพูเข้ม , 21-30 องศา อากาศค่อนข้างเย็น แสดงสีชมพูอ่อน , 31-37 องศาอากาศร้อน แสดงสีแดง อากาศร้อน แสดงสีส้ม , 38-45 องศาอากาศร้อน แสดงสีแดง

## Homework1

- ให้ Copy ข้อความภาษาอังกฤษอย่างน้อย 10 บรรทัด และเขียนโปรแกรมแปลงคัพท์จากภาษาอังกฤษเป็นภาษาไทย
- ให้นักศึกษาไปค้นคว้าหา Event Handle ของ Control ที่นักศึกษาสนใจ คงจะ 4 Controls โดยอิบายถึงการประกาศ Event นั้นๆ และให้คำนิยามเกี่ยวกับ Event นั้นๆ และให้เหตุผลว่า Event นี้มีความสัมพันธ์อย่างไรและสามารถประยุกต์ใช้งานอย่างไรกับ Controls นี้
- ระบุชื่อและส่งเป็น PDF งาน มาที่ [nutthapat.k@hotmail.com](mailto:nutthapat.k@hotmail.com) ภายในวันที่ 8 กุมภาพันธ์ 2552

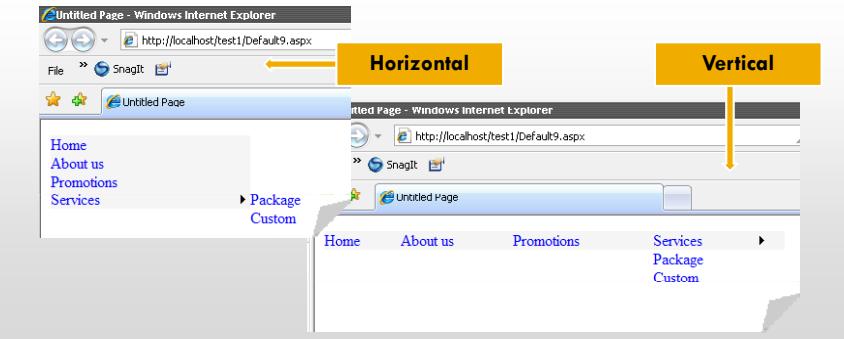
## ASP.NET WITH C# CLASS 3

อ.ณัฐภรณ์ แก้วรัตนภรณ์

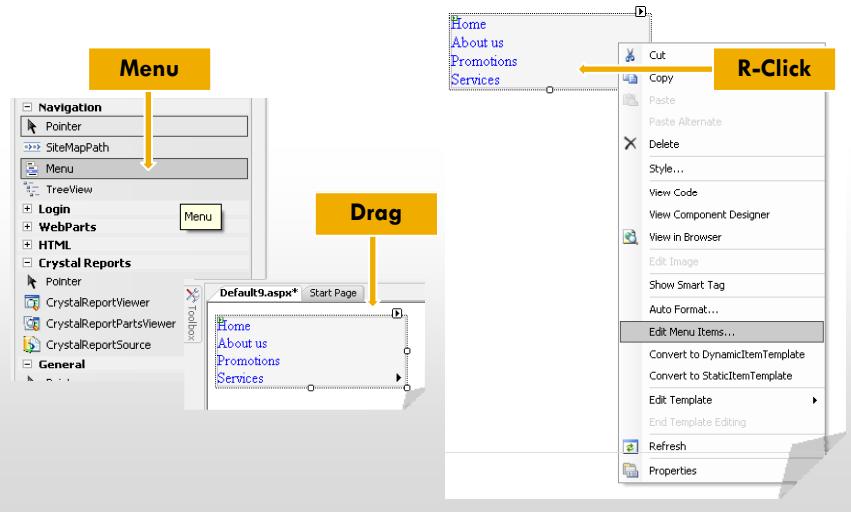


## Navigation : MENU

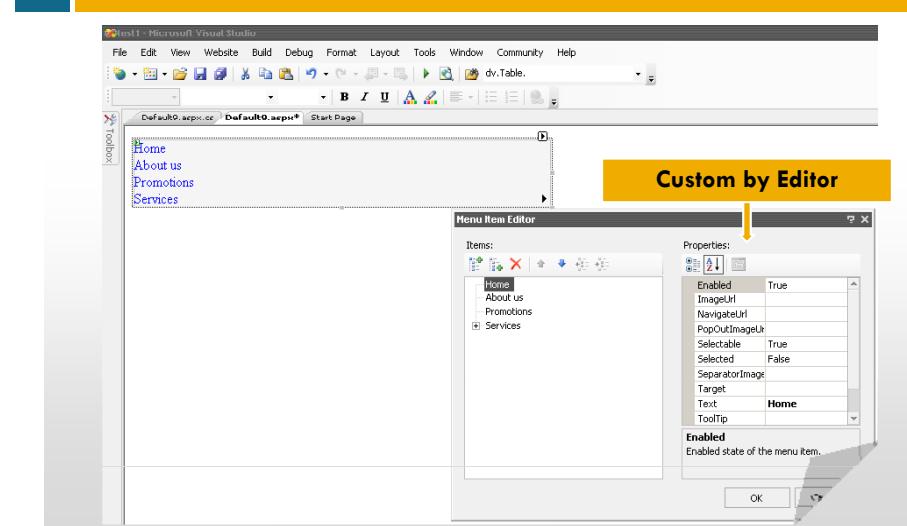
- **Menu Server Control** เป็น Control สำหรับการจัดการกับส่วนเมนู ซึ่งสามารถหยิบมาใช้งานได้ง่ายและสะดวก รวมถึงการปรับแต่งให้เหมาะสมกับแอปพลิเคชันได้อย่างสะดวกและรวดเร็ว



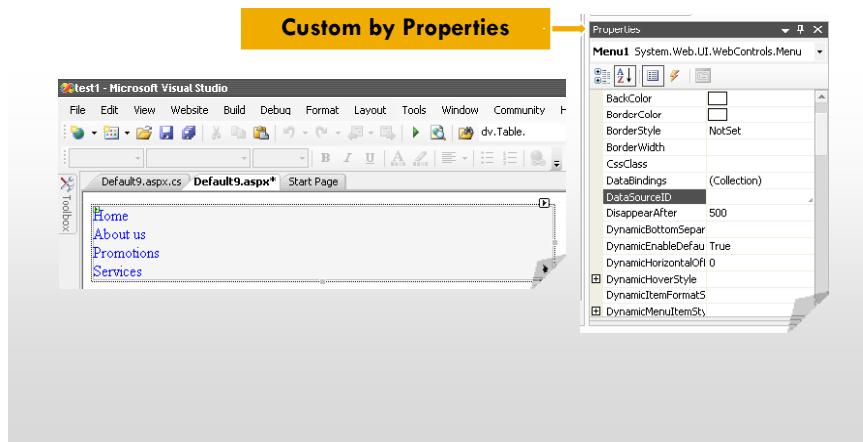
## DEMO : Navigation : MENU



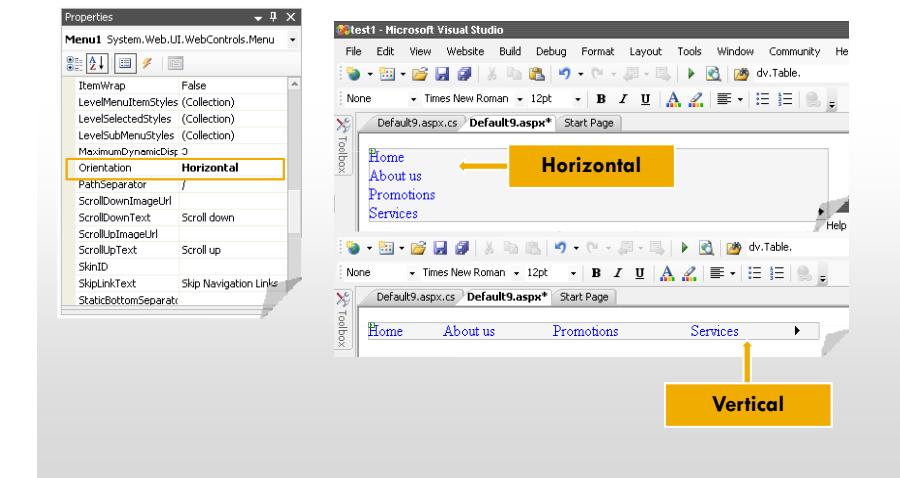
## DEMO : Navigation : MENU



## DEMO : Navigation : MENU



## DEMO : Navigation : MENU



## Programming in Menu

- ทุกๆ Server Control สามารถที่จะใช้วิธีการเขียนโปรแกรมเพื่อลดข้อจำกัด ทางด้านข้อมูลที่จะปรับปรุงให้แก่ Control ได้ ทำให้เราสามารถทำการ โปรแกรมแบบ Dynamic ได้อย่างอิสระมากยิ่งขึ้นและเกิดความหลากหลาย ของแอ็พพลิเคชันด้วย

### ตัวอย่าง Syntax ของ Menu Server Control

```
□ Menu1.Items.Add(new MenuItem("Contact us"));
□ Menu1.FindItem("Contact us").ChildItems.Add(new MenuItem("เพิ่มใหม่"));
□ Menu1.FindItem("Contact us/เพิ่มใหม่").ChildItems.Add(new MenuItem("เพิ่มใหม่2"));
□ Menu1.Items.Remove(Menu1.FindItem("Contact us"));
□ Menu1.Items.Clear();
□ Menu1.FindItem("Services").ChildItems.Clear();
□ Menu1.FindItem("Services").ChildItems.Remove(Menu1.FindItem("Services/Custom"));
□ Menu1.FindItem("Services/Custom").ChildItems.Remove(Menu1.FindItem("Services/Custom/del"));
```

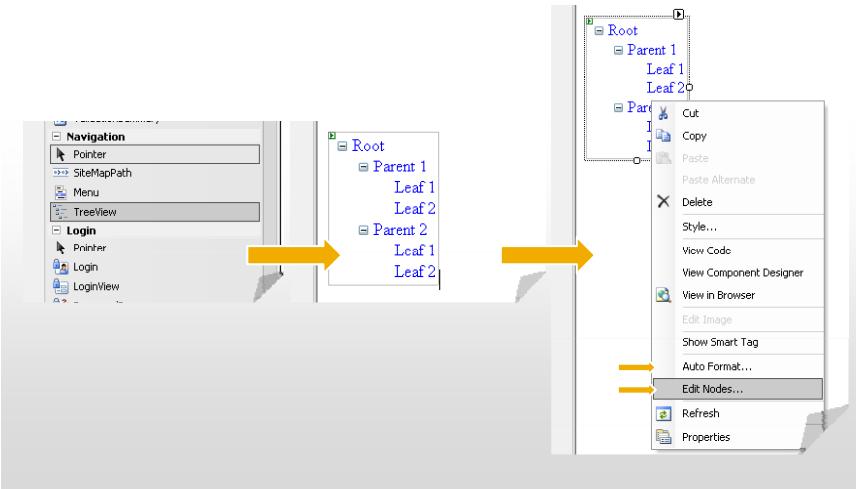


## Test : Menu

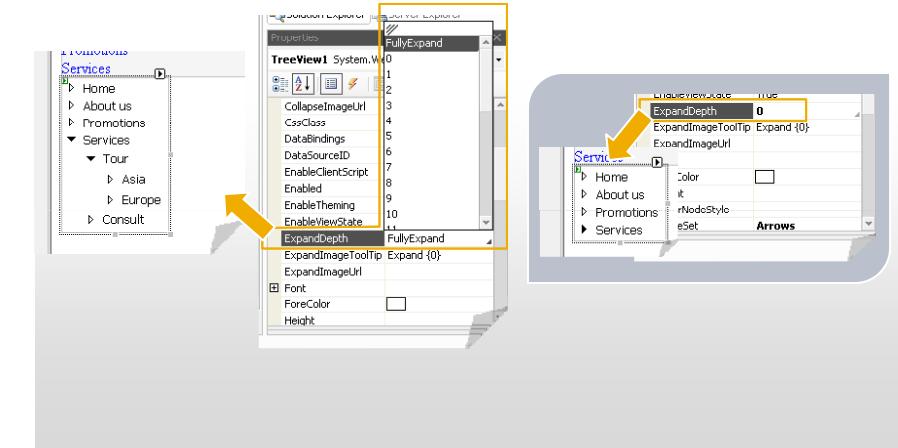
- ให้สร้าง Menu แบบ Static โดยมีข้อมูลดังนี้
  - Animal
    - Terrestrial Animal
    - Aquatic Animal
    - Poultry
  - Plant
    - Unicellular Plant
    - Multi-Cellular Plant
- จากนั้นให้ทำการเขียนโปรแกรม (Dynamic) รับชื่อสัตว์โดยใช้ Textbox และทำ DropDownList ให้เลือกประเภท 3 ประเภท เพื่อเพิ่มสัตว์ไว้หัตถ ประเภท เช่น หากเป็นสัตว์บกให้เพิ่มเข้าที่เมนู Terrestrial หากเป็นสัตว์น้ำ เพิ่มใน Aquatic และสัตว์ปีกให้เพิ่มเข้า Poultry



## Navigation : TREE VIEW



## TREE VIEW : ExpandDepth

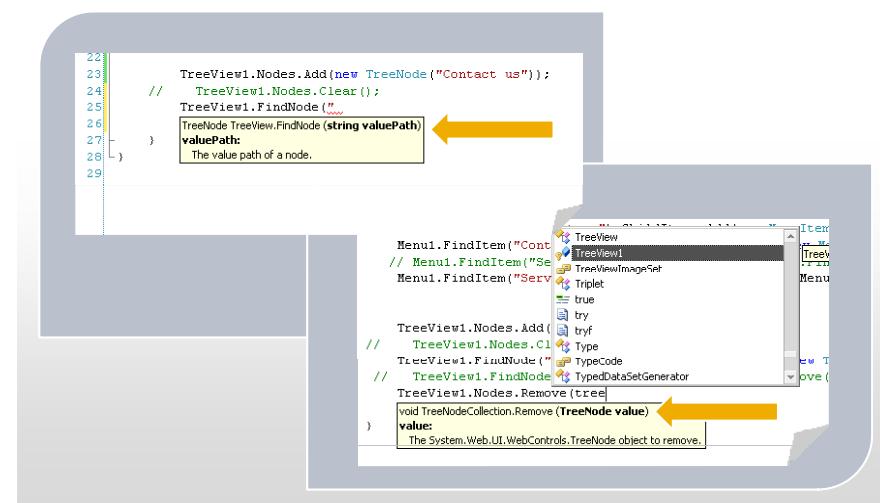


## TREE VIEW

### □ ตัวอย่าง Syntax ที่สำคัญของ TreeView

- `TreeView1.Nodes.Add(new TreeNode("Contact us"));`
- `TreeView1.Nodes.Remove(TreeView1.FindNode("Services"));`
- `TreeView1.Nodes.Clear();`
- `TreeView1.FindNode("Services/Tour").ChildNodes.Add(new TreeNode("เพิ่มใหม่"));`
- `TreeView1.FindNode("Services/Tour").ChildNodes.Remove(TreeView1.FindNode("Services/Tour/Asia"));`
- `TreeView1.FindNode("Services").ChildNodes.Clear();`

## TREE VIEW



## Test : Tree

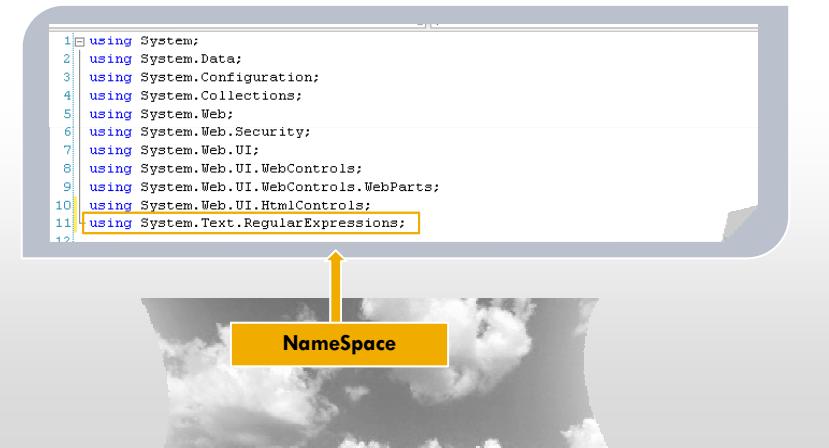
### □ นำโจทย์จาก Menu มาแก้ไขเป็น Tree

- ให้สร้าง Tree แบบ Static โดยมีข้อมูลดังนี้
  - Animal
    - Terrestrial Animal
    - Aquatic Animal
    - Poultry
  - Plant
    - Unicellular Plant
    - Multi-Cellular Plant
- จากนั้นให้ทำการเขียนโปรแกรม (Dynamic) รับชื่อสัตว์โดยใช้ Textbox และทำ DropDownList ให้เลือกประเภท 3 ประเภท เพื่อเพิ่มสัตว์ให้ตรงประเภท เช่น หาก เป็นสัตว์บกให้เพิ่มเข้าที่เมนู Terrestrial หากเป็นสัตว์น้ำเพิ่มใน Aquatic และสัตว์ ปีกให้เพิ่มเข้า Poultry

## Regular Expression to Constrain input

- Regular Expression คือ รูปแบบชนิดหนึ่งที่ใช้สำหรับกำหนดรูปแบบ ของคำหรือข้อความหรืออักษร โดยรูปแบบจะเป็นอักษรที่มีความหมาย สามารถเปลี่ยนไปเรื่อยๆได้
- Regular Expression มีข้อดี คือ ใช้ในการตรวจสอบอักษรที่จำเป็น หรือไม่จำเป็นใช้งาน เช่น
  - ใช้สำหรับตรวจสอบข้อมูลที่ได้รับจาก textbox ว่าเป็นเฉพาะตัวเลขหรือไม่
  - ตรวจสอบข้อมูลว่าถูกต้องตามหลักการการตั้งชื่อไฟล์หรือชื่อผู้ใช้ติดต่อ ระหว่างผู้คนหรือไม่
  - หรือใช้สำหรับตรวจสอบการตั้งชื่อไฟล์ข้อมูล เป็นต้น

## Regular Expression : NameSpace



## Regular Expression Characters

- ตัวอย่างอักษร Regular Expression ที่ใช้เป็นประจำ
  - @ คือ การบอกจุดเริ่มต้นให้แก่ Expression เช่น new Regex(@"\dNum");
  - \* คือ ไม่มีหรือมีจำนวนเท่าใดก็ได้ เช่น mic\*key เป็น mickey หรือ miccckkey ได้
  - + คือ มีอย่างน้อย 1 อักษร และจะมากเท่าใดก็ได้ เช่น mick\*ey เป็น mickkey หรือ mickkkkkkey
  - ? คือ ไม่มีหรือมีเพียงอักษระเดียวเท่านั้น เช่น mic?key เป็น mickey หรือ mickey ได้เท่านั้น
  - | คือ ใช้สำหรับเลือกอย่างใดอย่างหนึ่ง เช่น (A | B)?C เป็น C , AC หรือ BC
  - ( ) คือ การจัดกลุ่มและจะต้องปรากฏอย่างน้อย 1 อักษร เช่น (A | B)C เป็น AC หรือ BC
  - . คือ การแทนตัวอักษร 1 ตัว ยกเว้นตัวอักษรที่ใช้ขั้นบรรทัดใหม่ เช่น PO.23 เป็น PO.P23
  - [ ] คือ การแทนการเรียงอักษร เช่น [A-C] เป็น A หรือ B หรือ C

## Regular Expression Characters ,Cont

- **\d** คือ การแทนตัวเลขได้ เช่น NUM\d เป็น NUM123 หรือ NUM9 เป็นต้น
- **\D** คือ การแทนตัวอักษรใดๆที่ไม่ใช่ตัวเลข เช่น \D191 เป็น a191 หรือ mynum191 เป็นต้น
- **\n** ใช้สำหรับข้อบรรทัดใหม่
- **\w** ใช้แทนอักษรทุกตัว ได้แก่ 0-9 ,a-z,A-Z
- **\W** ใช้แทนอักษรทุกตัว ยกเว้น 0-9 ,a-z,A-Z
- **[ ^ ]** คือ การยกเว้นอักษรที่อยู่ด้านหลัง เชื่อ [^0-9] คือ การยกเว้นตัวเลขทุกตัว แต่ หากใช้แบบโดดๆ คือ ^[0-9] หมายความว่า ห้ามเป็นค่าว่าง(Null) ซึ่งจะต้องเป็นตัวเลข 0-9 ตัวใดก็ได้ หรือ หมายความว่าเป็นจุดเริ่มต้นของข้อความ
- **{min,max}** คือ การกำหนดค่าจำนวนน้อยที่สุดและมากที่สุดของตัวอักษร เช่น [A-Z]{2,4} คือ อักษรที่เป็นไปได้คือ A ถึง Z และต้องมีจำนวนน้อยที่สุด 2 ตัว อักษร และมากที่สุดไม่เกิน 3 อักษร เช่น AC,ABC
- **\** คือ การระบุว่าอักษรที่อยู่หลังสัญลักษณ์นี้เป็นอักษรหนึ่งเท่านั้น ไม่มีนายที่พ่อจะเป็น Regular Expression เช่น \? จะหมายถึงอักษร ? ซึ่งไม่ใช่สิ่งนี้ A

PDF

## Regular Expression Characters ,Cont

- ตัวอย่าง Syntax สำหรับตรวจสอบกับค่าที่รับเข้ามา

```
Regex rx = new Regex("[a-z]");  
bool ch = rx.IsMatch(textBox1.Text.ToString());  
Response.Write(ch.ToString());
```



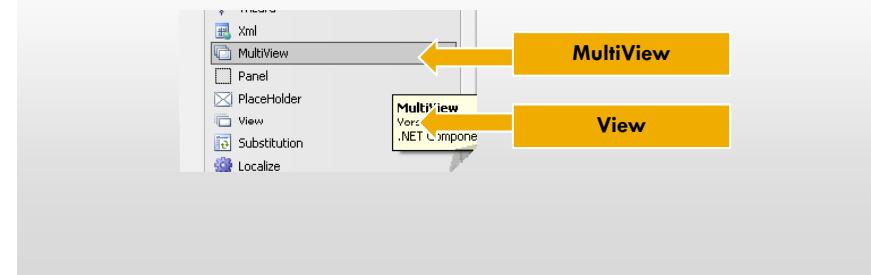
## Test : Regular Expression Characters

- ให้สร้างตัวกรองแยกเพศชายเพศหญิง
  - นาย
  - นางสาว
  - นาง
- ให้สร้างตัวกรองอีเมลล์



## Multiview & View Server Controls

- MultiView คือ กรอบหน้าต่างที่รวบรวม View Control ซึ่งข้อดีของ View คือ เรายสามารถเขียนโปรแกรมควบคุมการแสดงผลข้อมูลให้ตรงต่อความต้องการของระดับผู้ใช้แต่ละคนได้
- Syntax ที่ใช้บ่อย คือ MultiView1.SetActiveView(View1);

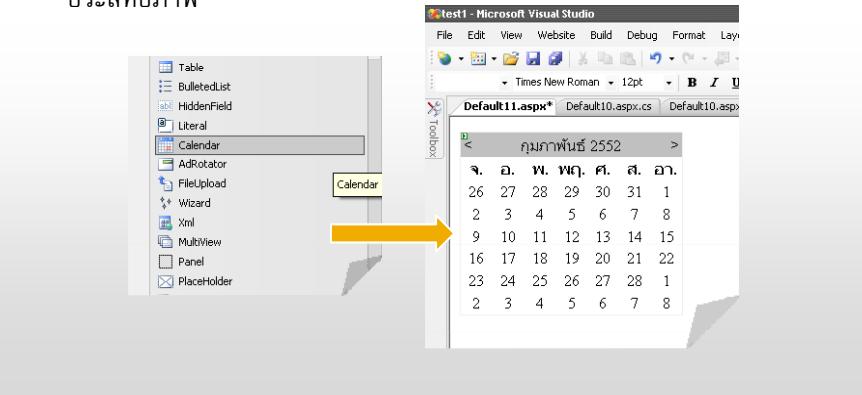


## Test : Multiview & View Server Controls

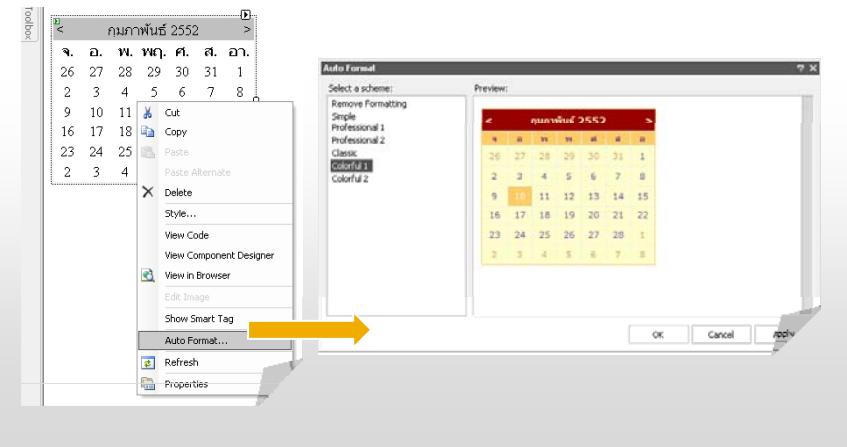
- มีข้อมูลดังนี้
  - Mr.John Smith
  - Miss.Nicole Kidman
  - Mrs.Jenifer Hudson
- ให้ทำการรับข้อมูลดังกล่าว โดยแยกประเภทของสถานะ เช่น หากเป็นผู้ชาย  
ให้แสดง View ที่เหมาะสมกับผู้ชาย

## Calendar Server Control

- คือ Server Control ที่ใช้สำหรับทำปฏิวัติในช่องทำให้ผู้แพฒนาแอ็พพลิเคชันมีความสะดวกรวดเร็ว อีกทั้งสามารถทำการโปรแกรมไปที่ตัวคอนโทรลได้อย่างมีประสิทธิภาพ



## Calendar Server Control



## Demo : Calendar Programming

```
26/1/2552 0:00:00
27/1/2552 0:00:00
28/1/2552 0:00:00
29/1/2552 0:00:00
30/1/2552 0:00:00
31/1/2552 0:00:00
1/2/2552 0:00:00
2/2/2552 0:00:00
3/2/2552 0:00:00
4/2/2552 0:00:00
5/2/2552 0:00:00
6/2/2552 0:00:00
7/2/2552 0:00:00
8/2/2552 0:00:00
9/2/2552 0:00:00
10/2/2552 0:00:00
11/2/2552 0:00:00
12/2/2552 0:00:00
13/2/2552 0:00:00
14/2/2552 0:00:00
15/2/2552 0:00:00
16/2/2552 0:00:00
17/2/2552 0:00:00
18/2/2552 0:00:00

Event : DayRender
All Render
```

The screenshot shows a code editor in Microsoft Visual Studio with C# code for a 'Default11' page. The code includes an 'Event : DayRender' block and an 'All Render' block. The 'DayRender' block contains logic to set the active view and write the day's date to the response. The code editor has syntax highlighting and line numbers. To the right of the code, a list of dates from January 26 to February 18, 2009, is displayed.

## Demo : Calendar Programming

Each Render

```
protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
{
    Response.Write(e.Day.Date.ToString());
    if (e.Day.Date.ToString() == DateTime.Now.ToString())
    {
        e.Cell.Text = "Today";
    }
    if (e.Day.Date.ToString() == "27/2/2552")
    {
        e.Cell.Text = "Get salary";
    }
    if (e.Day.Date.ToString() == "8/3/2552")
    {
        e.Cell.Text = "My Birthday";
    }
}

protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    Response.Write(Calendar1.SelectedDate.ToString());
}
```

## Test : Calendar

- วันที่ 3 กพ 2552 เป็นวันเกิด John Smith และทำสีวันเป็นสีเขียว
- วันที่ 16 กพ 2552 เป็นวันรับปริญญา ทำสีแดง
- วันที่ 30 กพ 2552 จ่ายเงินค่าค่อนโดย ทำสีม่วง
- วันที่ 28 มีค 2552 สอบป้ายภาค ทำสีแดง
- ช่วงวันที่ 1-15 มีค 2552 ไปเที่ยวภูเก็ต ทำสีส้ม

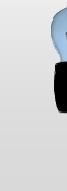


## Test : Calendar

กุมภาพันธ์ 2552								
≤	≤	≤	≤	≤	≤	≤	≤	≥
≤	26	27	28	29	30	31	1	≥
≥	2	3	4	5	6	7	8	≥
≥	9	Today	11	12	13	14	15	≥
≥	16	17	18	19	20	21	22	≥
≥	23	24	25	26	Get salary	28	Phuket	≥
≥	Phuket	Phuket	Phuket	Phuket	Phuket	Phuket	Phuket	≥

## Some Answer

```
protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
{
    Response.Write(i.ToString() + "/2/2552");
    if (e.Day.Date.ToString() == DateTime.Now.ToString())
    {
        e.Cell.Text = "Today";
    }
    if (e.Day.Date.ToString() == "27/2/2552")
    {
        e.Cell.Text = "Get salary";
    }
    if (e.Day.Date.ToString() == "8/3/2552")
    {
        e.Cell.Text = "My Birthday";
    }
    if (i <= 15)
    {
        if (e.Day.Date.ToString() == i.ToString() + "/3/2552")
        {
            e.Cell.Text = "Phuket";
            i++;
        }
    }
}
```



## Java Script : POP UP

- ตัวอย่างการแทรก Java Script ทำ pop up windows

```
string popupScript = "<script language='javascript'>" +  
    "function pop2(myurl){" +  
    "window.open(myurl, 'CustomPopUp', " +  
    "width=660 height=480 scrollbar=yes')}" +  
    "</script>";  
  
Page.RegisterStartupScript("PopupScript3", popupScript);  
  
Label5.Text = "<a href='##' onclick=pop2('aboutus.aspx') class='A1' >" + "About us" +  
    "</a>";
```

JAVA

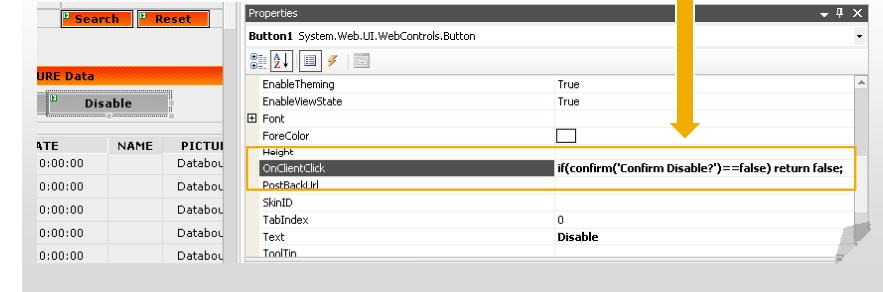
Use

Name of JavaScript

## Java Script : Alert MSG & Confirm

- Response.Write("<script language='javascript'>alert('Please Login!'); window.location='home.aspx';</script>");

Alert Msg

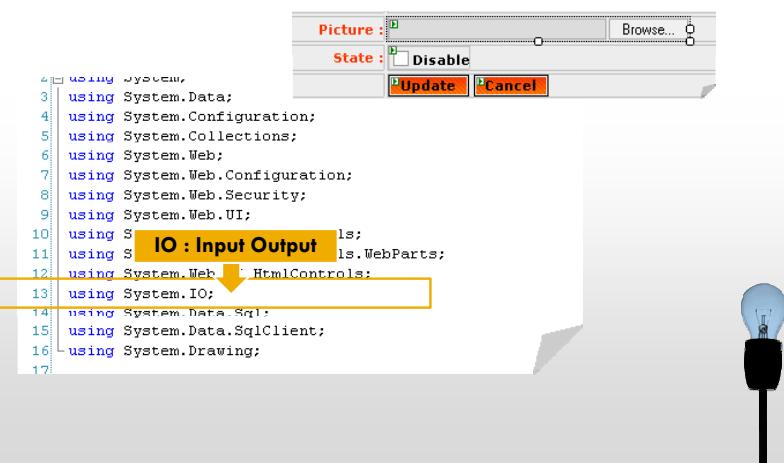


Confirm Box

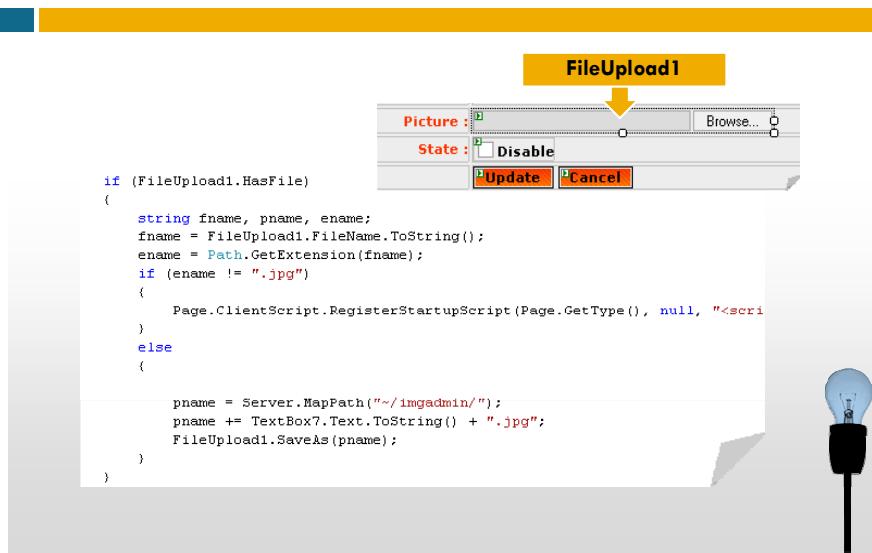
## Test : Msg Box

- ให้รับค่า Username และ Password หากถูกต้องให้แสดง MsgBox แสดงข้อความ “Correct” หากผิดแสดงข้อความ “Incorrect”
- ให้รับค่าหมายเลขบัตรประชาชน หากไม่ถึง 13 หลัก ให้แสดง MsgBox เตือนให้ผู้ใช้กรอกให้ครบ และหากไม่ได้กรอกเป็นตัวเลขทีให้แสดงเตือนให้กรอกเฉพาะตัวเลขเท่านั้น

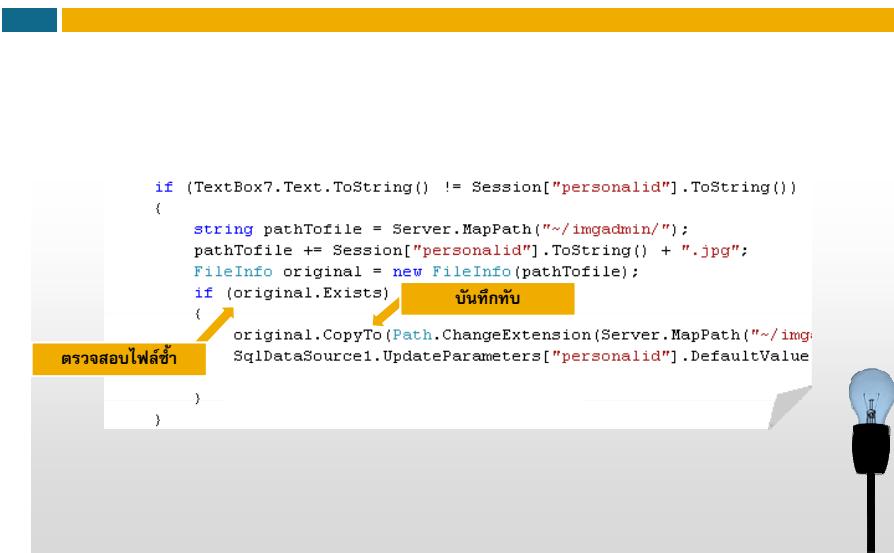
## File Upload Server Control



## File Upload Server Control



## File Upload Server Control



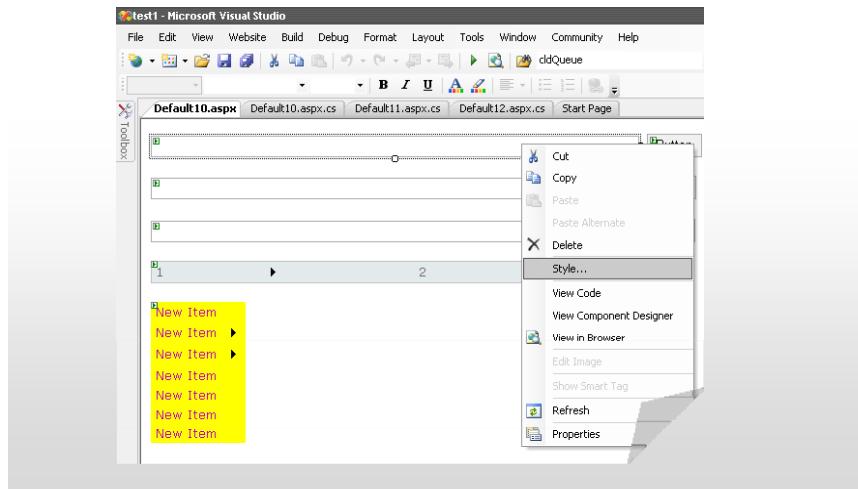
## Test : Upload File

- ให้สร้างตัวอัพโหลดไฟล์ที่อัพโหลดได้เฉพาะนามสกุล \*.txt เท่านั้น
- ให้สร้างตัวอัพโหลดไฟล์เฉพาะนามสกุล \*.gif เท่านั้น
- ให้สร้างตัวอัพโหลดไฟล์ที่ละ 5 ตัว พร้อมกัน โดยเฉพาะไฟล์นามสกุล \*.jpg , \*.JPG , \*.jpeg และ \*.JPEG เท่านั้น

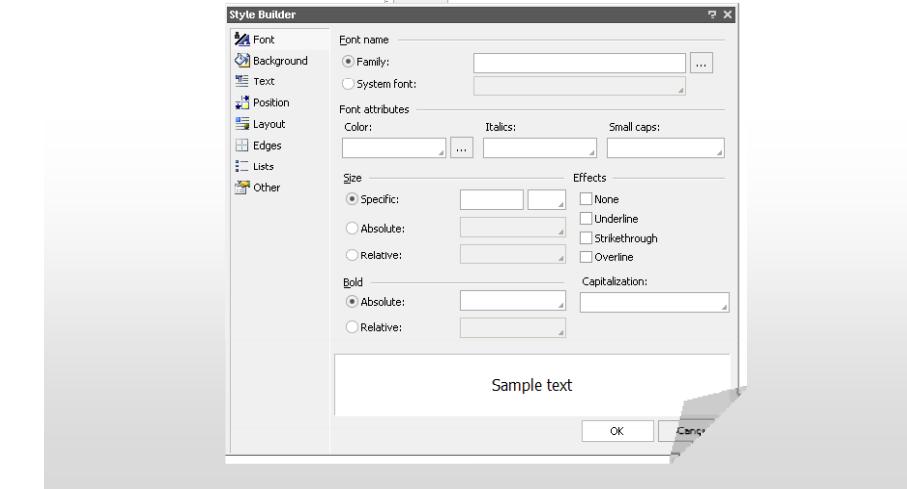
## CSS : Cascade Style Sheet

- คือ คำสั่งที่ใช้ในการกำหนดการแสดงผลของเว็บไซต์ ชื่่ง CSS จะช่วยให้เราสามารถตกแต่งหน้าเว็บหลายๆ เว็บได้พร้อมกัน โดยไม่ต้องทำการกำหนดใหม่ทำให้ได้ **Theme** ของเว็บเดียวกัน
- ข้อดีคือประยุกต์เวลาในการกำหนดการแสดงผล สามารถแก้ไขปรับปรุงการแสดงผลได้อย่างรวดเร็วและมีประสิทธิภาพ อีกทั้งยังสามารถทำส่วนการแสดงผลประเภท **Dynamic** ได้ด้วย เช่น เมื่อกดเหตุการณ์ทางเม้าส์บน **Control** เป็นต้น

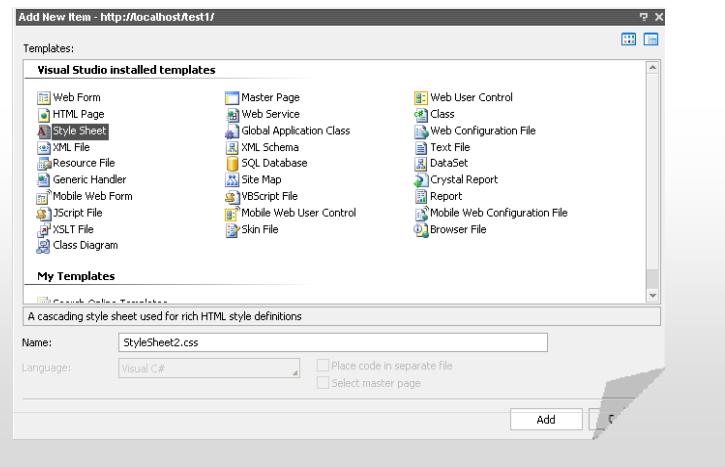
## Demo : CSS



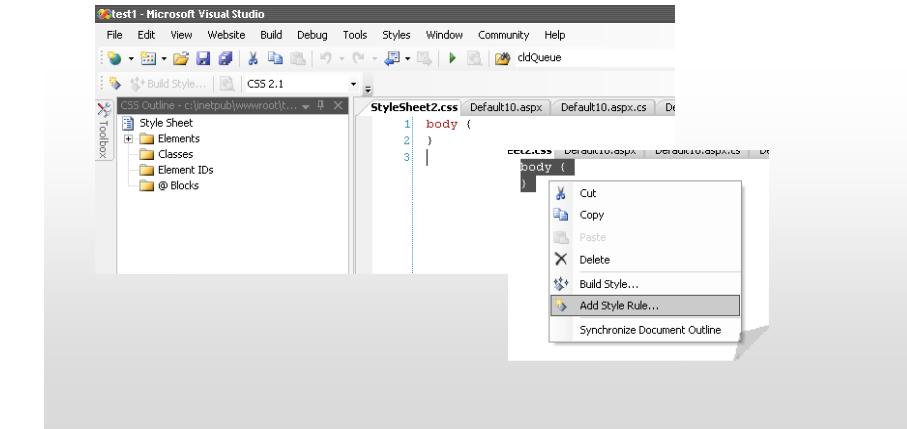
## Demo : CSS



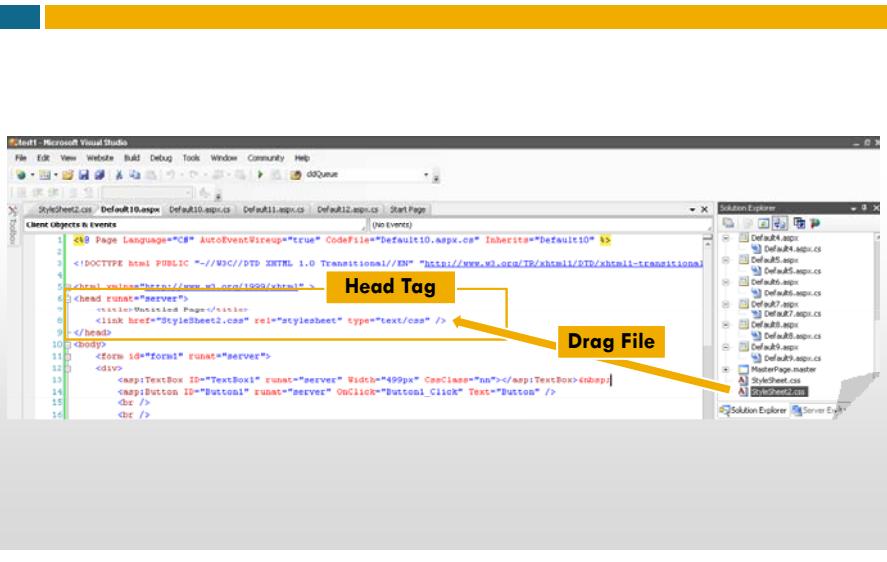
## Create Style Sheet File



## Create Style Sheet File

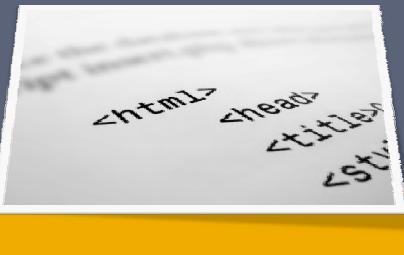


# Insert Style Sheet File



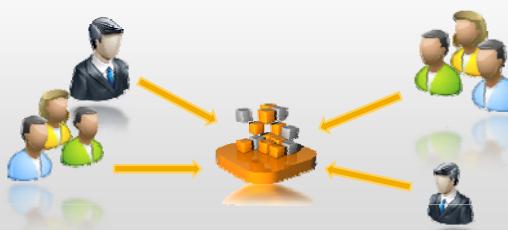
## ASP.NET WITH C# CLASS 4

อ.ณัฐภรณ์ แก้วรัตนภรณ์



## SQL Server 2005

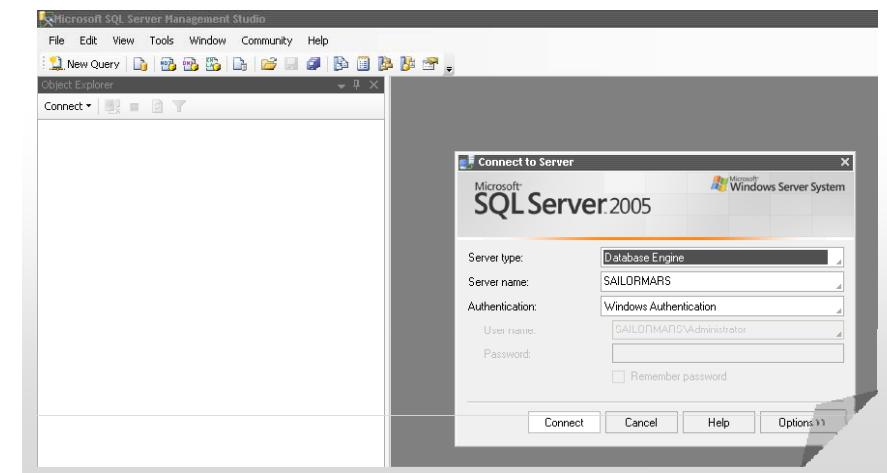
- SQL Server เป็นของบริษัทไมโครซอฟต์ โดยเป็นระบบการจัดการฐานข้อมูล (DBMS) ขนาดใหญ่แบบ Online ซึ่งเปรียบเสมือน Server Database ที่ให้บริการเข้ามต่อและให้บริการฐานข้อมูล และ SQL Server เหมาะกับการทำงานที่รองรับผู้ใช้งานจำนวนมาก เช่น ใช้ทำ Web-Based Application เป็นต้น



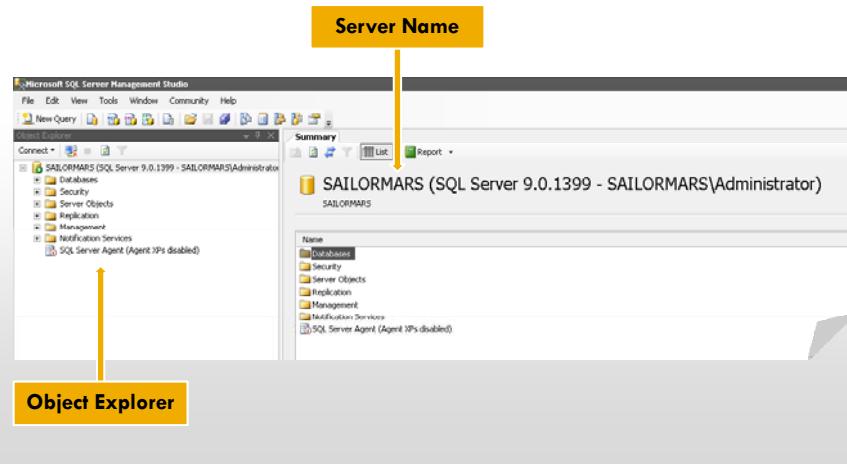
## SQL Server 2005



## Demo : SQL Server 2005



## Demo : SQL Server 2005



## Topics : SQL Server 2005

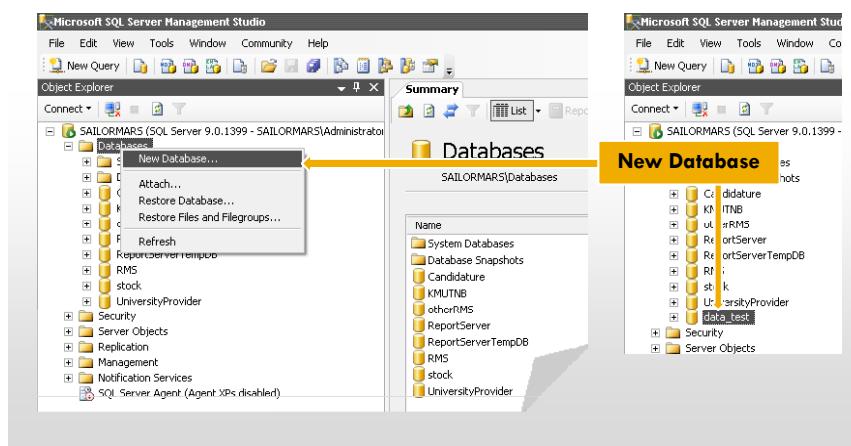
- การสร้างฐานข้อมูล (Create Database)
- การแก้ไข/ลบ ฐานข้อมูล (Update & Delete DB)
- การสร้างตาราง (Create Table)
- การแก้ไข/ลบ ตาราง (Update & Delete Table)
- การสร้างแอ็ตทริบิวต์ (Create Attribute)
- การแก้ไข/ลบ แอ็ตทริบิวต์ (Update & Delete Attribute)
- การกำหนดคีย์ (Primary Key)
- การจัดการข้อมูล (Data Management)
- การสร้างความสัมพันธ์ (Relationship Diagram)

## Topics : SQL Server 2005

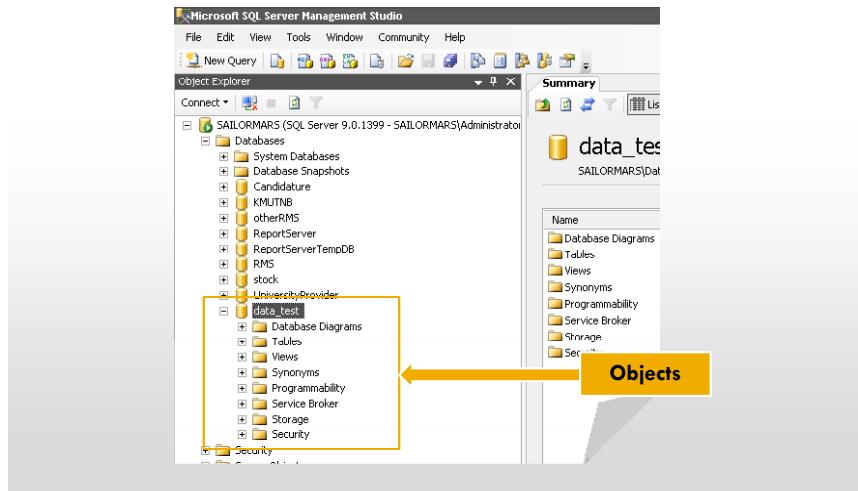
- การสืบค้นข้อมูล (Data Retrieval / SQL)
- การใช้ SQL Designer
- รู้จักการทำ PL/SQL เป็นต้น



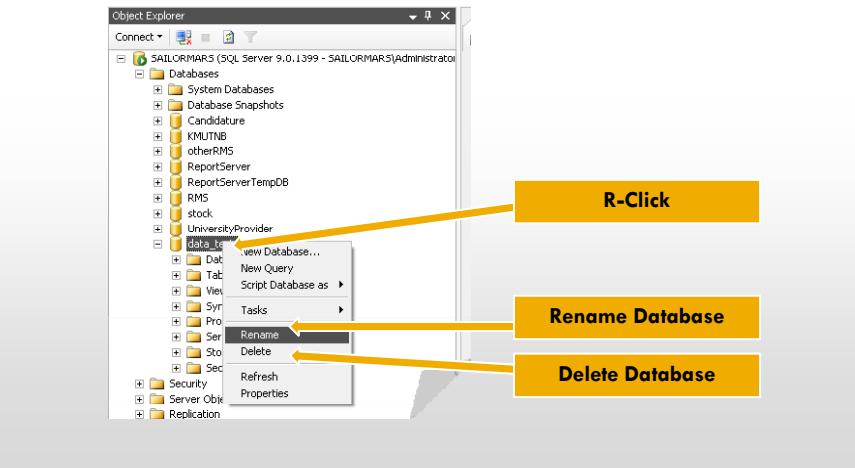
## Create Database



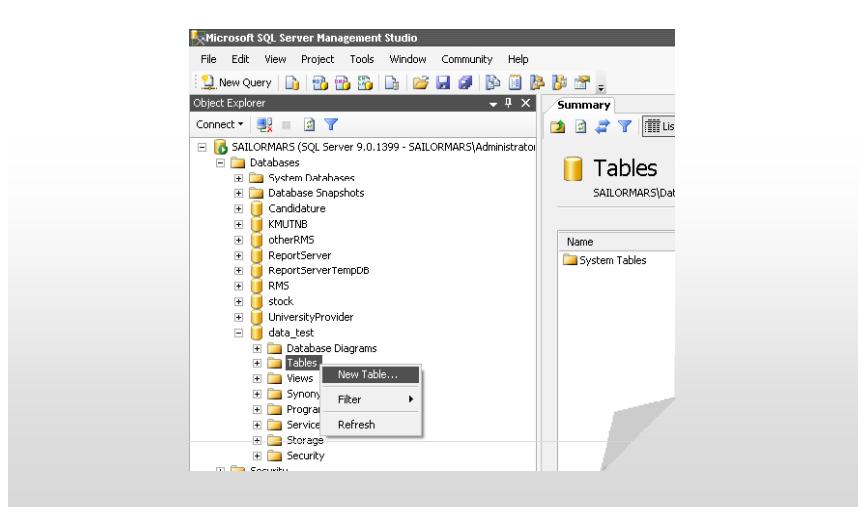
## Create Database



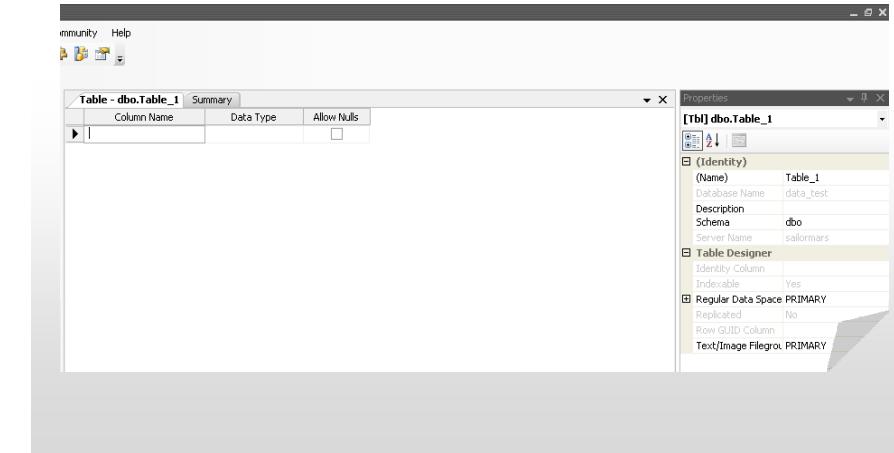
## Rename & Delete Database



## Create Table



## Create Table



# Data Type

Data Type Name	Class	Size in Bytes	Nature of the Data
Bit	Integer	1	The size is somewhat misleading. The first bit data type in a table takes up one byte; the next seven make use of the same byte. Allowing nulls causes an additional byte to be used.
Bigint	Integer	8	This just deals with the fact that we use larger and larger numbers on a more frequent basis. This one allows you to use whole numbers from $-2^{63}$ to $2^{63}-1$ . That's plus or minus about 92 quintillion.
Int	Integer	4	Whole numbers from $-2,147,483,648$ to $2,147,483,647$ .
SmallInt	Integer	2	Whole numbers from $-32,768$ to $32,767$ .
TinyInt	Integer	1	Whole numbers from 0 to 255.
Decimal or Numeric	Decimal/ Numeric	Varies	Fixed precision and scale from $-10^{38}-1$ to $10^{38}-1$ . The two names are synonymous.

# Data Type

Data Type Name	Class	Size in Bytes	Nature of the Data
SmallMoney	Money	4	Monetary units from $-214,748.3648$ to $+214,748.3647$ .
Float (also a synonym for ANSI Real)	Approximate Numerics	Varies	Accepts an argument (for example, <code>Float(20)</code> ) that determines size and precision. Note that the argument is in bits, not bytes. Ranges from $-1.79E + 308$ to $1.79E + 308$ .
Datetime	Date/Time	8	Date and time data from January 1, 1753, to December 31, 9999, with an accuracy of three-hundredths of a second.
SmallDateTime	Date/Time	4	Date and time data from January 1, 1900, to June 6, 2079, with an accuracy of one minute.
Money	Money	8	Monetary units from $-2^{63}$ to $2^{63}$ plus precision to four decimal places. Note that this could be any monetary unit, not just dollars.

# Data Type

Cursor	Special Numeric	1	Pointer to a cursor. While the pointer only takes up a byte, keep in mind that the result set that makes up the actual cursor also takes up memory — exactly how much will vary depending on the result set.
Timestamp/ rowversion	Special Numeric (binary)	8	Special value that is unique within a given database. Value is set by the database itself automatically every time the record is inserted or updated, even though the timestamp column wasn't referred to by the UPDATE statement. (You're actually not allowed to update the timestamp field directly.)
UniqueIdentifier	Special Numeric (binary)	16	Special Globally Unique Identifier (GUID). Is guaranteed to be unique across space and time.
char	Character	Varies	Fixed-length character data. Values shorter than the set length are padded with spaces to the set length. Data is non-Unicode. Maximum specified length is 8,000 characters.

# Data Type

Data Type Name	Class	Size in Bytes	Nature of the Data
Varchar	Character	Varies	Variable-length character data. Values are not padded with spaces. Data is non-Unicode. Maximum specified length is 8,000 characters, but you can use the <code>max</code> keyword to indicate it as essentially a very large character field (up to $2^{31}$ bytes of data).
Text	Character	Varies	Legacy support as of SQL Server 2005. Use <code>varchar(max)</code> instead.
NChar	Unicode	Varies	Fixed-length Unicode character data. Values shorter than the set length are padded with spaces. Maximum specified length is 4,000 characters.
Nvarchar	Unicode	Varies	Variable-length Unicode character data. Values aren't padded. Maximum specified length is 4,000 characters, but you can use the <code>max</code> keyword to indicate it as essentially a very large character field (up to $2^{31}$ bytes of data).

# Data Type

Ntext	Unicode	Varies	Like the Text data type, this is legacy support only. In this case, use nvarchar(max). Variable-length Unicode character data
Binary	Binary	Varies	Fixed-length binary data with a maximum length of 8,000 bytes.
VarBinary	Binary	Varies	Variable-length binary data with a maximum specified length of 8,000 bytes, but you can use the max keyword to indicate it as essentially a LOB field (up to 2^31 bytes of data).
Image	Binary	Varies	Legacy support only as of SQL Server 2005. Use varbinary(max) instead.
Table	Other	Special	This is primarily for use in working with result sets, typically passing one out of a User Defined Function. Not usable as a data type within a table definition.

# Insert Attribute

Microsoft SQL Server Management Studio

Object Explorer

Table - dbo.student\* Summary

Column Name	Data Type	Allow Nulls
student_id	nchar(10)	<input checked="" type="checkbox"/>
student_name	nchar(40)	<input checked="" type="checkbox"/>
student_lastname	nchar(50)	<input checked="" type="checkbox"/>
student_birthday	smalldatetime	<input checked="" type="checkbox"/>

Table - dbo.student\* Summary

Column Name	Data Type	Allow Nulls
student_id	nchar(10)	<input type="checkbox"/>
student_name	nchar(40)	<input checked="" type="checkbox"/>
student_lastname	nchar(50)	<input checked="" type="checkbox"/>
student_birthday	smalldatetime	<input checked="" type="checkbox"/>

Primary Key

# Open Table & Modify

Microsoft SQL Server Management Studio

Object Explorer

Table - dbo.student

student_id	student_name	student_lastname	student_birthday
222	aaaaaaaaaa	aaaaaaaaaa	... NULL
NULL	NULL	NULL	NULL

Table - dbo.student\* Summary

Column Name	Data Type	Allow Nulls
student_id	nchar(10)	<input type="checkbox"/>
student_name	nchar(40)	<input checked="" type="checkbox"/>
student_lastname	nchar(50)	<input checked="" type="checkbox"/>
student_birthday	smalldatetime	<input checked="" type="checkbox"/>

Modify

Open Table

# Exist Key

Microsoft SQL Server Management Studio

Object Explorer

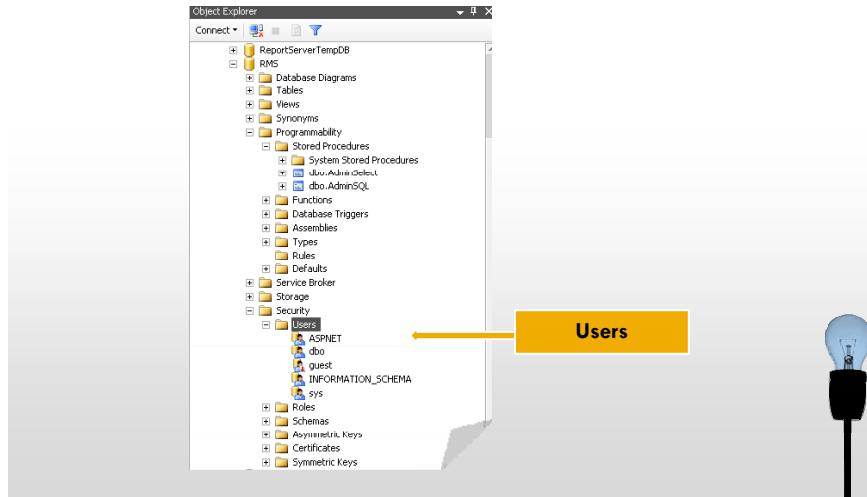
Table - dbo.student

student_id	student_name	student_lastname	student_birthday
222	Nutthapot	... Kewsawatthanapat	... 8/3/2526 0:00:00
223	Supakit	... Setakorn	... 8/3/2525 0:00:00
*	Doughnut	KK	3/3/2024
NULL	NULL	NULL	NULL

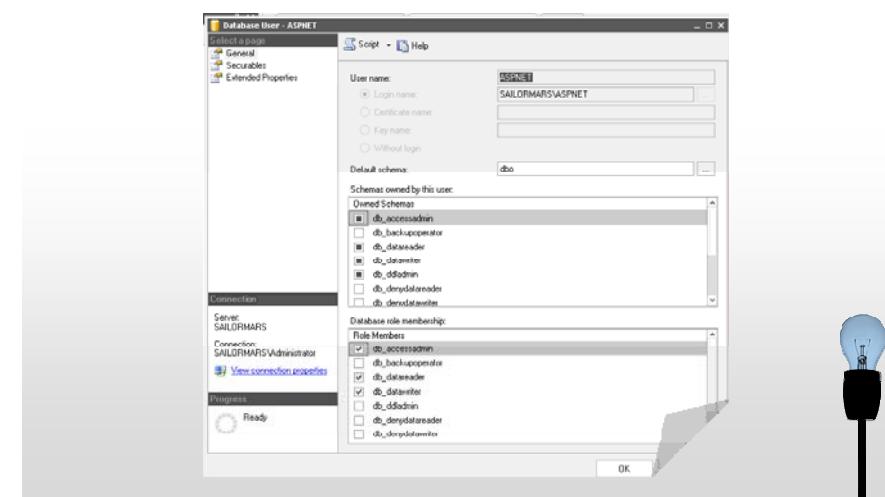
No row was updated.  
The data in row 3 was not committed.  
Error Source: .Net SqlClient Data Provider.  
Error Message: Violation of PRIMARY KEY constraint 'PK\_student'. Cannot insert duplicate key in object 'dbo.student'.  
The statement has been terminated.

OK

## Security : Authentication



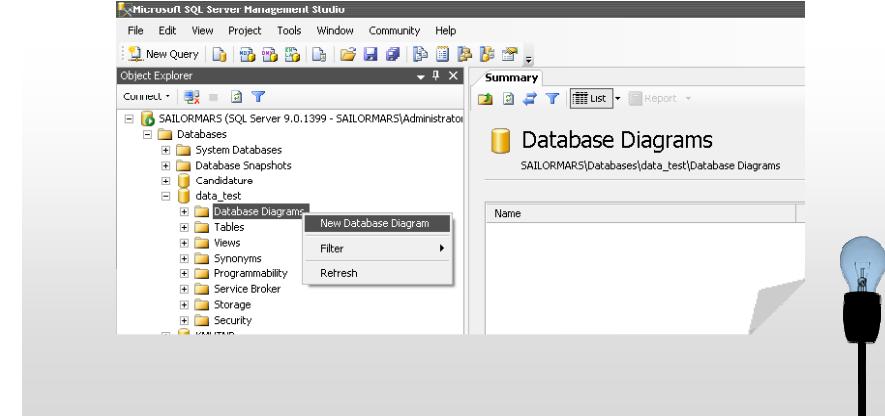
## Security : Authentication



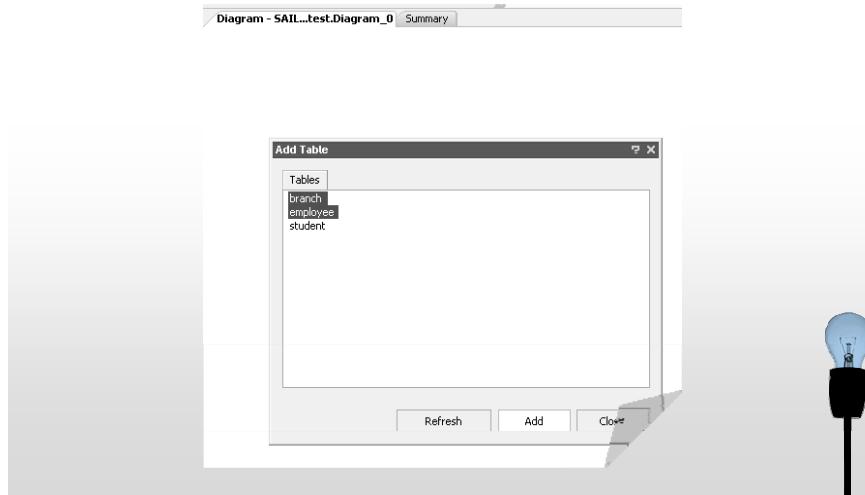
## Test : Create Database

- ให้ Normalize ข้อมูลดังนี้
  - Emp\_id,Emp\_name,Emp\_lastname,Emp\_salary,Emp\_birthday, Emp\_branch\_id,Emp\_branch\_name,Emp\_branch\_address
  - และให้สร้าง Table ตามข้อมูลข้างต้น โดยให้ระบุชนิดข้อมูลด้วย
  - ให้เพิ่มข้อมูลจริงลงแต่ละ Table

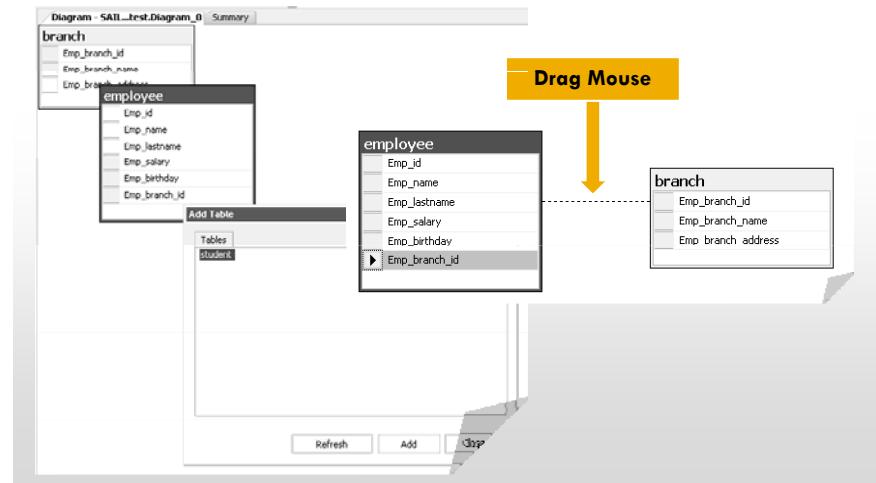
## Database Diagram



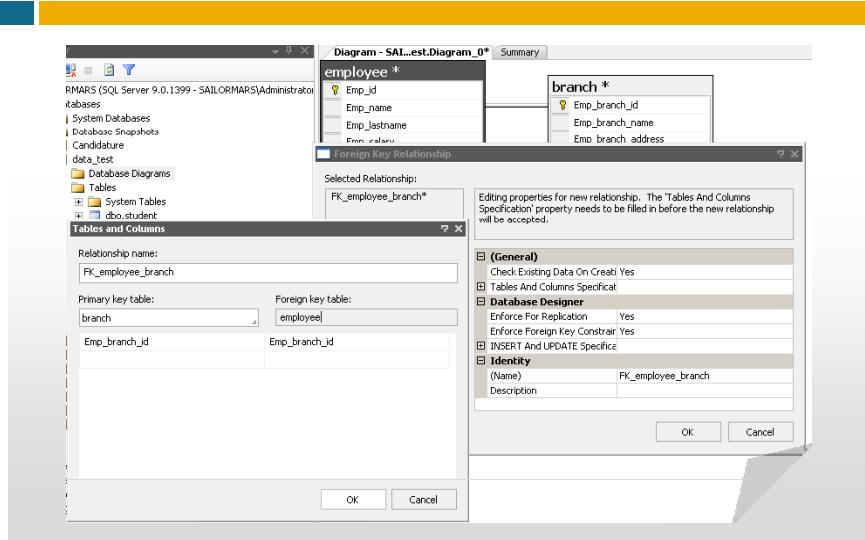
## Database Diagram : Relationship



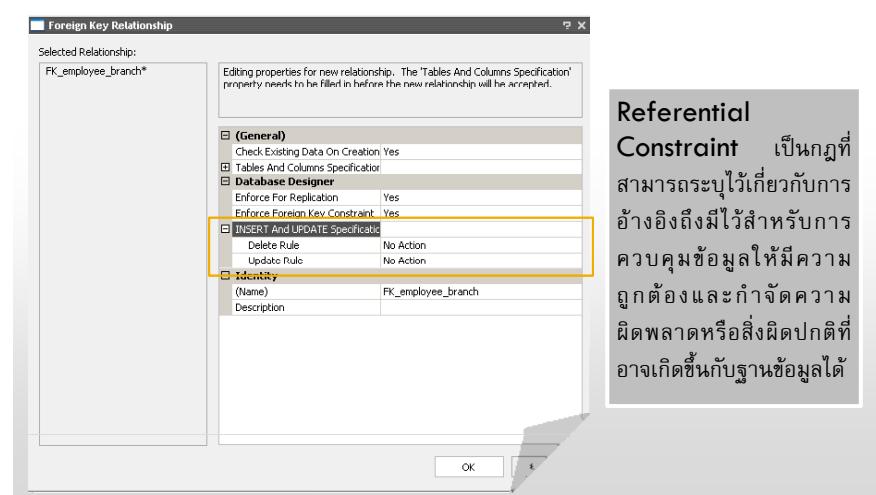
## Database Diagram : Relationship



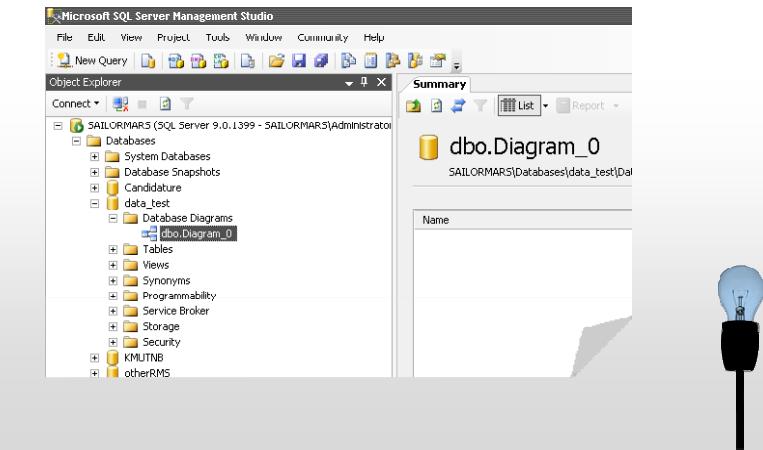
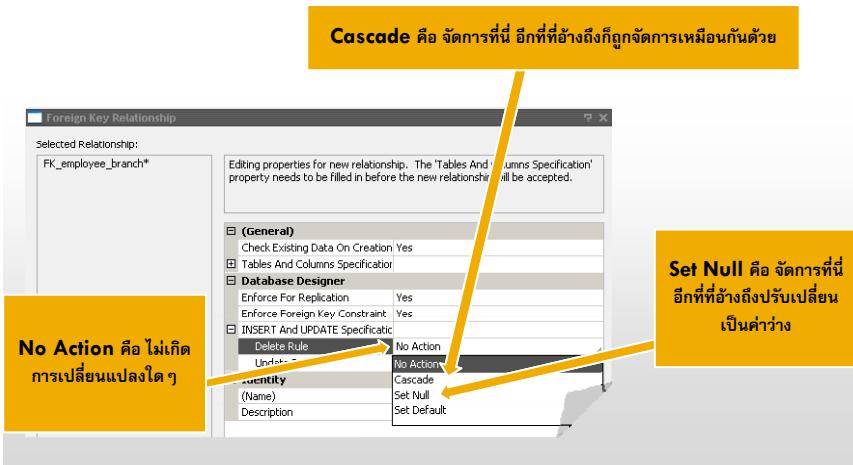
## Database Diagram : Relationship



## Referential Constraint



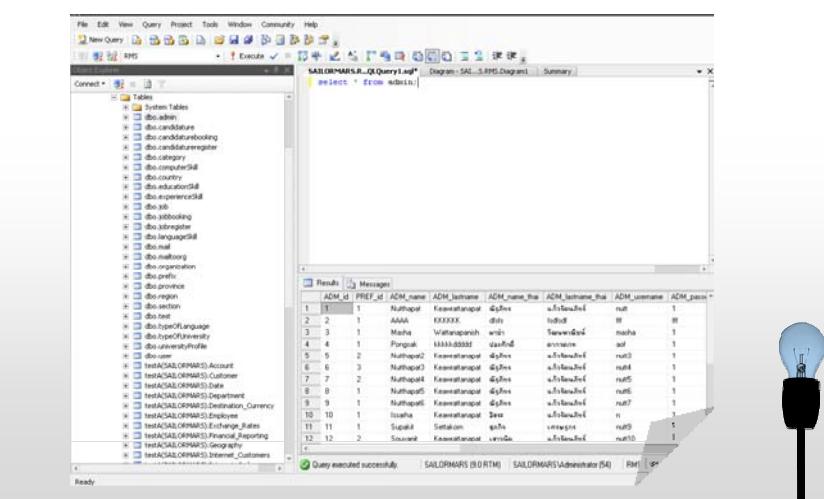
# Referential Constraint



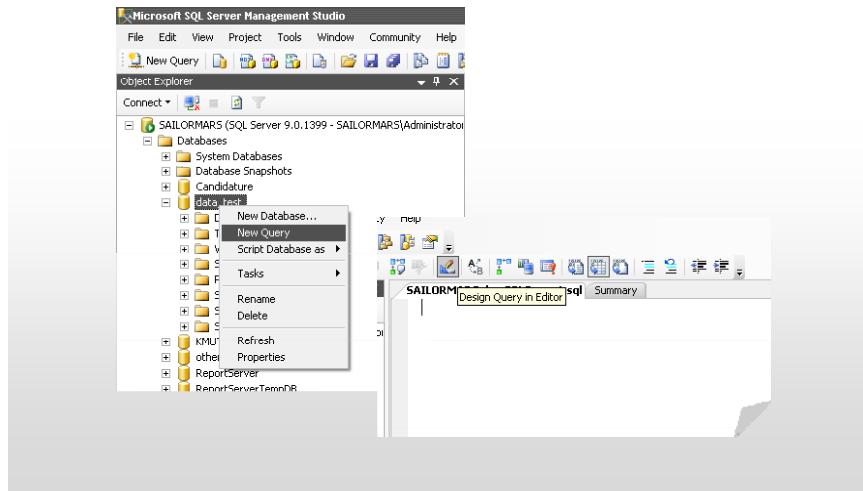
## DEMO : Referential Constraint

- ทำการแก้ไข Foreign Key และสังเกตความเปลี่ยนแปลงของการอ้างอิง
  - ทำการลบ Foreign Key และสังเกตความเปลี่ยนแปลงของการอ้างอิง
- 

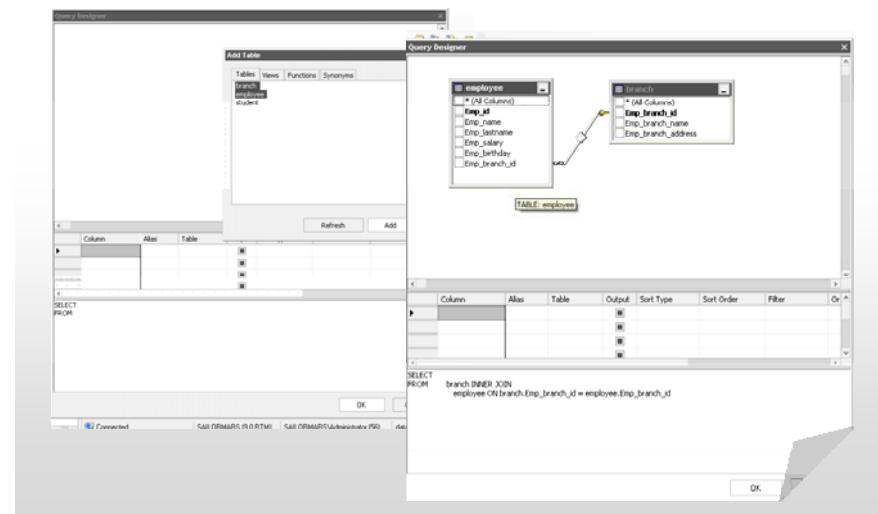
## Query Structure



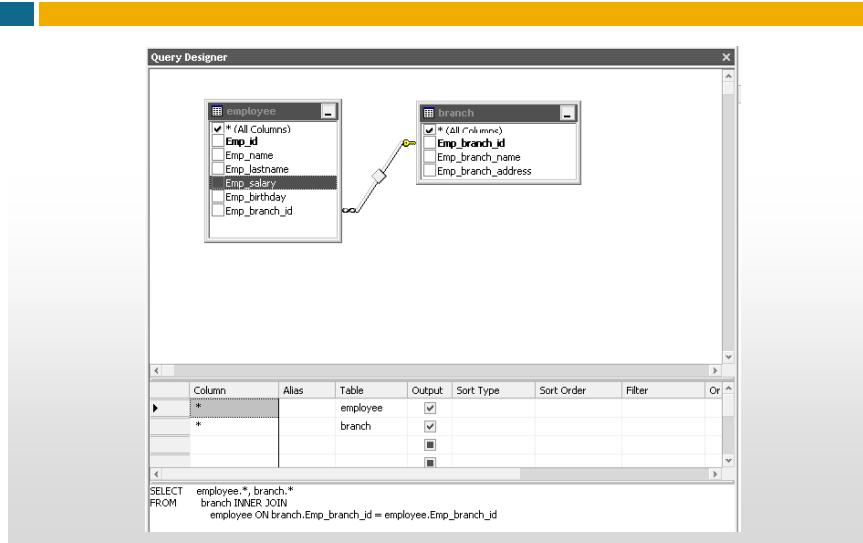
## Query Builder



## Query Builder



## Query Builder



## Query Builder

This screenshot shows the SQL Results window for the query 'SAILORMARS.dbo.SQLQuery1.sql'. The results show two rows of data from the joined 'employee' and 'branch' tables:

Emp_id	Emp_name	Emp_lastname	Emp_salary	Emp_birthday	Emp_branch_id	Emp_branch_id	Emp_branch_name
1	Nutthapat	Keawratthanapat	20000.00	1983-03-08 00:00:00	1	1	Lotus L...
2	Supakit	Settakorn	30000.00	1982-02-03 00:00:00	2	2	Lob...

## Sorting

The screenshot shows the Microsoft Query Designer interface. At the top, there are two tables: 'branch' and 'employee'. The 'branch' table has columns: Emp\_branch\_id, Emp\_branch\_name, and Emp\_branch\_address. The 'employee' table has columns: Emp\_id, Emp\_name, Emp\_lastname, Emp\_salary, Emp\_birthday, Emp\_branch\_id, and Emp\_branch\_name. In the main area, there is a grid for defining the query. The 'Sort Order' column for the 'Emp\_id' row is highlighted with a red box, showing three options: 'Ascending', 'Descending', and 'Unsorted'. Below the grid is the generated SQL query:

```
SELECT employee.Emp_id, employee.Emp_name, employee.Emp_lastname, employee.Emp_salary, employee.Emp_birthday, employee.Emp_branch_id, branch.Emp_branch_id AS Expr1, branch.Emp_branch_name, branch.Emp_branch_address
FROM branch INNER JOIN
employee ON branch.Emp_branch_id = employee.Emp_branch_id
```

## Set Order

The screenshot shows the Microsoft Query Designer interface. At the top, there are two tables: 'branch' and 'employee'. The 'branch' table has columns: Emp\_branch\_id, Emp\_branch\_name, and Emp\_branch\_address. The 'employee' table has columns: Emp\_id, Emp\_name, Emp\_lastname, Emp\_salary, Emp\_birthday, Emp\_branch\_id, and Emp\_branch\_name. In the main area, there is a grid for defining the query. The 'Sort Order' column for the 'Emp\_id' row is highlighted with a red box, showing two options: '1' and 'Unsorted'. Below the grid is the generated SQL query:

```
SELECT employee.Emp_id, employee.Emp_name, employee.Emp_lastname, employee.Emp_salary, employee.Emp_birthday, employee.Emp_branch_id, branch.Emp_branch_id AS Expr1, branch.Emp_branch_name, branch.Emp_branch_address
FROM branch INNER JOIN
employee ON branch.Emp_branch_id = employee.Emp_branch_id
```

## Filter

The screenshot shows the Microsoft Query Designer interface. At the top, there are two tables: 'branch' and 'employee'. The 'branch' table has columns: Emp\_branch\_id, Emp\_branch\_name, and Emp\_branch\_address. The 'employee' table has columns: Emp\_id, Emp\_name, Emp\_lastname, Emp\_salary, Emp\_birthday, Emp\_branch\_id, and Emp\_branch\_name. In the main area, there is a grid for defining the query. The 'Filter' column for the 'Emp\_salary' row is highlighted with a red box, containing the value '= 20000'. Below the grid is the generated SQL query:

```
SELECT employee.Emp_id, employee.Emp_name, employee.Emp_lastname, employee.Emp_salary, employee.Emp_birthday, employee.Emp_branch_id, branch.Emp_branch_id AS Expr1, branch.Emp_branch_name, branch.Emp_branch_address
FROM branch INNER JOIN
employee ON branch.Emp_branch_id = employee.Emp_branch_id
WHERE employee.Emp_salary = 20000
```

## EXE Query

The screenshot shows the Microsoft Query Designer interface. At the top, there is a single table 'branch'. The table has columns: Emp\_branch\_id, Emp\_branch\_name, Emp\_branch\_address, Emp\_id, Emp\_name, Emp\_lastname, Emp\_salary, and Emp\_birthday. In the main area, there is a grid for defining the query. The 'Filter' column for the 'Emp\_salary' row is highlighted with a yellow box, containing the value '(employee.Emp\_salary = 20000)'. Below the grid is the generated SQL query:

```
SAILORMARS.d..QLQuery1.sql* [Summary]
SELECT branch.*, employee.* , employee.Emp_salary AS Expr1
FROM branch INNER JOIN
employee ON branch.Emp_branch_id = employee.Emp_branch_id
WHERE (employee.Emp_salary = 20000)
```

Below the query is a results grid:

Emp_branch_id	Emp_branch_name	Emp_branch_address	Emp_id	Emp_name	Emp_lastname	Emp_salary	Emp_birthday
1	Lotus Charun	Charunsanitwong	1	Nuthapat	Keawrattanapat	20000.00	1983-03

OR

Query Designer

The screenshot shows the Query Designer interface with two tables selected: 'branch' and 'employee'. A join condition is set between 'branch.Emp\_branch\_id' and 'employee.Emp\_branch\_id'. The results grid displays columns from both tables, including 'Emp\_branch\_id', 'Emp\_branch\_name', 'Emp\_branch\_address', 'Employee.Emp\_id', 'Employee.Emp\_name', 'Employee.Emp\_lastname', and 'Employee.Emp\_salary'. Two filters are applied: '< 1000' and '> 2000' in the 'Filter' column. The generated T-SQL code is:

```
SELECT branch.Emp_branch_id, branch.Emp_branch_name, branch.Emp_branch_address, employee.Emp_id, employee.Emp_name, employee.Emp_lastname, employee.Emp_salary
FROM branch INNER JOIN
employee ON branch.Emp_branch_id = employee.Emp_branch_id
WHERE (employee.Emp_salary < 1000) OR
(employee.Emp_salary > 2000)
```

## Generate Script

The screenshot shows the Object Explorer in Microsoft SQL Server Management Studio. A context menu is open over a database named 'data\_1'. The 'Tasks' submenu is expanded, and the 'Generate Scripts...' option is highlighted with a yellow box.

Generate Script

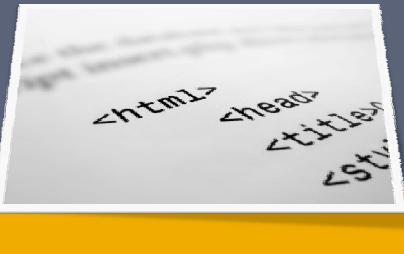
The screenshot shows a Windows desktop with a Notepad window titled 'aaaa.sql - Notepad' open. The script contains T-SQL code to create the 'branch' and 'employee' tables with their respective columns and constraints. A red arrow points to the file icon on the desktop, indicating the saved file 'aaaa.sql'.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[branch]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[branch]
(
    [Emp_branch_id] [int] NOT NULL,
    [Emp_branch_name] [nchar](50) NULL,
    [Emp_branch_address] [nchar](500) NULL,
    CONSTRAINT [PK_branch] PRIMARY KEY CLUSTERED
    (
        [Emp_branch_id] ASC
    ) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[employee]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[employee]
(
    [Emp_id] [int] NOT NULL,
    [Emp_name] [nchar](50) NULL,
    [Emp_lastname] [nchar](50) NULL,
    [Emp_salary] [money] NULL,
    [Emp_birthday] [smalldatetime] NULL,
    [Emp_branch_id] [int] NULL,
    CONSTRAINT [PK_employee] PRIMARY KEY CLUSTERED
```

# ASP.NET WITH C#

## CLASS 5

อ.ณัฐภรณ์ แก้วรัตนภรณ์



## Agenda

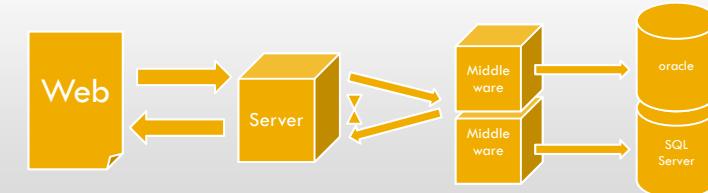
- สร้างฐานข้อมูลเพื่อใช้สำหรับการเขียนโปรแกรมติดต่อฐานข้อมูล
- ทดลองใช้ Server Control ที่เกี่ยวกับฐานข้อมูล
- การเพิ่ม ลบ และแก้ไขข้อมูล
- การค้นหาข้อมูล
- การเขียนโปรแกรมแบบ Connection Code

## Create Database

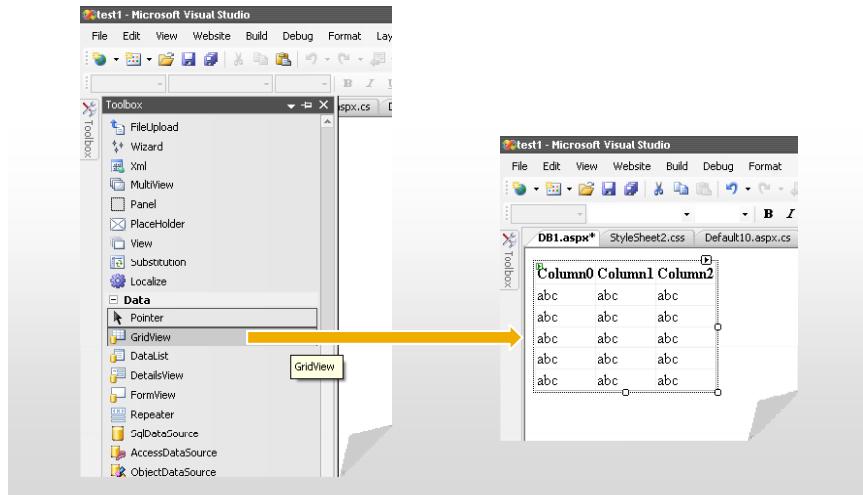
- สร้างฐานข้อมูลดังนี้
  - ▣ Book(book\_id,book\_name,book\_cost,book\_status)
  - ▣ Rent(rent\_id,customer\_id,book\_id,rent\_date,rent\_return\_date)
  - ▣ Customer(customer\_id,customer\_name,customer\_address,customer\_tel)
  - ▣ เมื่อสร้างเสร็จแล้วให้ทำการเชื่อมความสัมพันธ์ (Relational)
  - ▣ จากนั้นเพิ่มข้อมูลให้เหมาะสม

## Database Connection

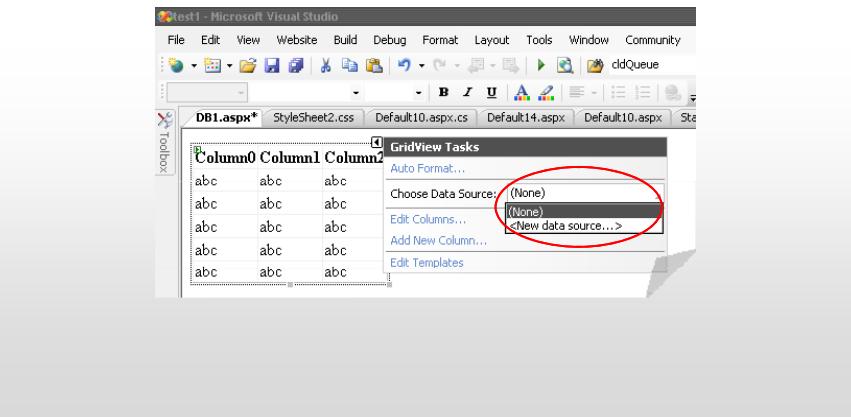
- การเชื่อมโยงข้อมูลภายในฐานข้อมูลกับแอ็พพลิเคชันมีหลากหลายวิธี เช่น
  - ▣ การเชื่อมโยงด้วยการใช้ Server Control ผูกโดยตรงกับฐานข้อมูล
  - ▣ การเชื่อมโยงด้วยการเขียน Code และผูก Database Object กับ Server Control



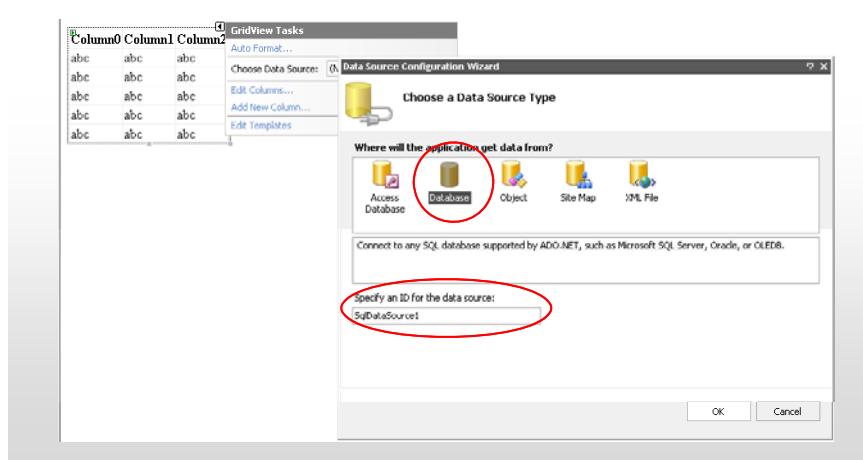
## Grid View Server Control



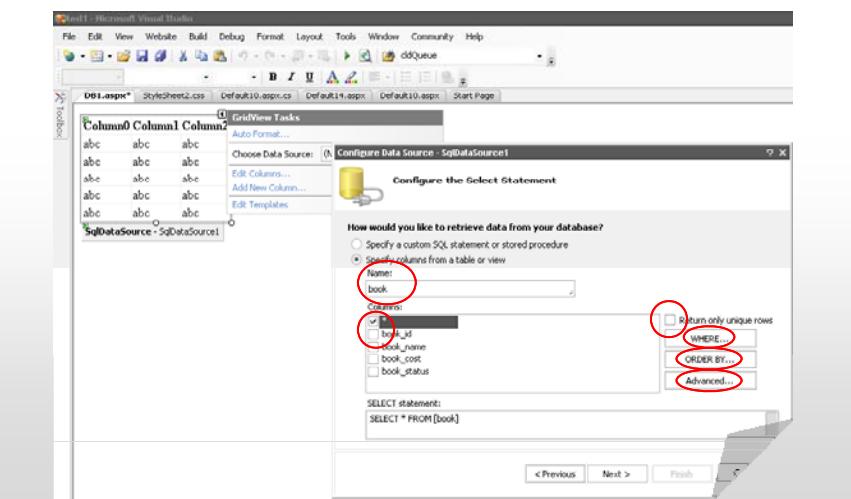
## Grid View Server Control



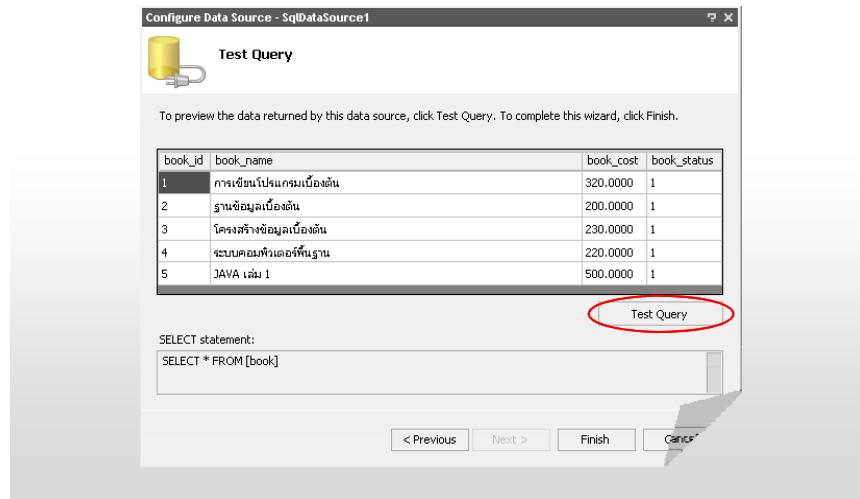
## Grid View : New Connection



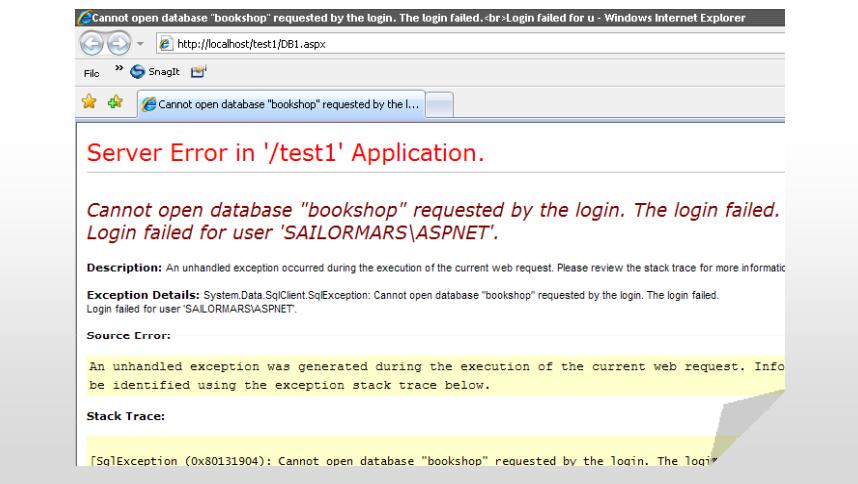
## Grid View : Configure the Select Statement



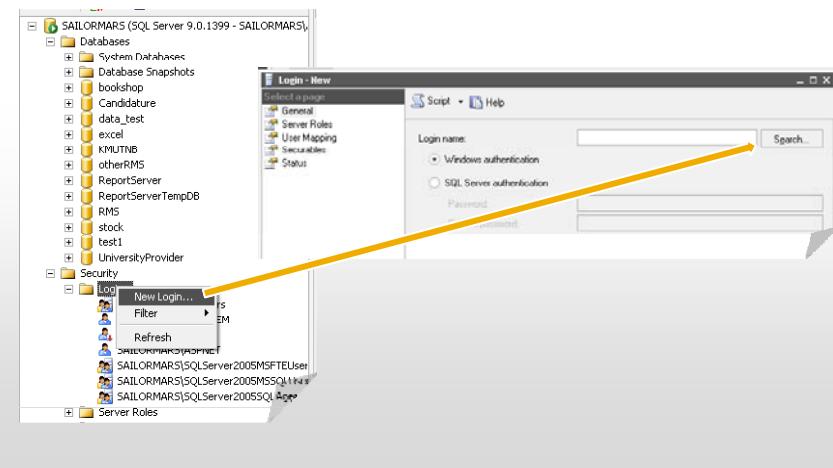
## Grid View : Test Query



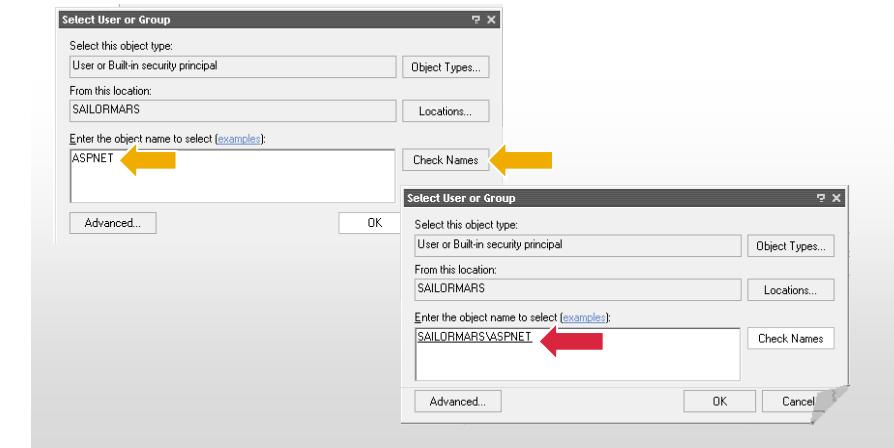
## Error : Database Permission



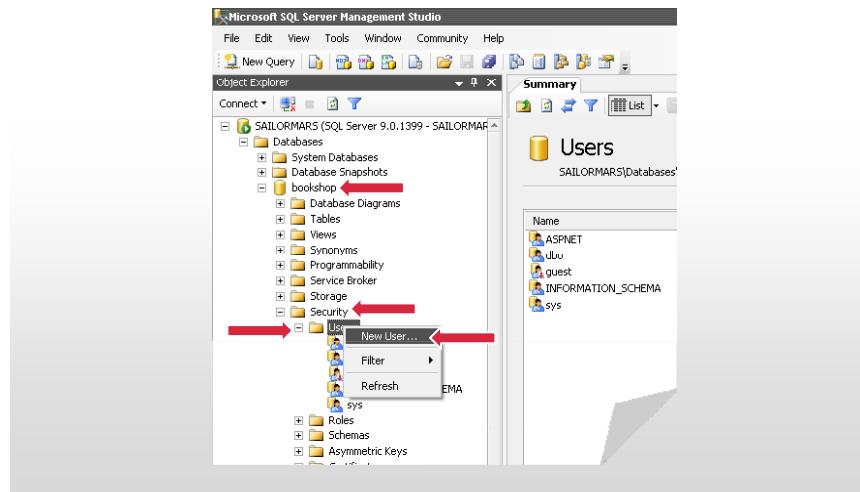
## Security : Database Login



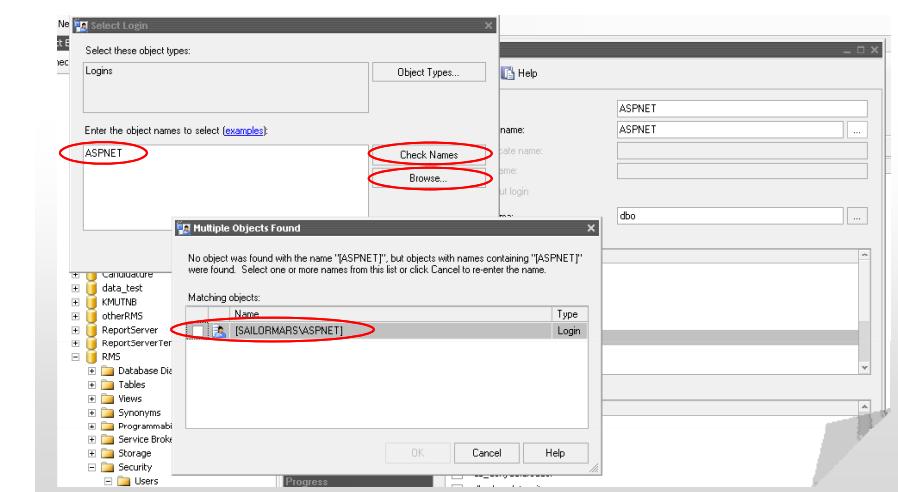
## Security : Database Login



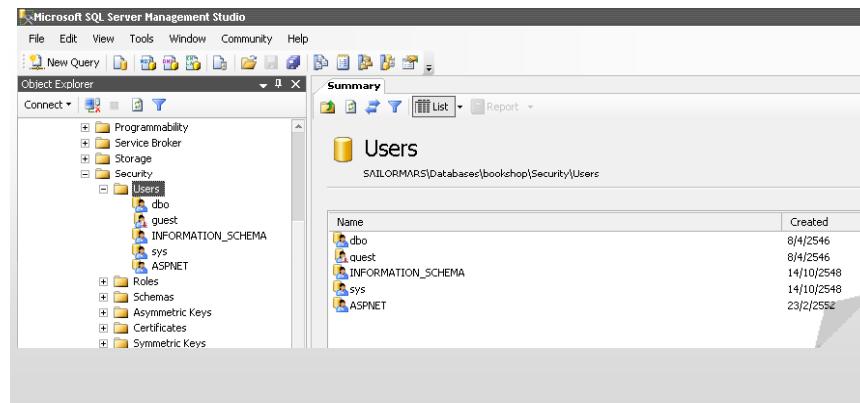
## Security : Security->Users



## Security : Security->Users



## Security : Security->Users



## Grid View

book_id	book_name	book_cost	book_status
1	การเขียนโปรแกรมเบื้องต้น	320.0000	1
2	ฐานข้อมูลเบื้องต้น	200.0000	1
3	โครงสร้างข้อมูลเบื้องต้น	230.0000	1
4	ระบบคอมพิวเตอร์พื้นฐาน	220.0000	1
5	JAVA เล่ม 1	500.0000	1

## Grid View Customization

Screenshot of Microsoft Visual Studio showing a GridView control in design mode. The grid displays data from a SQLDataSource. A yellow arrow points to the 'Properties' window, which shows the 'GridView1' properties. The 'HeaderStyle' section is expanded, showing 'BackColor' set to '#8080FF'.

## Grid View Customization

Screenshot of Microsoft Visual Studio showing a GridView control in design mode. The grid displays data from a SQLDataSource. A yellow arrow points to the 'Properties' window, which shows the 'GridView1' properties. The 'HeaderStyle' section is expanded, showing 'BackColor' set to '#8080FF'. Another yellow arrow points to the 'Font' section, where 'ForeColor' is set to 'Yellow'.

## Enable Paging

Screenshot of Microsoft Visual Studio showing a GridView control in design mode. The grid displays data from a SQLDataSource. A yellow arrow points to the 'GridView Tasks' context menu, which includes an option 'Enable Paging' with a checked checkbox. The 'book\_status' column header is circled in red.

## Insert Image into Grid View

Screenshot of Microsoft SQL Server Management Studio showing the 'Table - dbo.book' table. The 'book\_picture' column contains image file paths like 'photo/assem.jpg'. A yellow arrow points to the 'book\_picture' column header.

## Refresh Attribute in Grid View

A screenshot of Microsoft Visual Studio showing a grid view. The grid has columns: book\_id, book\_name, book\_cost, and book\_status. The data is as follows:

book_id	book_name	book_cost	book_status
0	abc	0	0
1	abc	0.1	1
2	abc	0.2	2
12			

The cell containing '12' is highlighted with a blue border. Below the grid, there is a SqlDataSource control labeled 'SqlDataSource1'.

## Insert Image

A screenshot of Microsoft Visual Studio showing a grid view. The grid has columns: book\_id, book\_name, book\_cost, and book\_status. The data is as follows:

book_id	book_name	book_cost	book_status
0	abc	0	0
1	abc	0.1	1
2	abc	0.2	2
12			

The cell containing '12' is highlighted with a blue border. Below the grid, there is a SqlDataSource control labeled 'SqlDataSource1'. A small image icon is visible in the bottom right corner of the screen.

## Insert Image

- {0} ແທນ Attribute ຂີ່ອ book\_picture ທີ່ຈະກື້ອຂໍ້ມູນ photo/sql2000.jpg
- ~/ ແທນ Path ຈະງໍາທີ່ໄປ Server

A screenshot of Microsoft Visual Studio showing the 'Fields' dialog for a grid view. The 'Selected fields' list contains book\_id, book\_name, book\_cost, book\_status, and book\_picture. In the 'ImageField properties:' section, the 'DataImageUrlField' is set to 'book\_picture' and 'DataImageUrlFormat' is set to '{0}'. The 'Data' section shows the field 'book\_picture' circled with a red oval. Below the dialog is a preview of the grid view showing three rows of data with images.

## Edit Field Name

A screenshot of Microsoft Visual Studio showing a grid view. The grid has columns: book\_id, book\_name, book\_cost, book\_status, and book\_status. The data is as follows:

book_id	book_name	book_cost	book_status	book_status
0	abc	0	0	Databound
1	abc	0.1	1	Databound
2	abc	0.2	2	Databound
12				

The cell containing '12' is highlighted with a blue border. Below the grid, there is a SqlDataSource control labeled 'SqlDataSource1'.

## Programming in Grid View

- **Grid View** เป็นเครื่องมือที่มีประสิทธิภาพมากในการแสดงผลรูปแบบตาราง โดยสามารถปรับปรุงและเข้าถึงเครื่องมือได้อย่างสะดวกและรวดเร็ว ซึ่งเราสามารถทำการโปรแกรมโดยตรงกับ **Grid View** เช่น

- การเปลี่ยนชื่อความบน **Header** ใช้ `GridView1.Columns[1].HeaderText = "Name";`
- การเลือกไม่แสดงบางคอลัมน์ใช้ `GridView1.Columns[0].Visible = false;`
- หรือ การวนและปรับเปลี่ยนค่า (**RowDataBound**) ก็สามารถทำได้จ่าย

## Event : RowDataBound (วนผูกข้อมูล)

The screenshot shows the Visual Studio IDE with multiple tabs open. The current tab is 'DB1.aspx.cs'. In the center, there is a 'GridView1' control displaying a table with columns: ID, Book Name, Price, Status, and Picture. The rows show data: ID 0, Book Name abc, Price 0, Status 0, Picture Databound; ID 1, Book Name abc, Price 0.1, Status 1, Picture Databound; ID 2, Book Name abc, Price 0.2, Status 2, Picture Databound. Below the grid, the 'Properties' window is open, showing the 'GridView1' type as 'System.Web.UI.WebControls.GridView'. Under the 'Events' section, the 'RowDataBound' event is selected, highlighted with a yellow arrow. A tooltip for 'RowDataBound' is displayed: 'Fires after a row has been databound.'

- เป็น Event ที่เกิดขึ้นทุกครั้ง ที่เกิดการวนแต่ละแถวและแต่ละคอลัมน์ใน **GridView**

## Event : RowDataBound (วนผูกข้อมูล)

- การทดลองเขียนข้อมูลลงคอลัมน์
- สร้าง **Template Field** เพื่อให้มีช่องกรอกข้อมูลว่าง

The screenshot shows the Visual Studio IDE with multiple tabs open. The current tab is 'DB1.aspx.cs'. In the center, there is a 'GridView1' control displaying a table with columns: ID, Book Name, Price, Status, and Picture. The rows show data: ID 0, Book Name abc, Price 0, Status 0, Picture Databound; ID 1, Book Name abc, Price 0.1, Status 1, Picture Databound; ID 2, Book Name abc, Price 0.2, Status 2, Picture Databound. Below the grid, the 'Properties' window is open, showing the 'GridView1' type as 'System.Web.UI.WebControls.GridView'. Under the 'Events' section, the 'RowDataBound' event is selected, highlighted with a yellow arrow. A tooltip for 'RowDataBound' is displayed: 'Fires after a row has been databound.'

## Event : RowDataBound (วนผูกข้อมูล)

- การโปรแกรมเมื่อ มีการวนผูกข้อมูล

```
protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if(e.Row.RowType == DataControlRowType.DataRow)
    {
        string temp = Convert.ToString(DataBinder.Eval(e.Row.DataItem, "book_cost"));
        e.Row.Cells[5].Text = temp.ToString();
    }
}
```

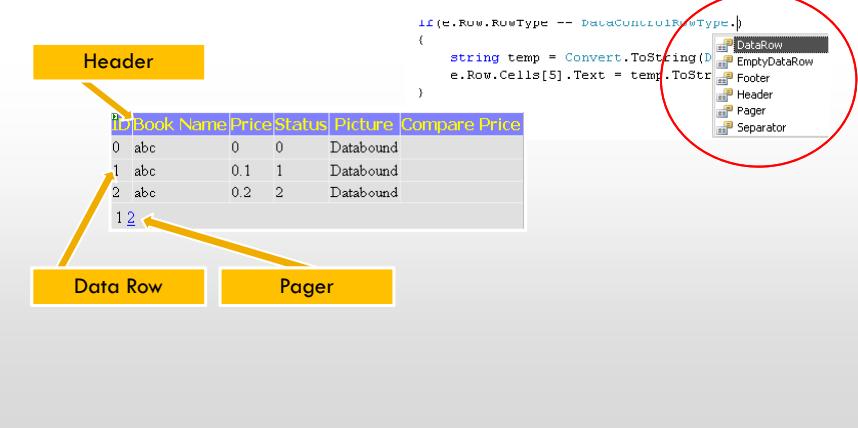
ID	Book Name	Price	Status	Picture	Compare Price
0	abc	0	0	Databound	
1	abc	0.1	1	Databound	
2	abc	0.2	2	Databound	

## DataBinder.Eval

- เป็นการดึงค่าจากการผูกข้อมูล โดยข้อมูลมาจากการฐานข้อมูล ซึ่งไม่ใช้การเรียกมาจากข้อมูลที่ผูกแล้ว

```
DataBinder.Eval(e.Row.DataItem, "book_cost")
```

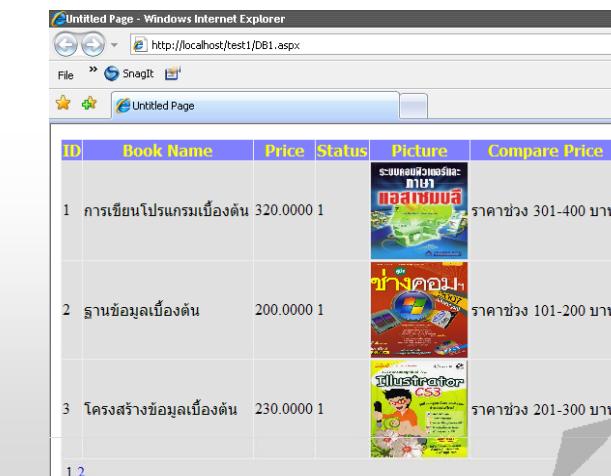
## Event : RowDataBound (วนผูกข้อมูล)



## Example Code

```
protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if(e.Row.RowType == DataControlRowType.DataRow)
    {
        double temp = Convert.ToDouble(DataBinder.Eval(e.Row.DataItem, "book_cost"));
        if (temp <= 100)
        {
            e.Row.Cells[5].Text = "ราคาช่วง 1-100 บาท";
        }
    }
}
```

## Event : RowDataBound (วนผูกข้อมูล)



## Enable Sorting

Screenshot of Microsoft Visual Studio showing a GridView control. The column headers are highlighted with a red circle. A red arrow points from the 'Enable Sorting' checkbox in the 'GridView Tasks' context menu to the 'Enable Sorting' checkbox in the 'Edit Columns...' dialog.

## Test : Use Data Set from MS-Access

- ใน ASP.NET มีตัวกลางในการเชื่อมต่อฐานข้อมูลชนิดต่างๆ มากมาย เช่น MS-Access , Oracle , MySQL และ Informix เป็นต้น

Screenshot of Microsoft Visual Studio showing the Solution Explorer. A yellow arrow points from the 'New Folder' option in the context menu to the 'data' folder in the Solution Explorer.

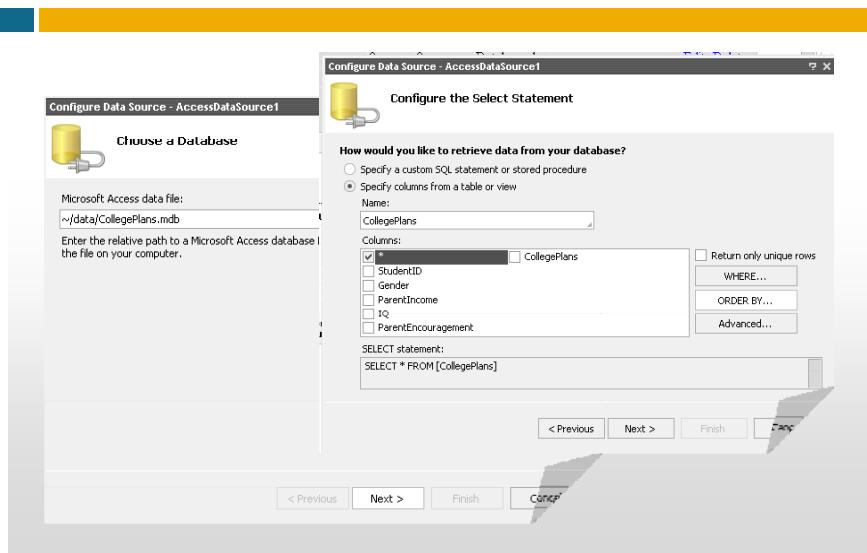
## Test : Use Data Set from MS-Access

Screenshot of Microsoft Visual Studio showing two instances of the 'GridView Tasks' context menu. The top one has 'Choose Data Source: (None)' selected. The bottom one has 'SqlDataSource1' selected under 'Choose Data Source:'. A red circle highlights the 'Choose Data Source:' dropdown in both menus.

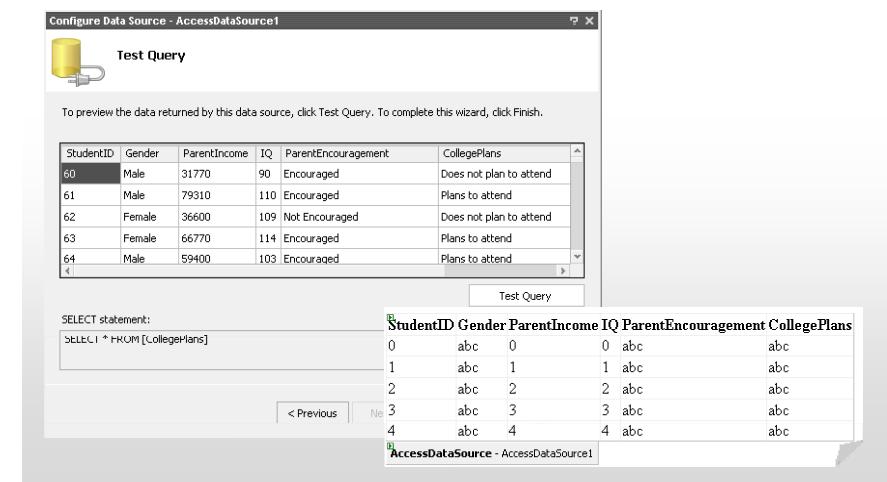
## Test : Use Data Set from MS-Access

Screenshot of the 'Select Microsoft Access Database' dialog. It shows the 'Project folders' tree with 'CollegePlans.mdb' selected in the 'data' folder. The 'Access Database' icon is highlighted with a red circle in the 'Where will the application get data from?' section. The 'Microsoft Access Databases (\*.mdb)' dropdown also has a red circle around it.

## Test : Use Data Set from MS-Access

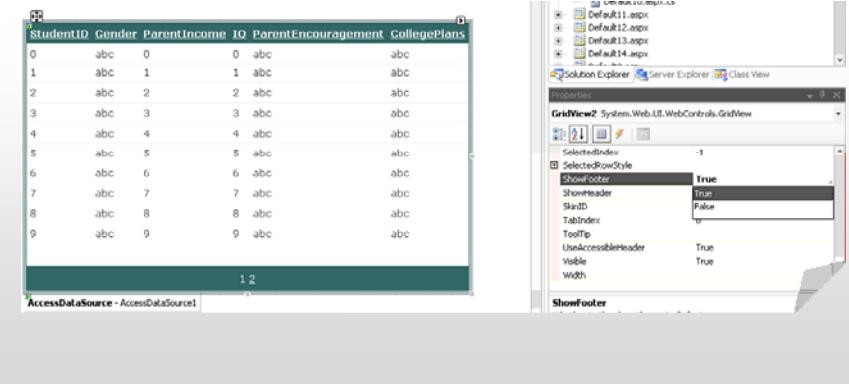


## Test : Use Data Set from MS-Access



## Test : Use Data Set from MS-Access

### □ วิธีการทำ Footer ใน GridView



## Test : Use Data Set from MS-Access

```
protected void GridView2_RowDataBound(object sender,
    GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.Footer)
    {
        e.Row.Cells[2].Text = temp.ToString();
    }
    else if(e.Row.RowType == DataControlRowType.DataRow)
    {
        temp += Convert.ToInt32(e.Row.Cells[2].Text.ToString());
    }
}
```

## Test : Use Data Set from MS-Access

`e.Row.RowType == DataControlRowType.DataRow`

StudentID	Gender	ParentIncome	IQ	ParentEncouragement	CollegePlans
60	Male	31770	90	Encouraged	Does not plan to attend
61	Male	79310	110	Encouraged	Plans to attend
62	Female	36600	109	Not Encouraged	Does not plan to attend
63	Female	66770	114	Encouraged	Plans to attend
64	Male	59400	103	Encouraged	Plans to attend
65	Female	61700	100	Not Encouraged	Does not plan to attend
66	Female	60720	99	Encouraged	Plans to attend
67	Female	33570	104	Encouraged	Does not plan to attend
68	Male	63360	102	Encouraged	Plans to attend
69	Female	68530	92	Encouraged	Plans to attend
561730					

`e.Row.RowType == DataControlRowType.Footer`

1 2 3 4 5 6 7 8 9 10 ...

## Test

- ให้ทำสรุปยอดจำนวนเพศชาย และเพศหญิง , สรุป Parent Income จำแนกตามเพศ

StudentID	Gender	ParentIncome	IQ	ParentEncouragement	CollegePlans
60	Male	31770	90	Encouraged	Does not plan to attend
61	Male	79310	110	Encouraged	Plans to attend
62	Female	36600	109	Not Encouraged	Does not plan to attend
63	Female	66770	114	Encouraged	Plans to attend
64	Male	59400	103	Encouraged	Plans to attend
65	Female	61700	100	Not Encouraged	Does not plan to attend
66	Female	60720	99	Encouraged	Plans to attend
67	Female	33570	104	Encouraged	Does not plan to attend
68	Male	63360	102	Encouraged	Plans to attend
69	Female	68530	92	Encouraged	Plans to attend
561730					

1 2 3 4 5 6 7 8 9 10 ...

## Count Page

StudentID	Gender	ParentIncome	IQ	ParentEncouragement	CollegePlans
0	abc	0	0	abc	abc
1	abc	1	1	abc	abc
2	abc	2	2	abc	abc
3	abc	3	3	abc	abc
4	abc	4	4	abc	abc
5	abc	5	5	abc	abc
6	abc	6	6	abc	abc
7	abc	7	7	abc	abc
8	abc	0	0	abc	abc
9	abc	9	9	abc	abc
561820					

1 2

AccessDataSource - AccessDataSource1

## Count Page

StudentID	Gender	ParentIncome	IQ	ParentEncouragement	CollegePlans
60	Male	31770	90	Encouraged	Does not plan to attend
61	Male	79310	110	Encouraged	Plans to attend
62	Female	36600	109	Not Encouraged	Does not plan to attend
63	Female	66770	114	Encouraged	Plans to attend
64	Male	59400	103	Encouraged	Plans to attend
65	Female	61700	100	Not Encouraged	Does not plan to attend
66	Female	60720	99	Encouraged	Plans to attend
67	Female	33570	104	Encouraged	Does not plan to attend
68	Male	63360	102	Encouraged	Plans to attend
69	Female	68530	92	Encouraged	Plans to attend
70	Female	11600	75	Not Encouraged	Does not plan to attend
71	Female	60170	109	Encouraged	Plans to attend
72	Female	35550	73	Encouraged	Does not plan to attend
73	Male	61820	107	Encouraged	Plans to attend
74	Female	36610	64	Encouraged	Does not plan to attend
75	Female	55330	109	Encouraged	Plans to attend
76	Female	38200	100	Not Encouraged	Does not plan to attend
77	Female	43010	99	Encouraged	Plans to attend
78	Female	16900	101	Not Encouraged	Does not plan to attend
79	Female	20800	65	Not Encouraged	Does not plan to attend
561730					

1 2 3 4 5 6 7 8 9 10 ...

## Count Page

```
protected void GridView2_DataBound(object sender, EventArgs e)
{
    Label1.Text = Convert.ToInt32(GridView2.PageIndex + 1).ToString();
    Label1.Text += "/" + GridView2.PageCount.ToString();
}
```

## Highlight for interest row

StudentID	Gender	ParentIncome	IQ	ParentEncouragement	CollegePlans
60	Male	31770	90	Encouraged	Does not plan to attend
61	Male	79310	110	Encouraged	Plans to attend
62	Female	36600	109	Not Encouraged	Does not plan to attend
63	Female	66770	114	Encouraged	Plans to attend
64	Male	59400	103	Encouraged	Plans to attend
65	Female	61700	100	Not Encouraged	Does not plan to attend
66	Female	60720	99	Encouraged	Plans to attend
67	Female	33570	104	Encouraged	Does not plan to attend
68	Male	63360	102	Encouraged	Plans to attend
69	Female	68530	92	Encouraged	Plans to attend
70	Female	11600	75	Not Encouraged	Does not plan to attend
71	Female	60170	109	Encouraged	Plans to attend
72	Female	35550	73	Encouraged	Does not plan to attend
73	Male	61820	107	Encouraged	Plans to attend
74	Female	38610	64	Encouraged	Does not plan to attend
75	Female	55330	109	Encouraged	Plans to attend
76	Female	38200	100	Not Encouraged	Does not plan to attend
77	Female	43010	99	Encouraged	Plans to attend
78	Female	16900	108	Not Encouraged	Does not plan to attend
79	Female	20800	65	Not Encouraged	Does not plan to attend
			943720		

## Highlight for interest row

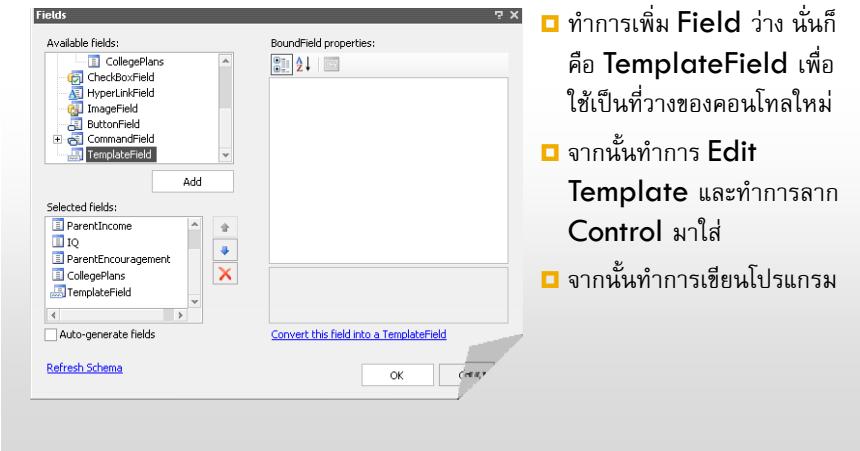
```
protected void GridView2_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.Footer)
    {
        e.Row.Cells[2].Text = temp.ToString();
    }
    else if(e.Row.RowType == DataControlRowType.DataRow)
    {
        temp += Convert.ToInt32(e.Row.Cells[2].Text.ToString());
        if (Convert.ToInt32(e.Row.Cells[3].Text.ToString()) > 100)
        {
            e.Row.BackColor = Color.Pink;
        }
    }
}
```

## Add CheckBox to Template Field

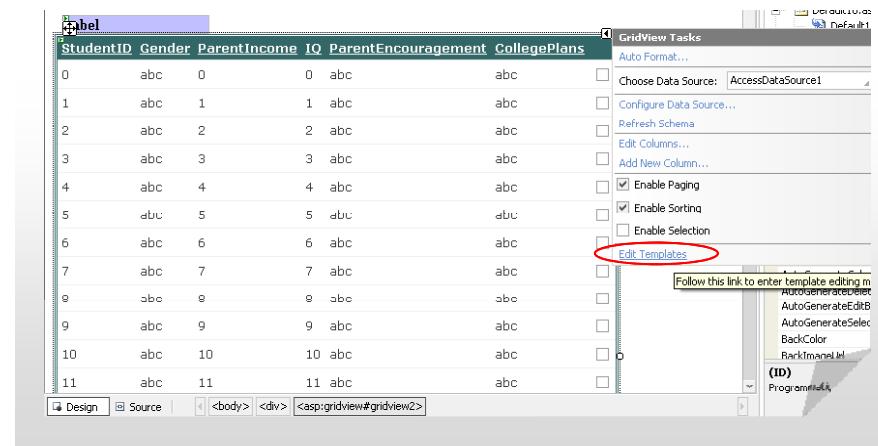
- ในการพัฒนาระบบจำเป็นต้องมีการเลือกข้อมูลในตารางเพื่อนำมาใช้งาน ต่างๆ เช่น เลือกนำมารายงาน หรือเลือกมาลบ เป็นต้น โดยสามารถเพิ่ม คอลล็อกที่ใช้ประจำ เช่น **CheckBox** เข้ามาในคอลล็อก **Gridview** ได้โดยง่าย

## Add CheckBox to Template Field

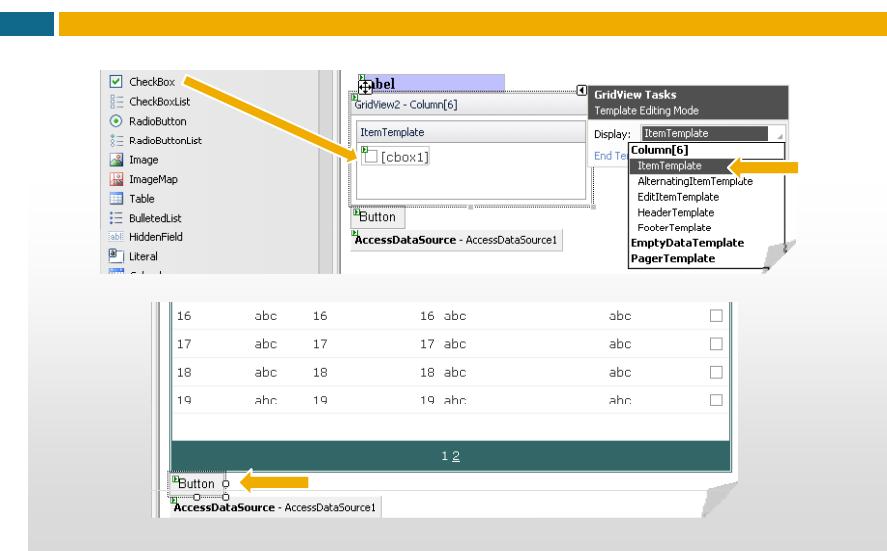
### วิธีการ



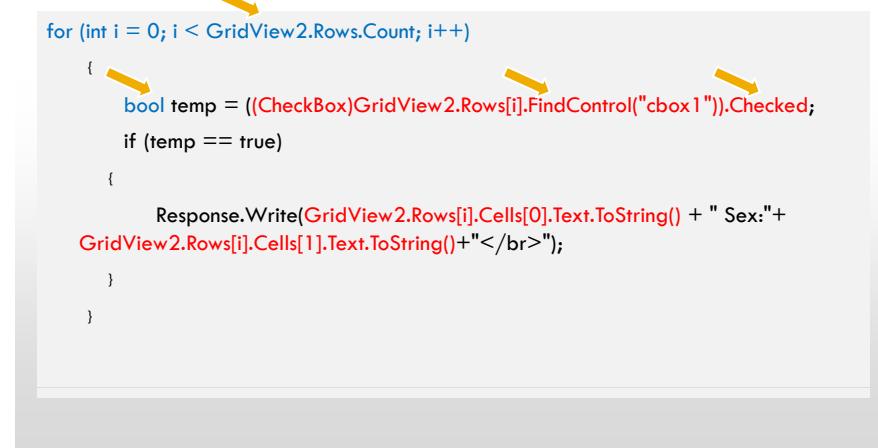
## Add CheckBox to Template Field



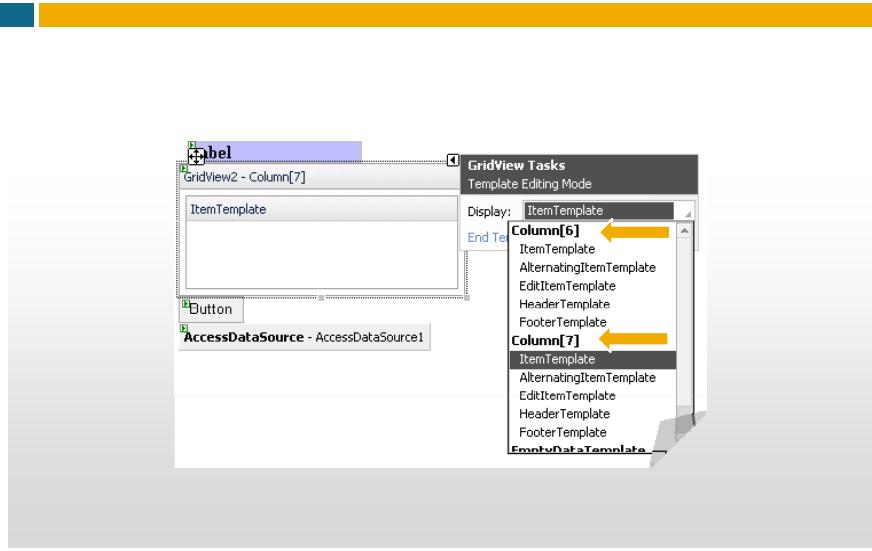
## Add CheckBox to Template Field



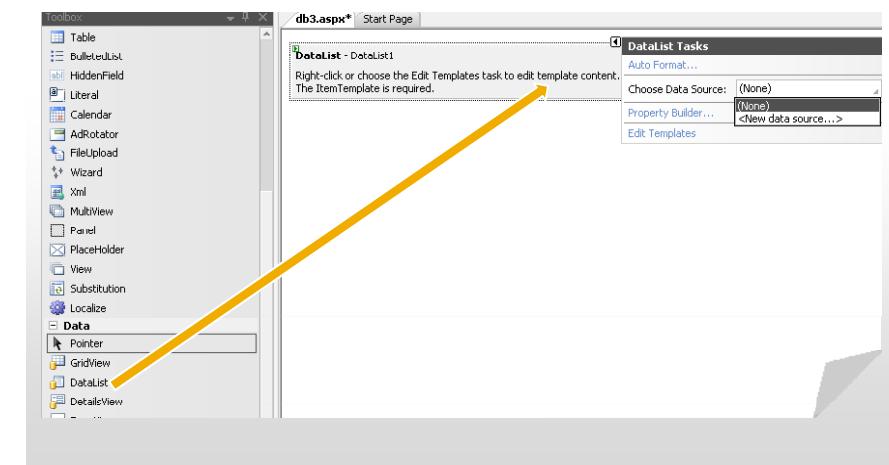
## Add CheckBox to Template Field



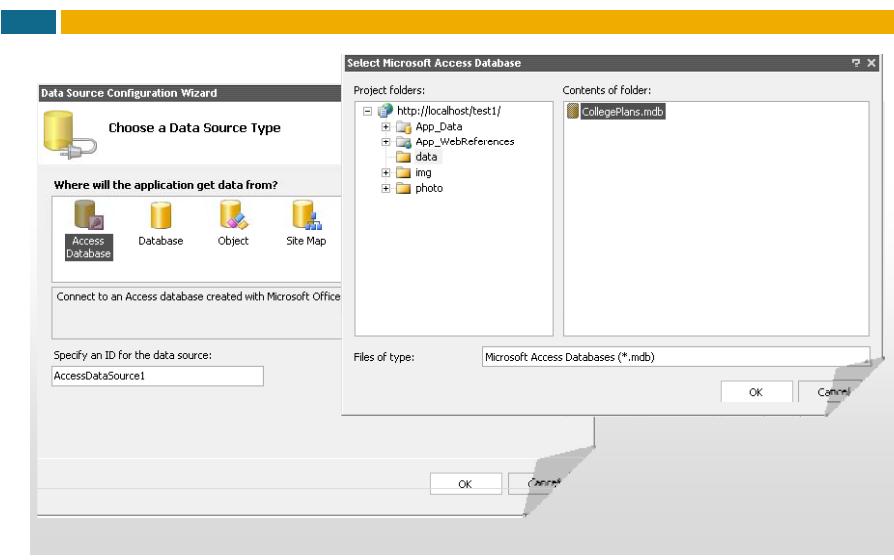
## Add Controls to Template Field



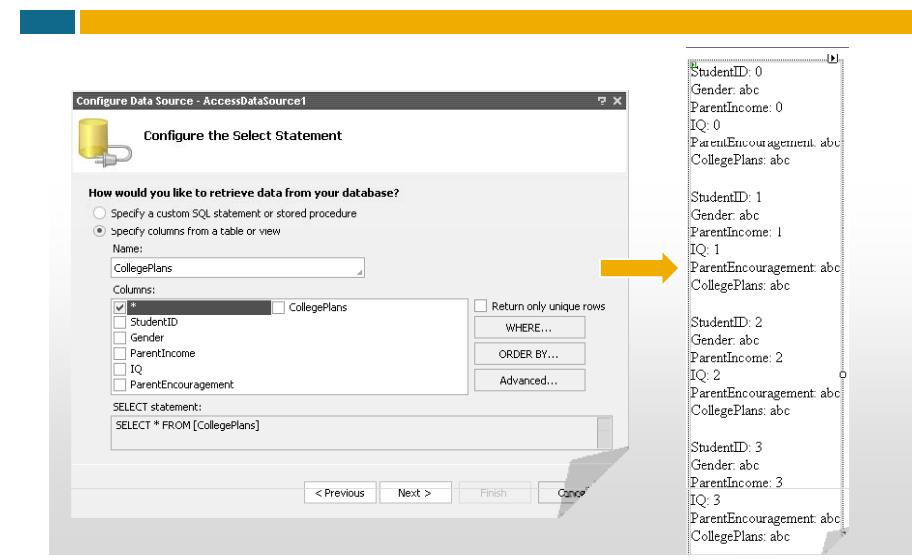
## Data List Control



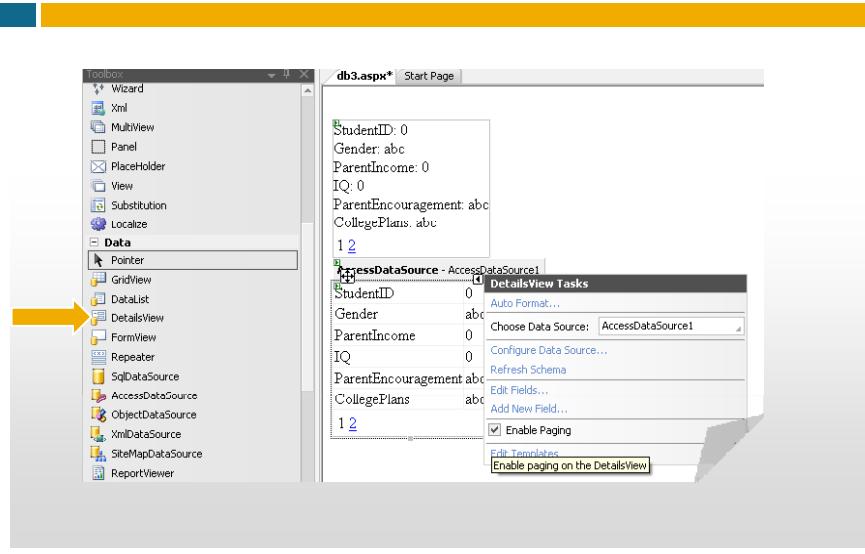
## Data List Control



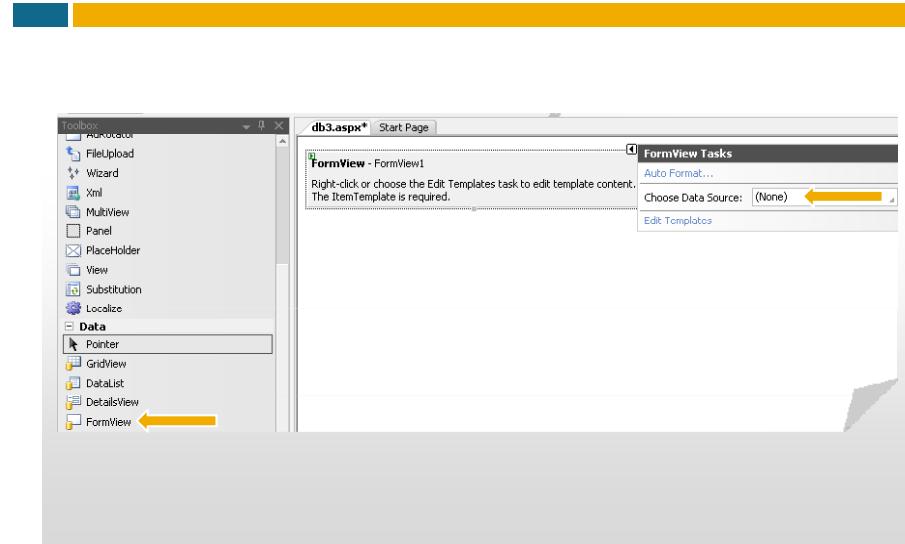
## Data List Control



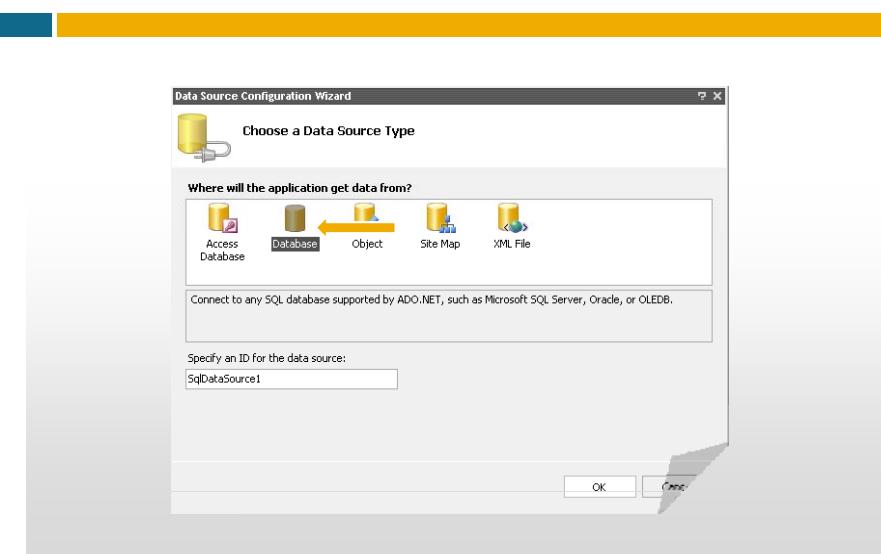
## DetailsView



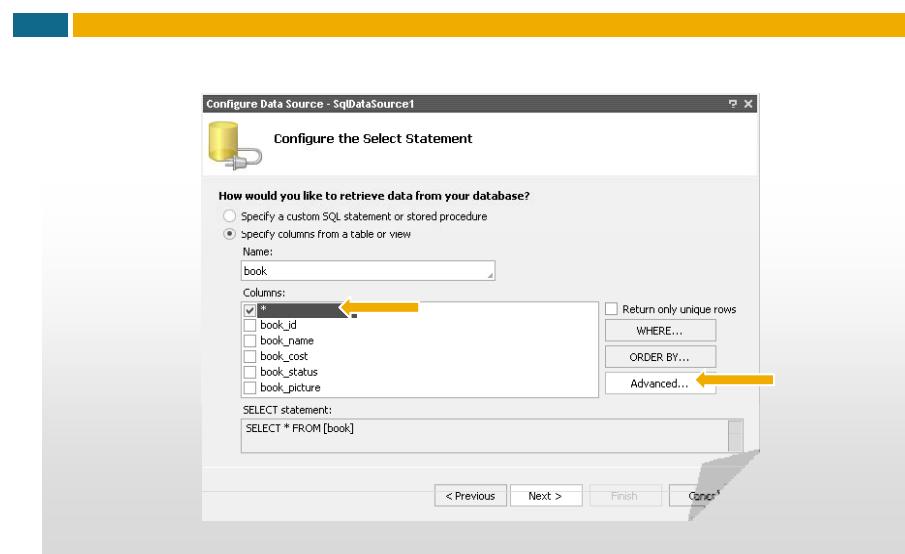
## FormView



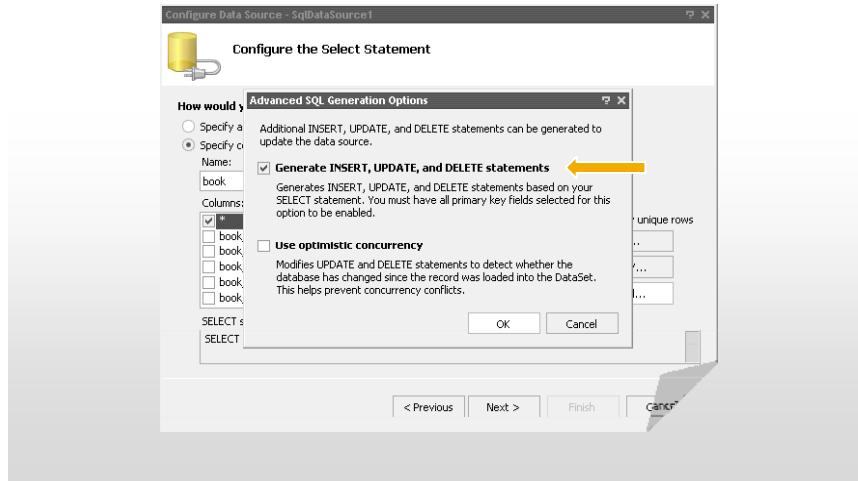
## FormView



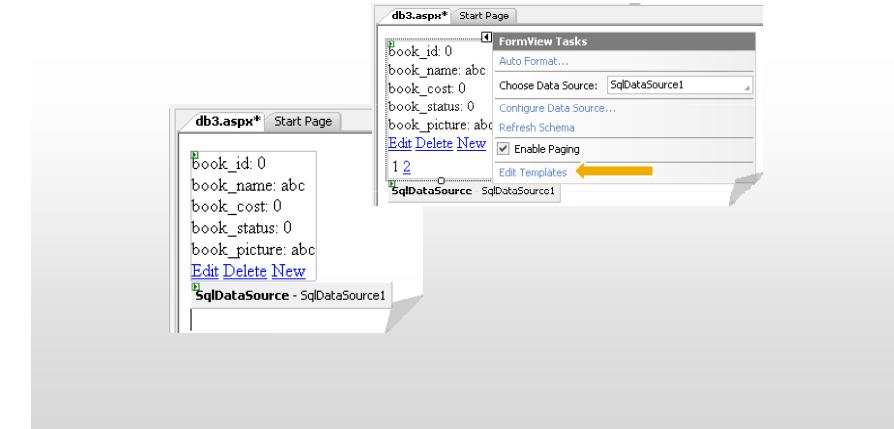
## FormView



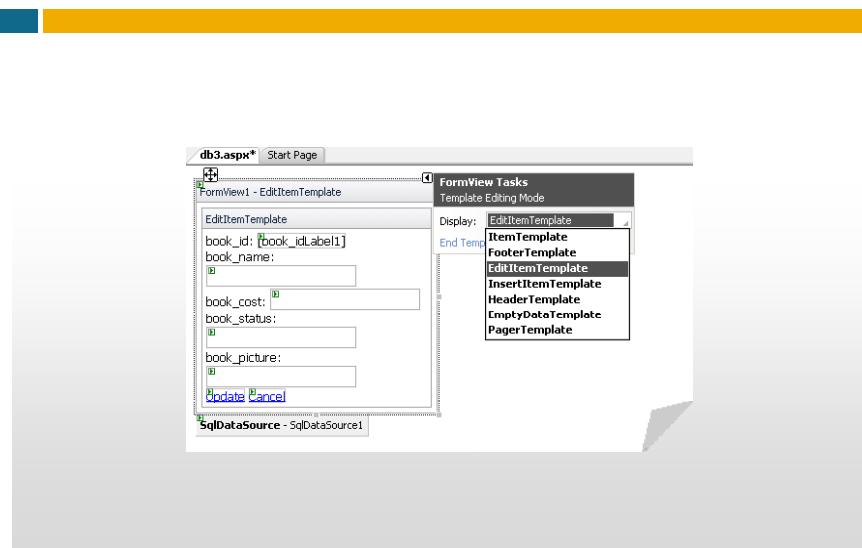
## FormView



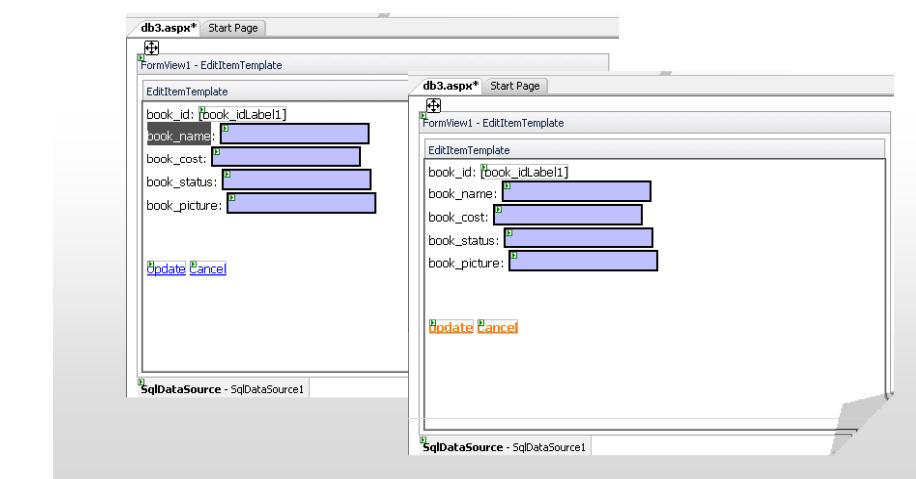
## FormView



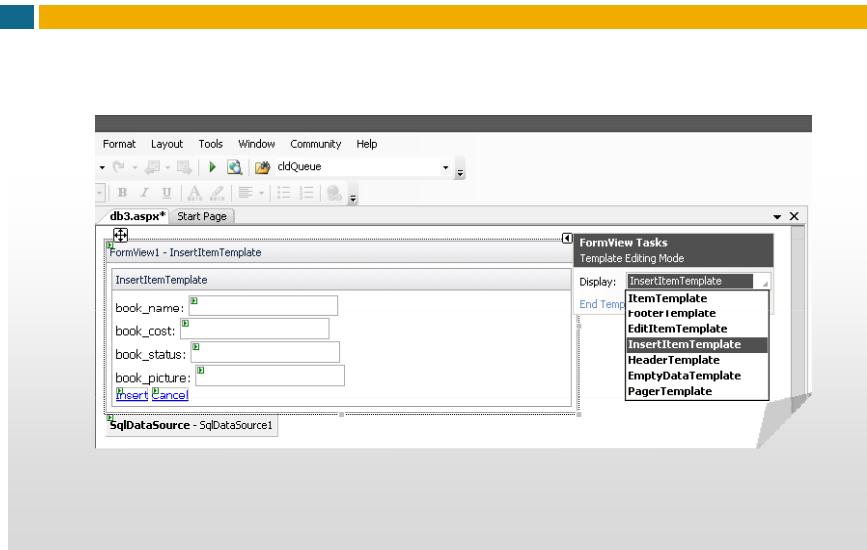
## FormView



## FormView



## FormView



## FormView

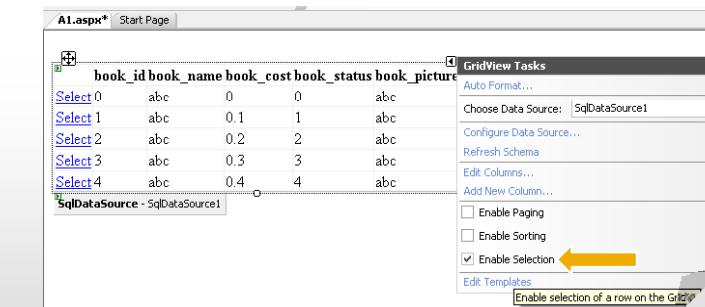


## Variable Transference

- การส่งค่าตัวแปรเพื่อมาใช้ในการเลือกชุดข้อมูลขึ้นมาแก้ไขหรือลบสามารถทำได้ง่ายๆ ดังตัวอย่างต่อไปนี้
  - ▣ วิธีการแรก คือ วิธีการส่งตัวแปรข้ามหน้าเว็บเพื่อไปค้นหาข้อมูลขึ้นมาจัดการ โดยการใช้ **QueryString**
  - ▣ วิธีการที่สอง คือ การเข้าถึงตัวแปรจาก **Control** โดยตรง
  - ▣ วิธีการที่สาม คือ การเขียนโปรแกรมล้วนๆ

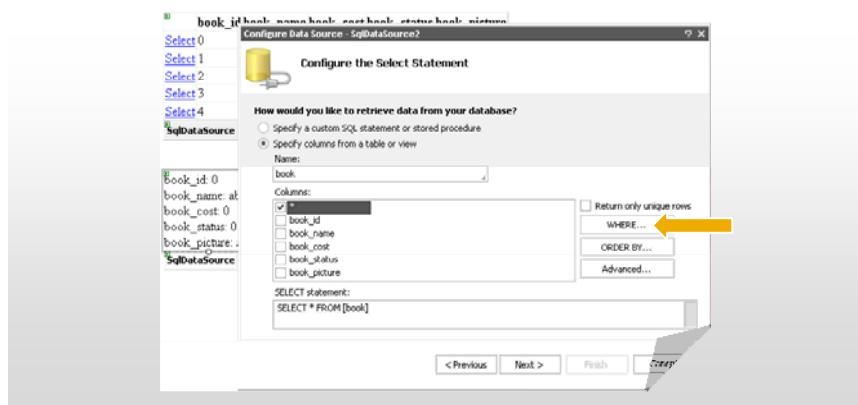
## Variable Transference

- สร้าง GridView และเลือก Enable Selection

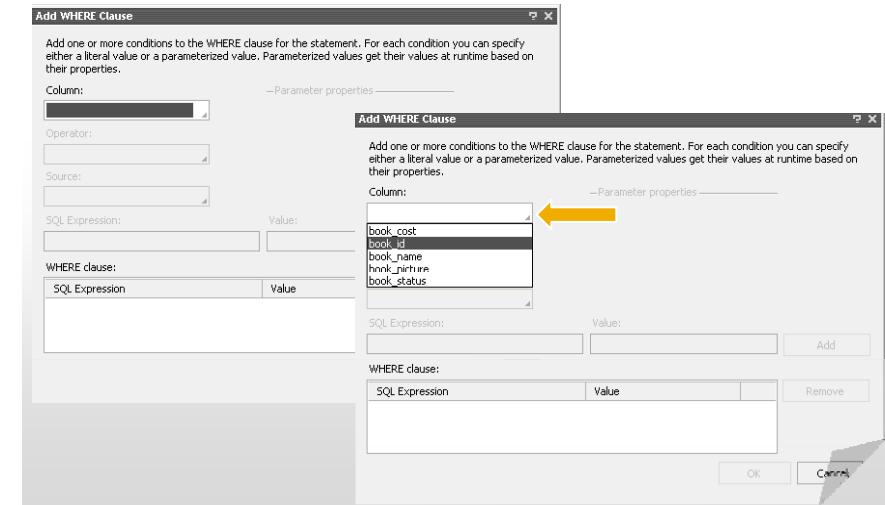


## Variable Transference

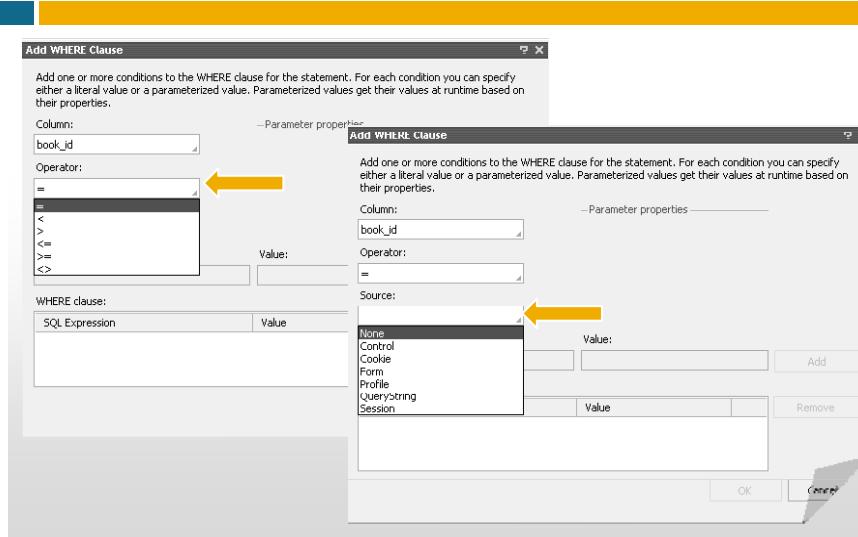
- สร้าง FormView โดยทำการสร้าง Connection ใหม่ และทำการเลือก Where เพื่อใส่เงื่อนไขการคิวเรียกข้อมูล



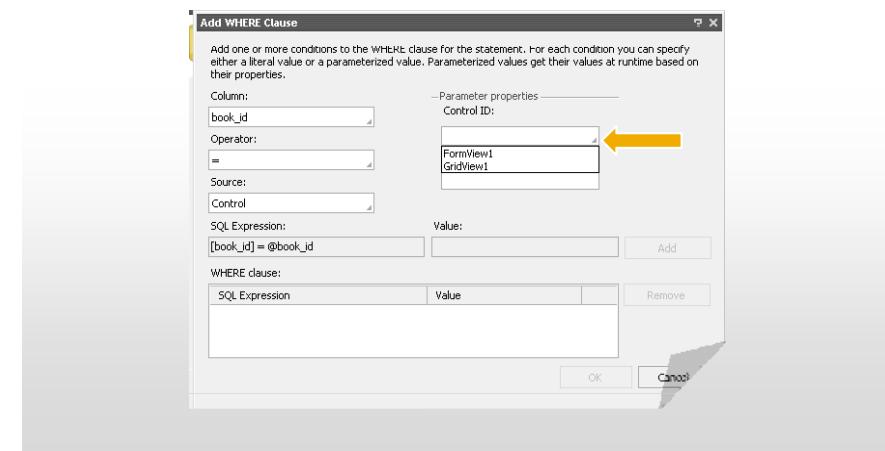
## Variable Transference



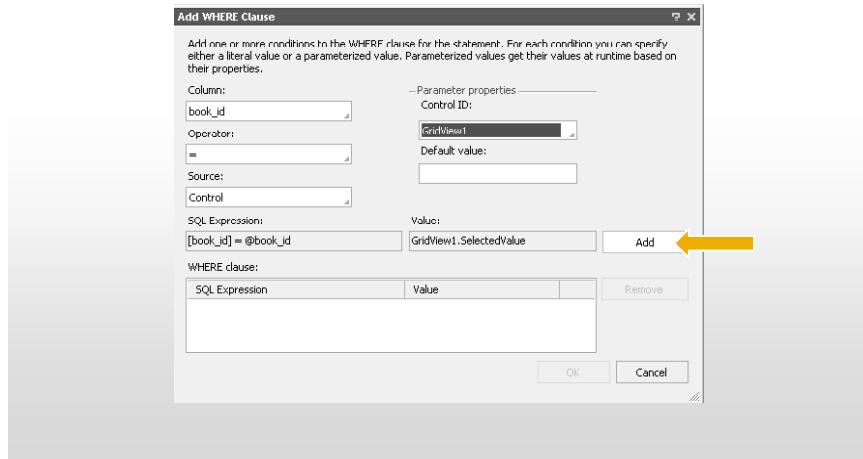
## Variable Transference



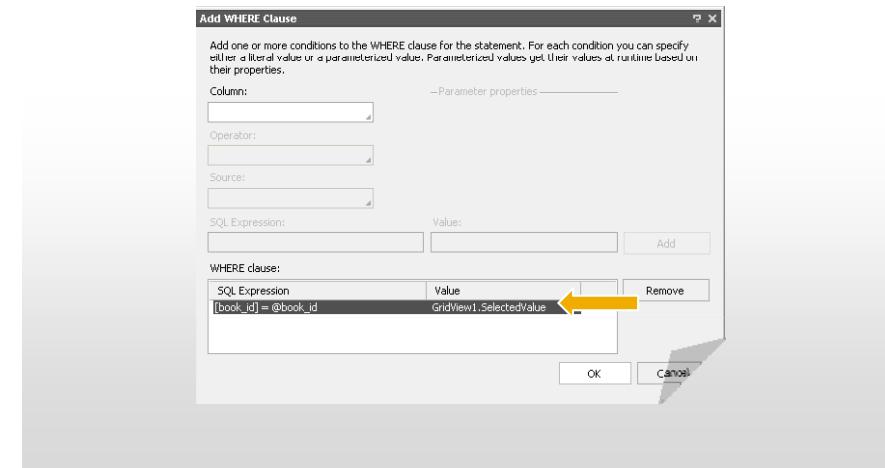
## Variable Transference



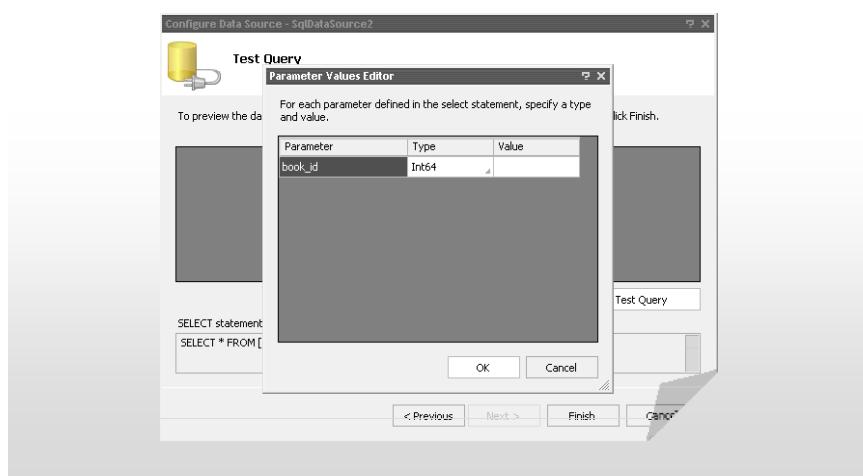
## Variable Transference



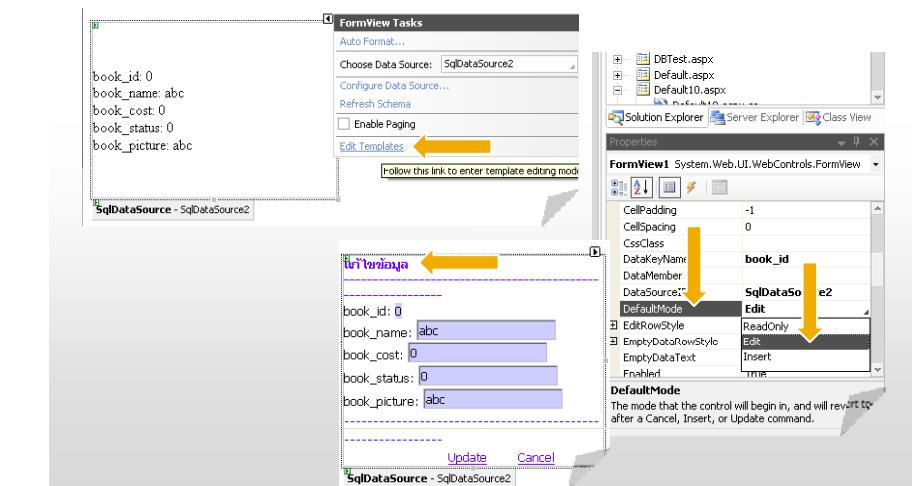
## Variable Transference



## Variable Transference



## Variable Transference

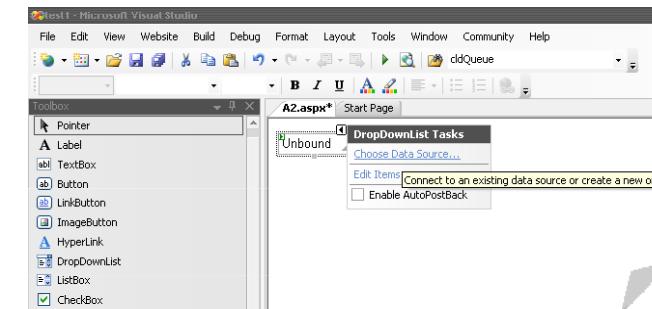


## Variable Transference

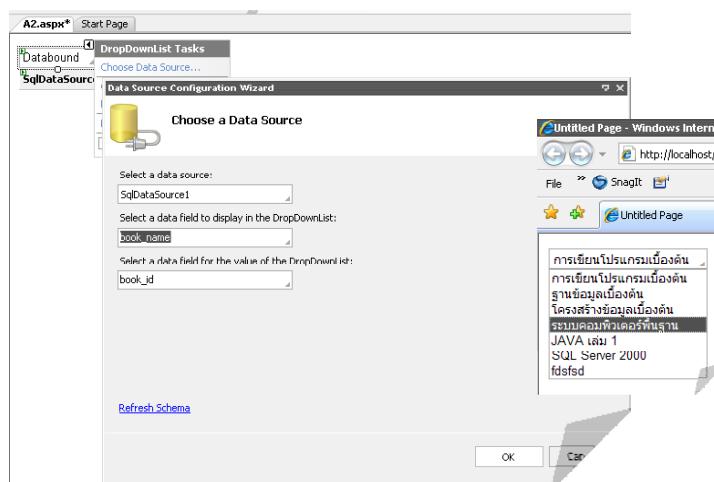
The screenshot shows a web application interface. At the top, there is a table with columns: book\_id, book\_name, book\_cost, book\_status, and book\_picture. The table contains 7 rows labeled Select 1 to Select 7. Below the table, a message "แก้ไขข้อมูล" (Edit) is displayed. A modal dialog box is open, showing the details for book ID 5: book\_name: JAVA เล่ม 1, book\_cost: 500.0000, book\_status: 1, and book\_picture: photo/os.jpg. There are "Update" and "Cancel" buttons at the bottom of the modal.

## Variable Transference

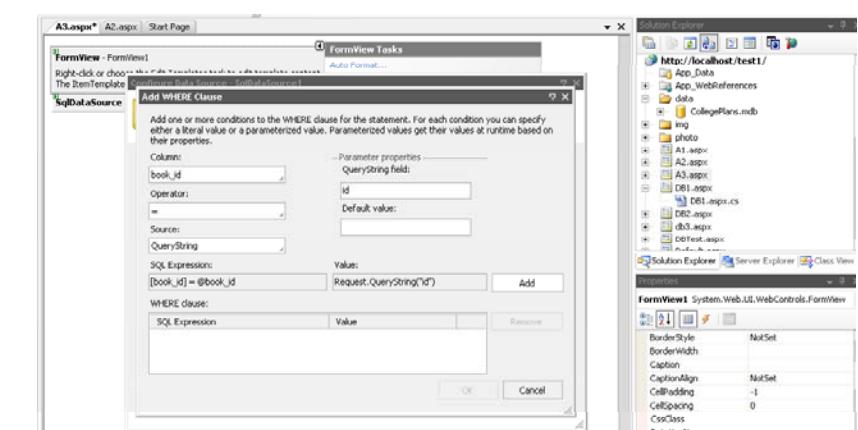
- ตัวอย่างการส่งตัวแปรแบบข้ามเว็บเพจ



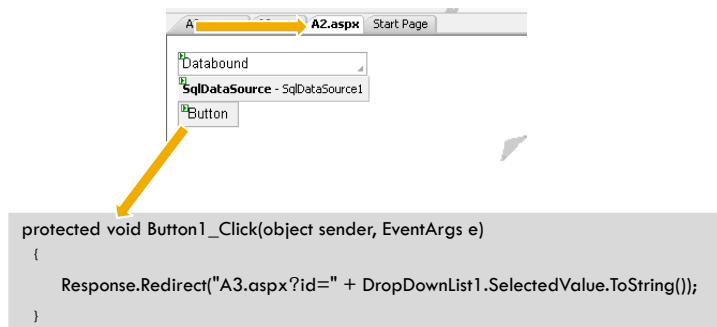
## Variable Transference



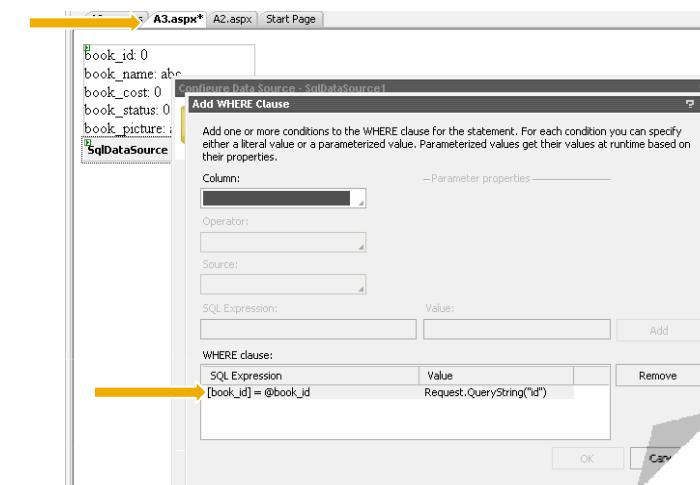
## Variable Transference



## Variable Transference



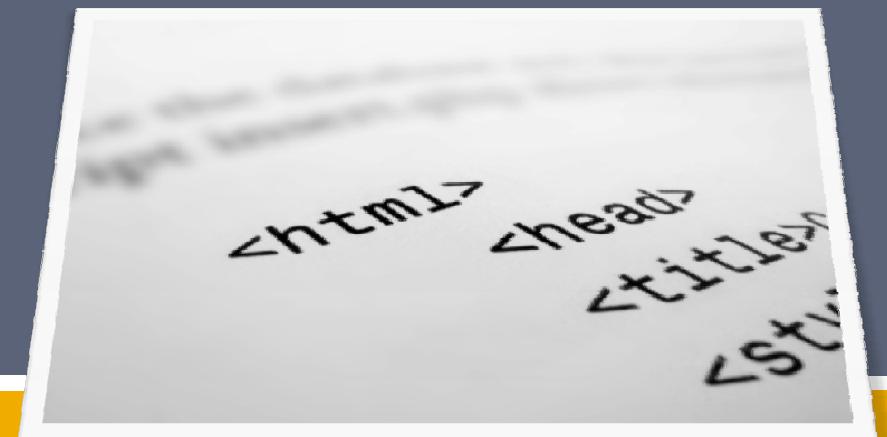
## Variable Transference



# ASP.NET WITH C#

## CLASS 6

อ.ณัฐภัทร แก้วรัตนภัทร์



# Database Connection Programming

- การเขื่อมต่อฐานข้อมูลโดยการเขียนโปรแกรมสามารถทำได้ โดยมีขั้นตอนหลัก ๆ ดังนี้
  - สร้างตัวติดต่อกับฐานข้อมูล
  - خذข้อมูลออกมาจากฐานข้อมูล
  - ถ่ายข้อมูลลง Data Set
  - ผูก Control เข้ากับ Data Set

# NameSpace for SQL Server

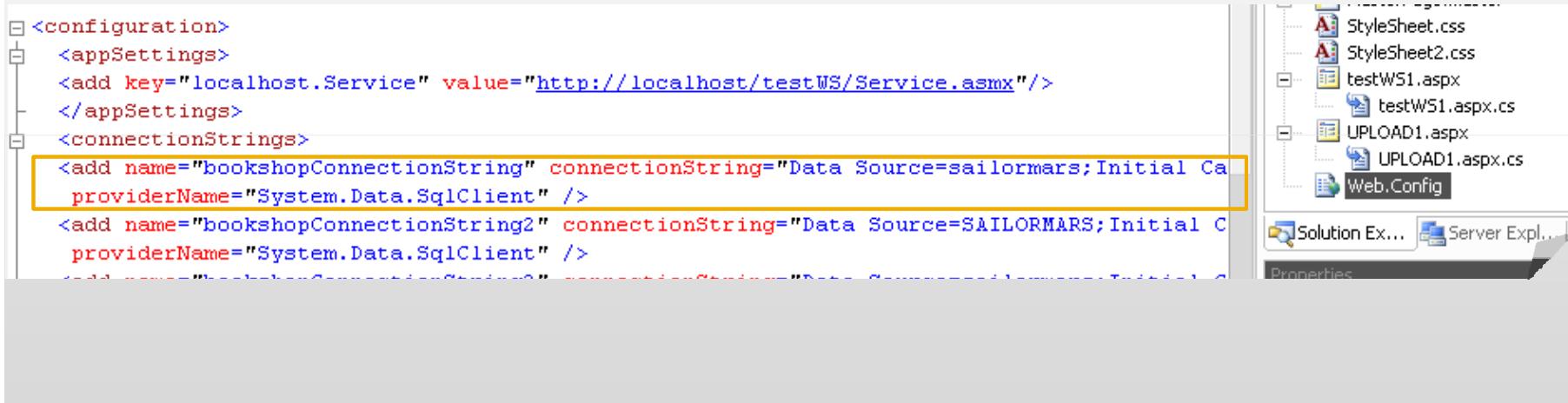
- `using System.Data.Sql;`
- `using System.Data.SqlClient;`

**Table 17-1** *Namespaces Used to Access ADO.NET Objects*

<b>Namespace</b>	<b>Contains</b>
<code>System.Data</code>	ADO.NET base objects
<code>System.Data.OleDb</code>	Managed OLEDB data store objects
<code>System.Data.SqlClient</code>	SQL Server-specific implementation of the ADO.NET objects
<code>System.Data.SqlTypes</code>	SQL Server data types

# DataBase Connection Style

- การเขียนต่อฐานข้อมูลและแสดงผลหรือนำข้อมูลมาจัดการและวิเคราะห์ต่างๆ มีวิธีการหรือมี **Style** หลากหลายวิธี ดังนี้
  - ▣ วิธีการแรก ใช้วิธีการใช้ **Control** ชื่อ **SQL Data Source** โดยไม่ต้องทำการเขียน **Code** ในส่วนของ **Connection** ติดต่อเอง
  - ▣ ชี้ **Code** การติดต่อนั้นจะไปเก็บไว้ในไฟล์ **Web.Config**



The screenshot shows the Visual Studio IDE interface. On the left, the Solution Explorer displays a project structure with files like StyleSheet.css, StyleSheet2.css, testWS1.aspx, testWS1.aspx.cs, UPLOAD1.aspx, UPLOAD1.aspx.cs, and Web.Config. The Web.Config file is currently selected and open in the main code editor window. The code in the editor is as follows:

```
<configuration>
  <appSettings>
    <add key="localhost.Service" value="http://localhost/testWS/Service.asmx"/>
  </appSettings>
  <connectionStrings>
    <add name="bookshopConnectionString" connectionString="Data Source=sailormars;Initial Catalog=sailormars;User ID=sa;Password=123456" providerName="System.Data.SqlClient" />
    <add name="bookshopConnectionString2" connectionString="Data Source=SAILORMARS;Initial Catalog=sailormars;User ID=sa;Password=123456" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

A yellow box highlights the connection string definition for 'bookshopConnectionString'.

# Example Code

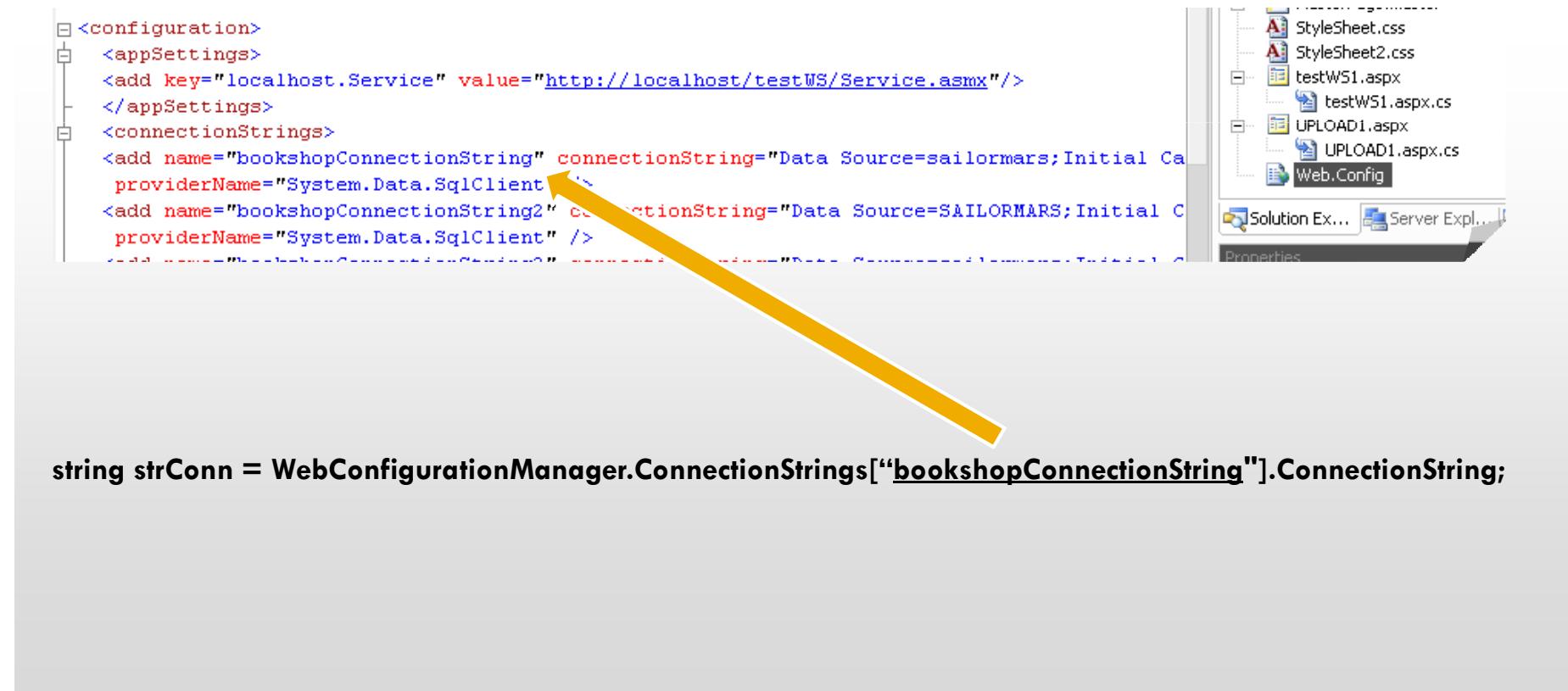
```
<connectionStrings>  
  <add name="bookshopConnectionString" connectionString="Data  
    Source=sailormars;Initial Catalog=bookshop;Integrated Security=True"  
    providerName="System.Data.SqlClient" />  
</connectionStrings>
```

# วิธีการแรก : ใช้ WebConfig

- วิธีการนี้เป็นวิธีการที่สามารถสร้าง **Code** การเขียนต่อไว้ที่ **Web.Config** ก่อนจากนั้นเราสามารถเข้าถึง **Code** นี้ได้โดยไม่ต้องสร้าง **Code** สำหรับ **Connection** ป้อยๆ แต่สามารถเรียกใช้จาก **Web.Config**

```
3 | using System.Data;
4 | using System.Configuration;
5 | using System.Collections;
6 | using System.Web;
7 | using System.Web.Configuration;
8 | using System.Web.Security;
9 | using System.Web.UI;
10| using System.Web.UI.WebControls;
11| using System.Web.UI.WebControls.WebParts;
12| using System.Web.UI.HtmlControls;
13| using System.IO;
14| using System.Data.SqlClient;
15| using System.Data.SqlClient;
16|
```

# WebConfigurationManager



The screenshot shows a Windows Development environment with a code editor and a Solution Explorer. The code editor displays the `Web.Config` file with several connection strings defined under the `<connectionStrings>` section. A yellow arrow points from the text `string strConn = WebConfigurationManager.ConnectionStrings["bookshopConnectionString"].ConnectionString;` in the code below to the `bookshopConnectionString` entry in the configuration file.

```
<configuration>
  <appSettings>
    <add key="localhost.Service" value="http://localhost/testWS/Service.asmx"/>
  </appSettings>
  <connectionStrings>
    <add name="bookshopConnectionString" connectionString="Data Source=sailormars;Initial Catalog=sailormars;User ID=sa;Password=sa" providerName="System.Data.SqlClient" />
    <add name="bookshopConnectionString2" connectionString="Data Source=SAILORMARS;Initial Catalog=sailormars;User ID=sa;Password=sa" providerName="System.Data.SqlClient" />
    <add name="bookshopConnectionString3" connectionString="Data Source=SAILORMARS;Initial Catalog=sailormars;User ID=sa;Password=sa" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

string strConn = WebConfigurationManager.ConnectionStrings["**bookshopConnectionString**"].ConnectionString;

# Object ด้านฐานข้อมูล

## □ Object ที่ควรทราบ

- **DataSet** คือ Object ที่เอาไว้เก็บตารางหรือ Data Table โดยสามารถเก็บ DataTable ได้หลาย Table
- **DataTable** คือ Object ของ ชุดของข้อมูลที่เราดึงมาจากฐานข้อมูล เช่น ค่า Table ที่ได้มาจาก Select \* from student เสมือนการดึง Table ออกมาจากฐานข้อมูล
- **DataView** คือ Object ที่เข้ามาช่วยเพิ่มความสามารถให้กับ DataTable เช่น ช่วยในการ Filter หรือ Sort ข้อมูลของ Record ตามที่เราต้องการ

# Object ด้านฐานข้อมูล

- **SqlConnection** เป็น Object ที่ใช้ในการสร้างการเชื่อมต่อไปยังฐานข้อมูล

ตัวอย่างการสร้างการเชื่อมต่อด้วย **SqlConnection**

```
string strConn =  
WebConfigurationManager.ConnectionStrings["RMSConnecti  
onString"].ConnectionString;
```

```
SqlConnection Conn = new SqlConnection(strConn);  
Conn.Open();
```

# Example Code

```
string sql = "select * from bookshop";  
  
string strConn = WebConfigurationManager.ConnectionStrings["RMSConnectionString"].ConnectionString;  
  
SqlConnection Conn = new SqlConnection(strConn);  
  
Conn.Open();  
  
SqlDataAdapter da = new SqlDataAdapter(sql, Conn);  
  
DataSet ds = new DataSet();  
  
da.Fill(ds, "temp");  
  
DropDownList1.DataSource = ds.Tables["temp"];  
  
DropDownList1.DataTextField = ds.Tables["temp"].Columns[1].ToString();  
  
DropDownList1.DataValueField = ds.Tables["temp"].Columns[0].ToString();  
  
DropDownList1.DataBind();
```

# Object ด้านฐานข้อมูล

- **SqlDataAdapter** เปรียบเสมือนตัวถังข้อมูลจากฐานข้อมูลโดยใช้คำสั่ง SQL ในการดึงข้อมูลที่สนใจมาเก็บไว้
- **DataSet** คือ Object ที่ใช้สำหรับถ่ายข้อมูลจาก DataAdapter มาเก็บไว้และสามารถนำไปใช้งานได้กับ Server Control ของ ASP.NET

ตัวอย่างการเชื่อมต่อ

Constructor ด้วยคำสั่ง sql  
และระบุตัวเชื่อมต่อ

```
SqlDataAdapter da = new SqlDataAdapter(sql, Conn);  
DataSet ds = new DataSet();  
da.Fill(ds, "temps");
```

Adapter จะแปลง  
ข้อมูลที่ดึงมาเก็บลง  
DataSet

ตั้งชื่อให้กับตารางที่จะ  
เก็บลง DataSet

# Example

- ทดลองนำข้อมูลมาแสดงใน **DropDownList**

```
DropDownList1.DataSource = ds.Tables["temps"];
```

ระบุแหล่งข้อมูลที่ใช้

```
DropDownList1.DataTextField = ds.Tables["temps"].Columns[1].ToString();
```

```
DropDownList1.DataValueField = ds.Tables["temps"].Columns[0].ToString();
```

```
DropDownList1.DataBind();
```

ผูกข้อมูลกับ Control

ใช้ข้อมูลจากคอลัมน์ที่  
0 ทั้งหมด

# Clear & Changing

- หากต้องการเปลี่ยนการคิวรีข้อมูล ให้กระทำกับ **DataAdapter** ตัวเดียว ดัง **Code** นี้

```
sql = "select * from admin where ADM_id ='" +  
Session["adminid"].ToString() + "'";  
  
da = new SqlDataAdapter(sql, Conn);  
  
ds.Tables["temp"].Clear();  
  
da.Fill(ds, "temp");
```

Query ข้อมูลใหม่

ลบตารางชื่อ Temp  
ใน DataSet กรนี  
ต้องการอัพเดทข้อมูล  
ใหม่ในตาราง temp

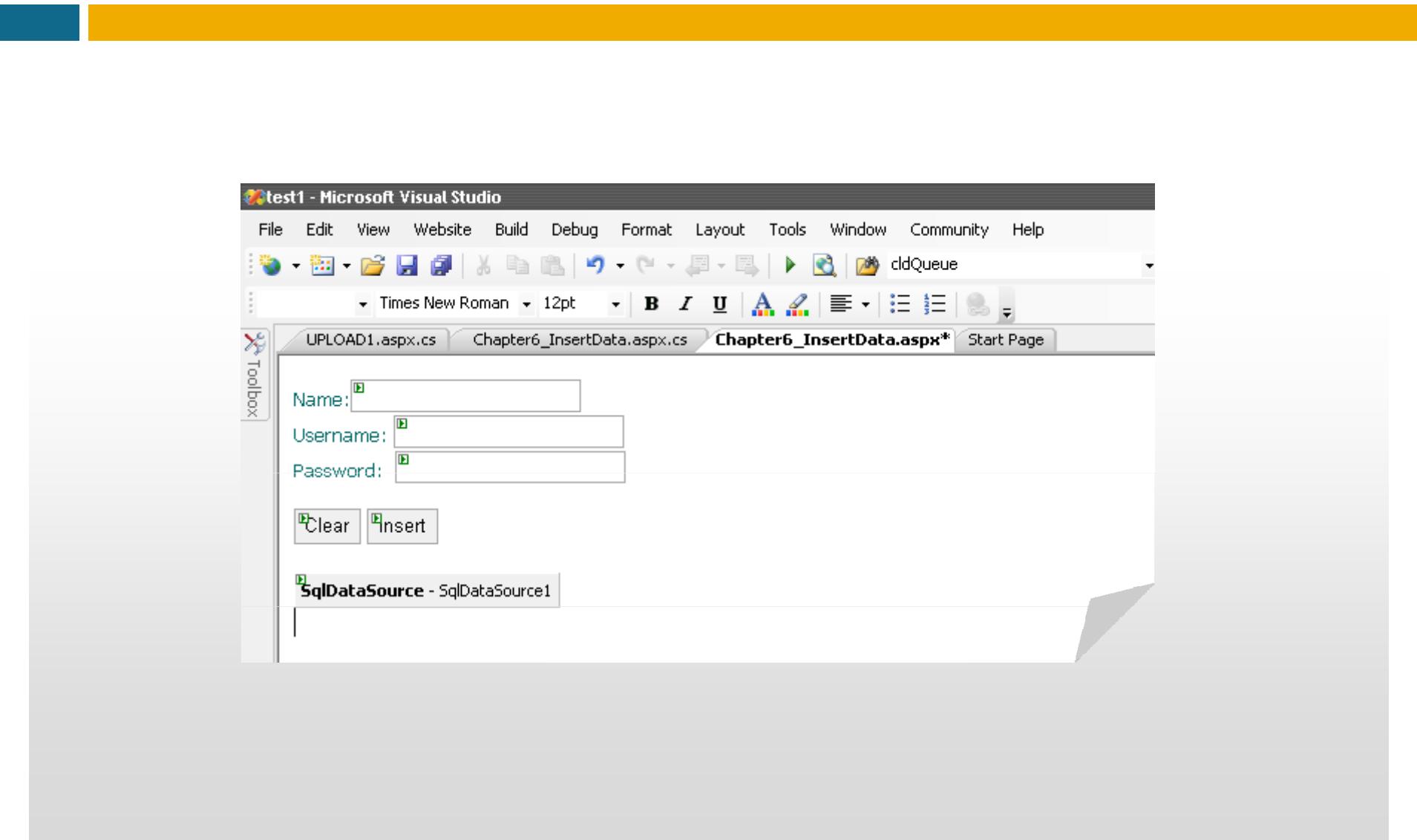
Fill (เติม) ค่าใหม่ลง  
ไปในตาราง Temp

# Row Data

- หากต้องการนำข้อมูลที่คิวอาร์มาแสดงให้ใช้ **Code** ดังนี้
  - `lbl_id.Text = ds.Tables["temps"].Rows[0]["ADM_id"].ToString();`



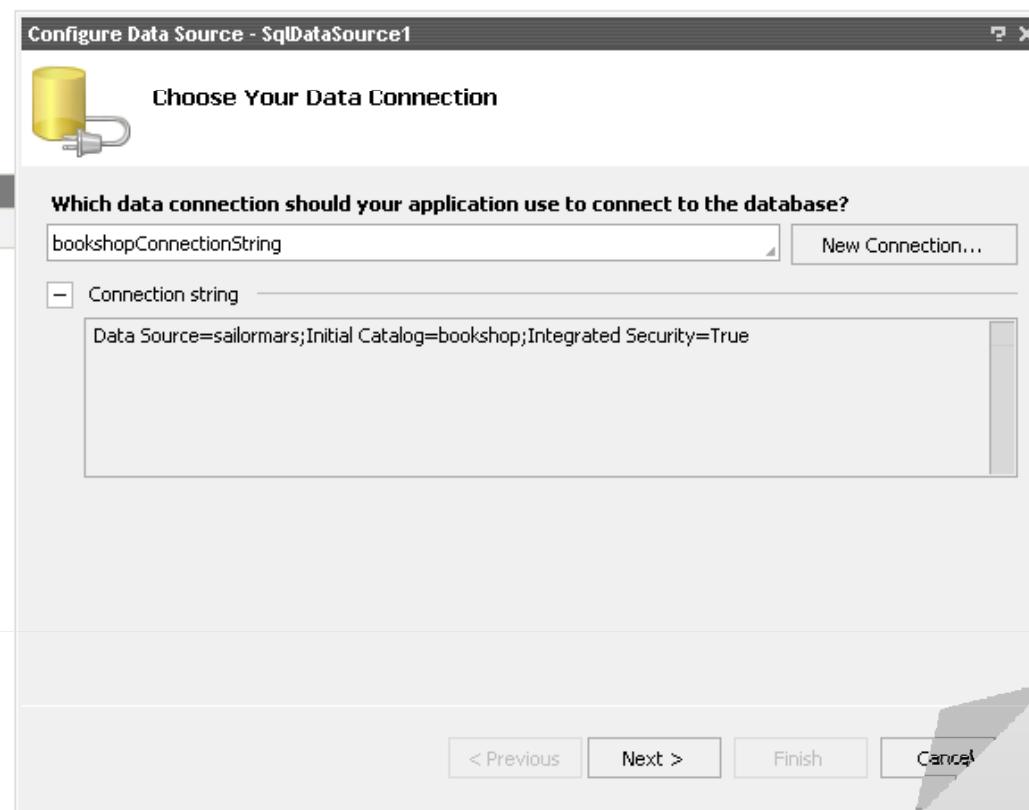
# Insert Data



# Insert Data

Username:   
Password:

**SqlDataSource - SqlDataSource1**  
[Configure Data Source...](#)



# Insert Data

The screenshot shows a web application interface with a blue header bar. Below the header, there are three input fields: 'Name:' with value 'admin1', 'Username:' with value 'admin', and 'Password:' with value '123456'. Below these fields are two buttons: 'Clear' and 'Insert'. To the right of these fields is a configuration dialog window titled 'Configure Data Source - SqlDataSource1'.

**Configure Data Source - SqlDataSource1**

**Configure the Select Statement**

**How would you like to retrieve data from your database?**

Specify a custom SQL statement or stored procedure  
 Specify columns from a table or view

Name: admin1

Columns:

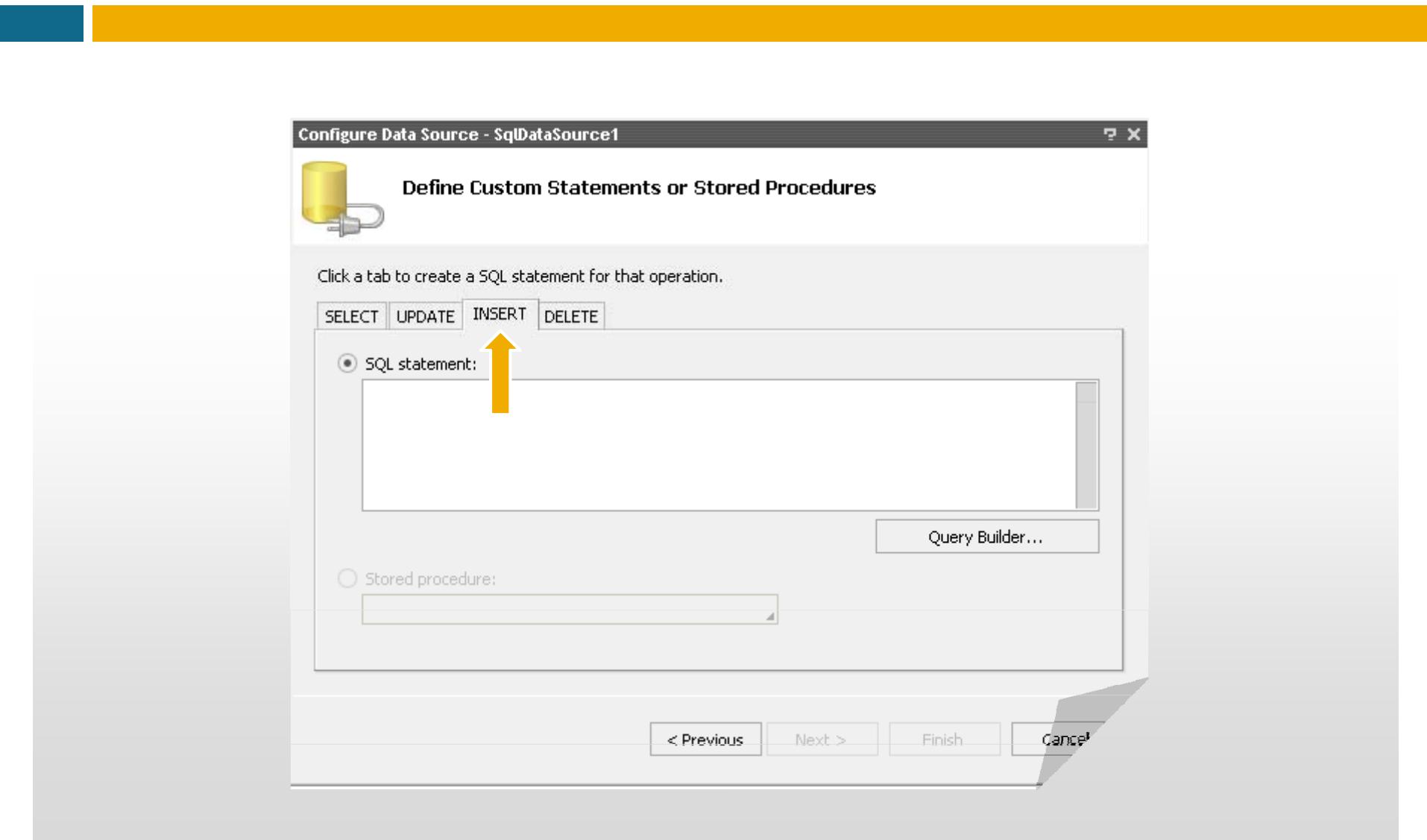
- \*
- admin\_id
- admin\_name
- admin\_username
- admin\_password

Return only unique rows  
WHERE...  
ORDER BY...  
Advanced...

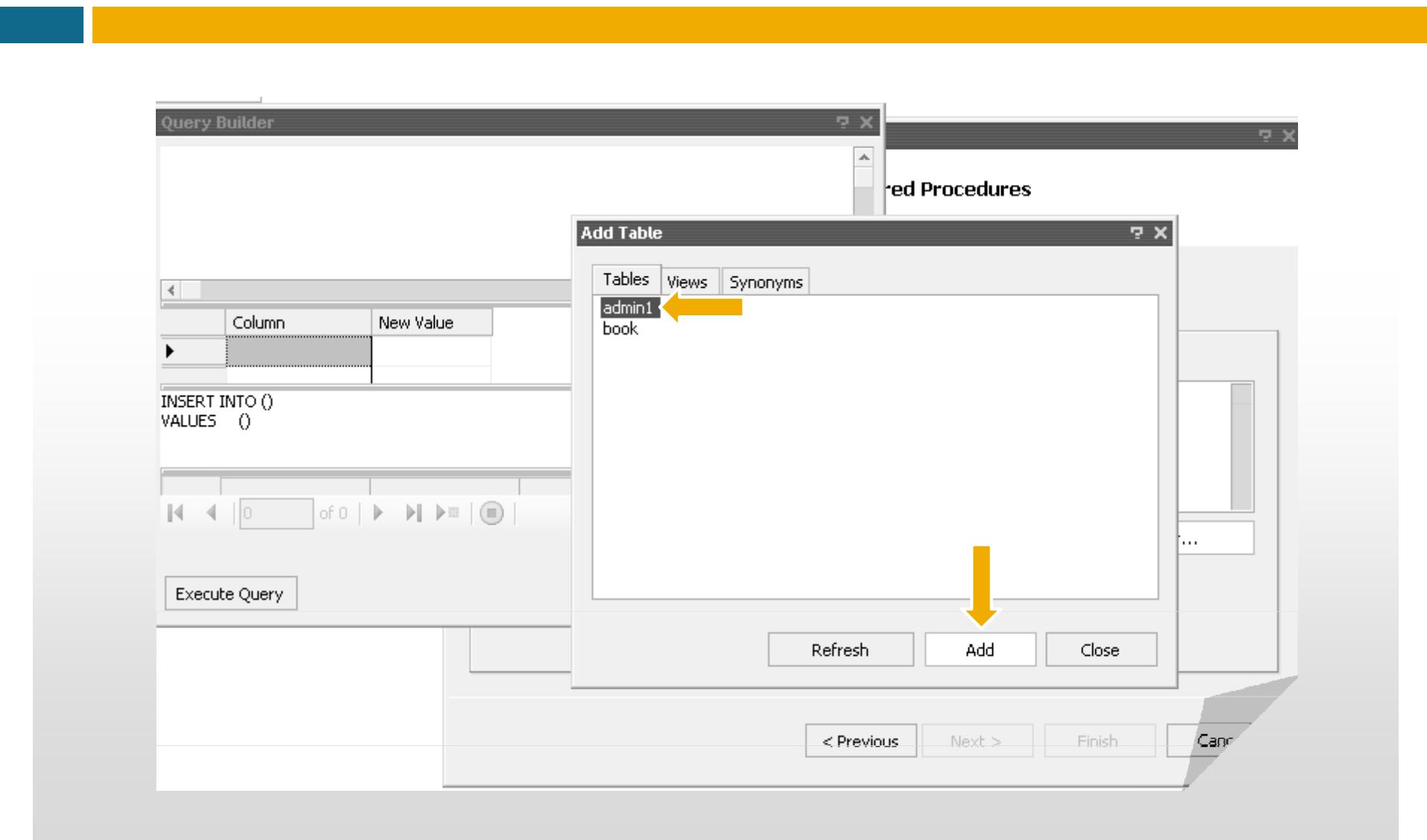
SELECT statement:

< Previous    Next >    Finish    Cancel

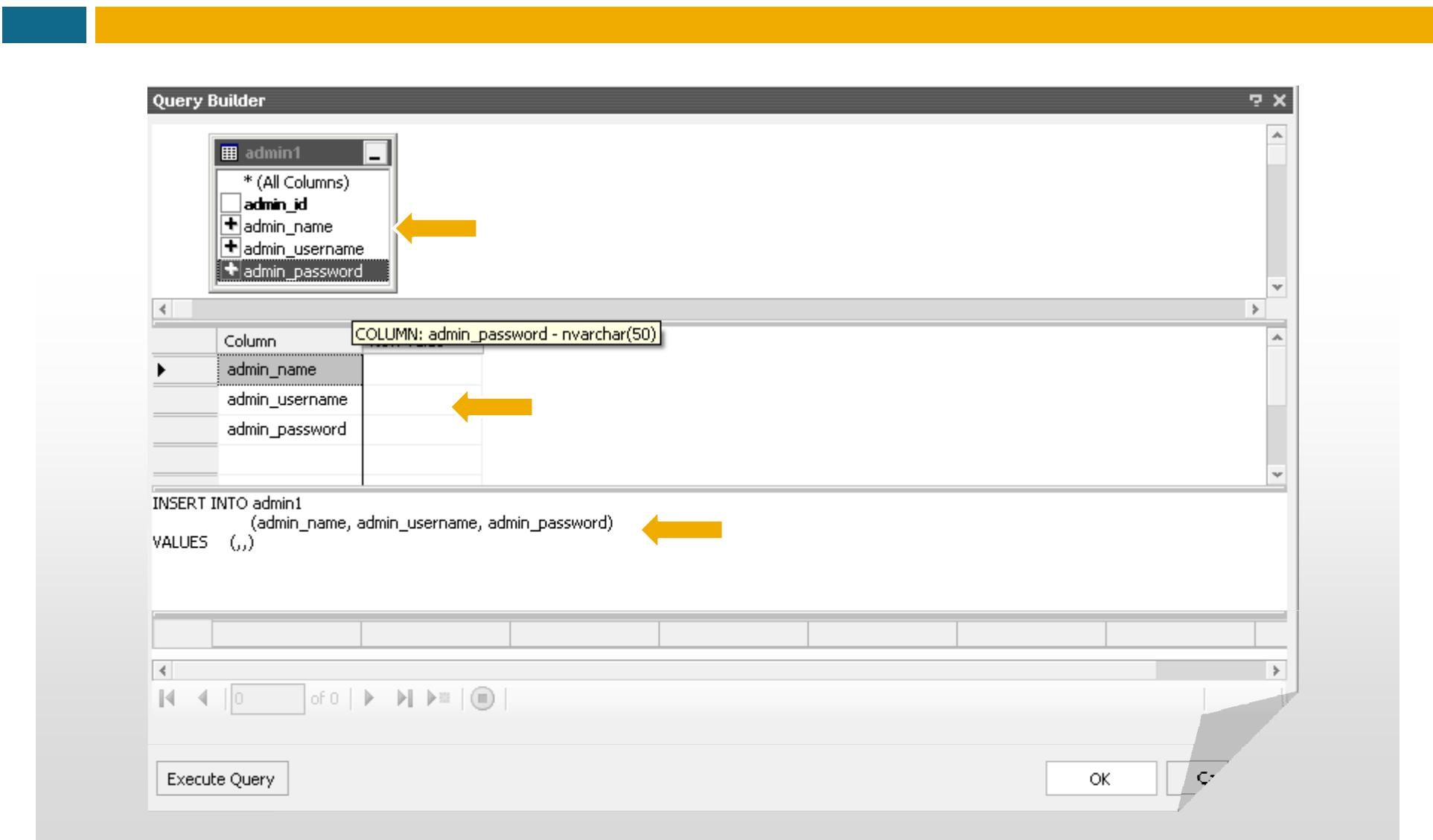
# Insert Data



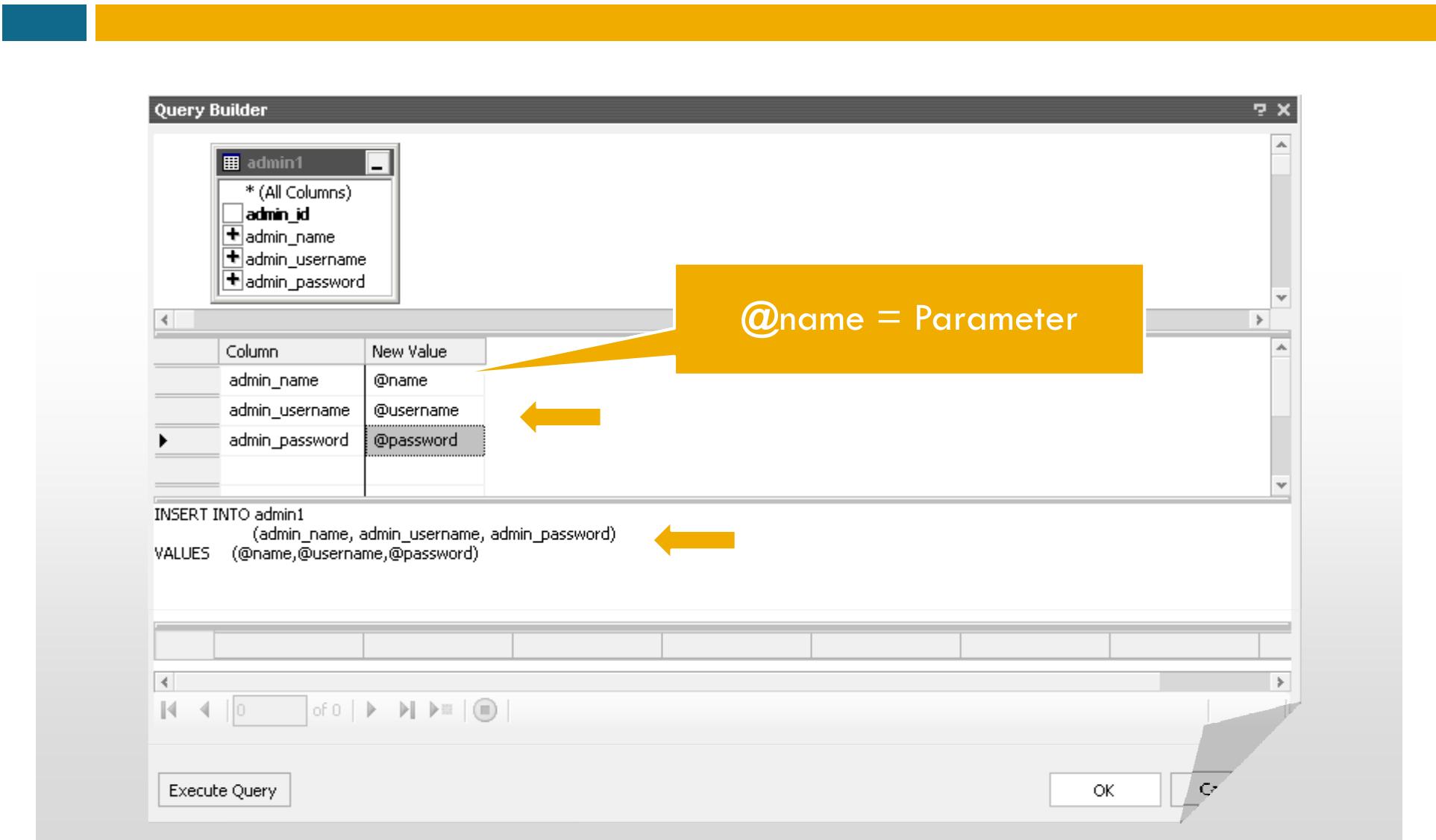
# Insert Data



# Insert Data



# Insert Data



# Insert Data

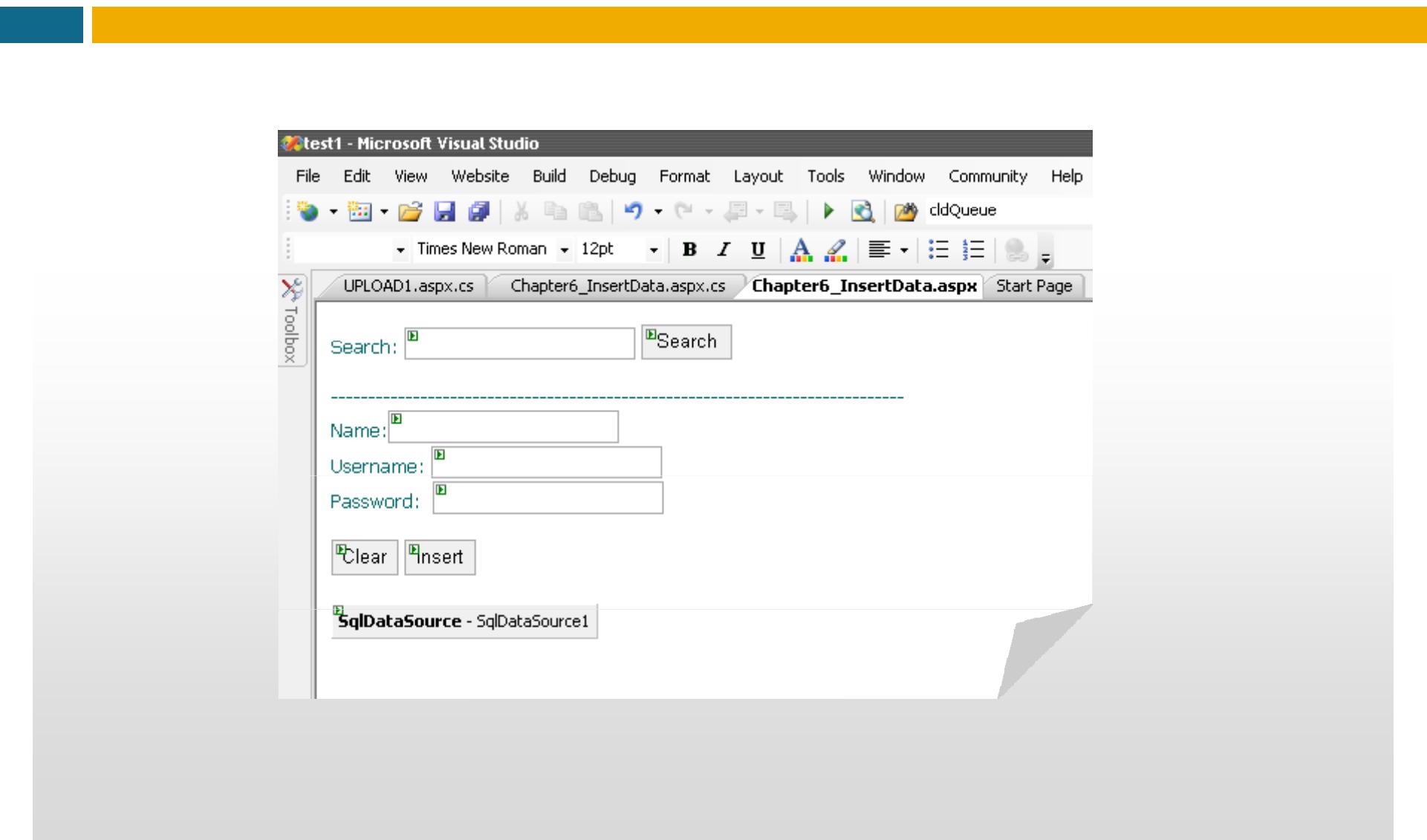
```
protected void Button2_Click(object sender, EventArgs e)
{
    SqlDataSource1.InsertParameters["name"].DefaultValue = TextBox1.Text.ToString();
    SqlDataSource1.InsertParameters["username"].DefaultValue = TextBox2.Text.ToString();
    SqlDataSource1.InsertParameters["password"].DefaultValue = TextBox3.Text.ToString();
    SqlDataSource1.Insert();
    cleardata();
}
```

ชื่อ Parameter จะต้องตรงกับที่  
สร้างไว้ในคำสั่ง SQL (@name)

```
SqlDataSource1.InsertParameters["name"].DefaultValue = TextBox1.Text.ToString();
SqlDataSource1.InsertParameters["username"].DefaultValue = TextBox2.Text.ToString();
SqlDataSource1.InsertParameters["password"].DefaultValue = TextBox3.Text.ToString();
SqlDataSource1.Insert();
```

Function สำหรับการ Insert

# Update Data



# Update Data

Query Builder

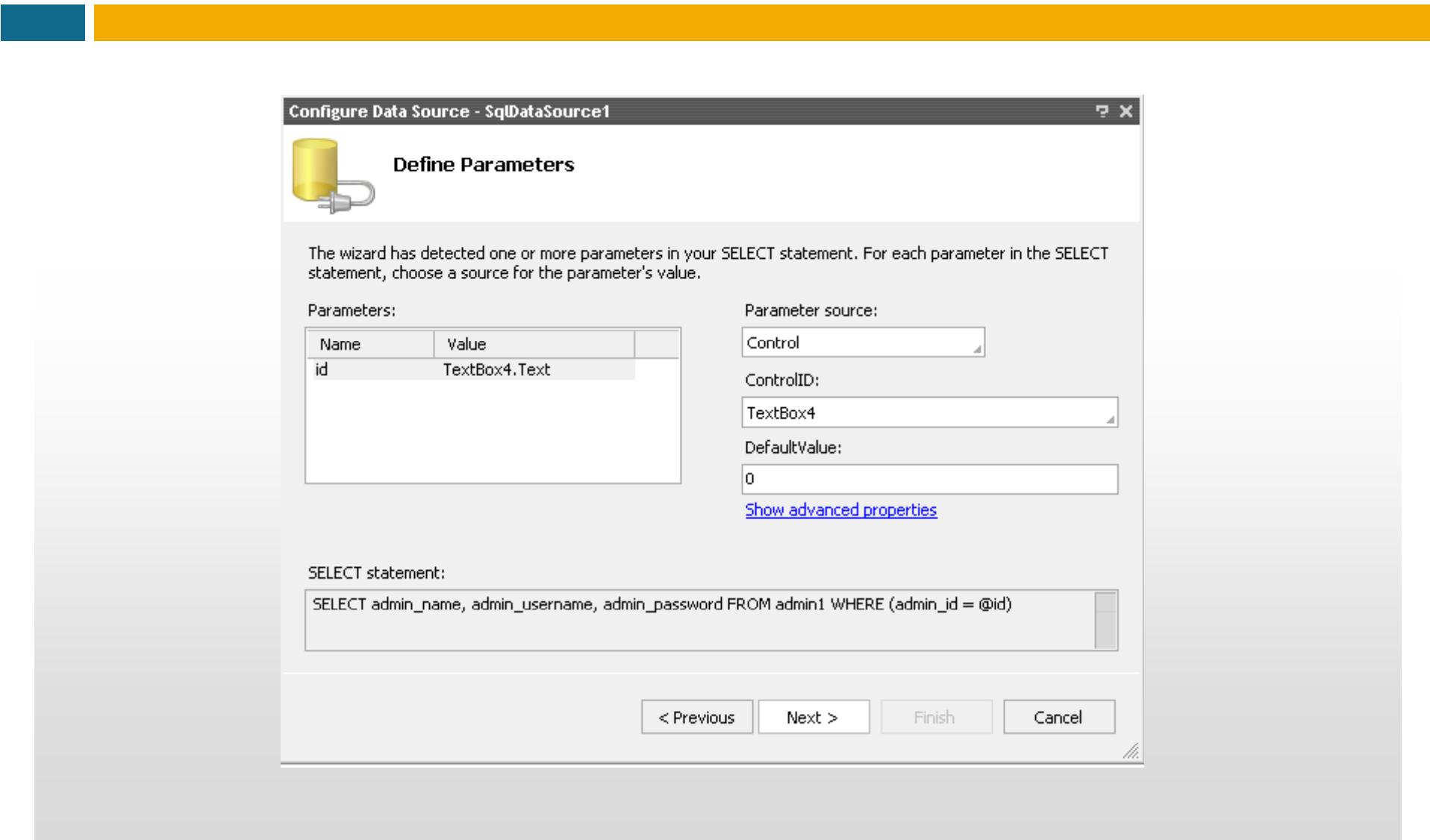
Filter ด้วย @id

The screenshot shows the Query Builder interface with the following details:

- Table Selection:** A tree view on the left shows a table named "admin1" with columns: \* (All Columns), admin\_id, admin\_name, admin\_username, and admin\_password. The "admin\_id" column is selected.
- Query Grid:** The main grid displays the query structure. The first row has columns: Column, Alias, Table, Output, Sort Type, Sort Order, Filter, Or..., Or..., Or... . The "admin\_id" column is mapped to the alias "@id" with the filter condition "@id".
- SQL Preview:** Below the grid, the generated SQL code is shown:

```
SELECT admin_name, admin_username, admin_password
FROM admin1
WHERE (admin_id = @id)
```
- Execution Panel:** At the bottom, there is a toolbar with navigation icons (back, forward, search) and a status bar indicating "0 of 0".
- Buttons:** At the very bottom are "Execute Query" and "OK" buttons.

# Update Data



# Update Data

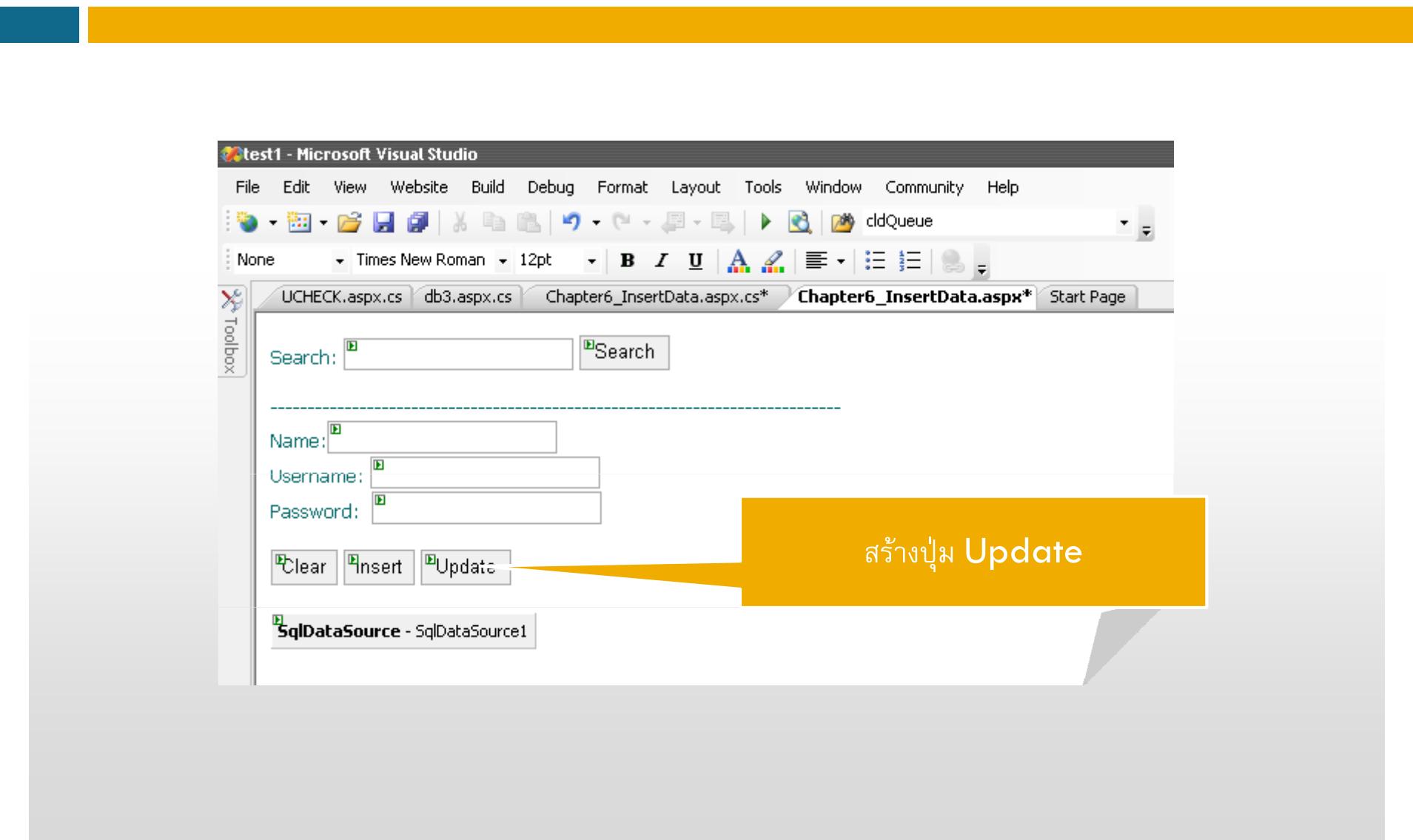
```
protected void Button3_Click(object sender, EventArgs e)
{
    SqlDataSource1.SelectParameters["id"].DefaultValue = TextBox4.Text.ToString();
    DataView dv = (DataView)SqlDataSource1.Select(DataSourceSelectArguments.Empty);
    TextBox1.Text = dv.Table.Rows[0][0].ToString();
    TextBox2.Text = dv.Table.Rows[0][1].ToString();
    TextBox3.Text = dv.Table.Rows[0][2].ToString();
}
```

Select Data ที่ต้องการแก้ไข

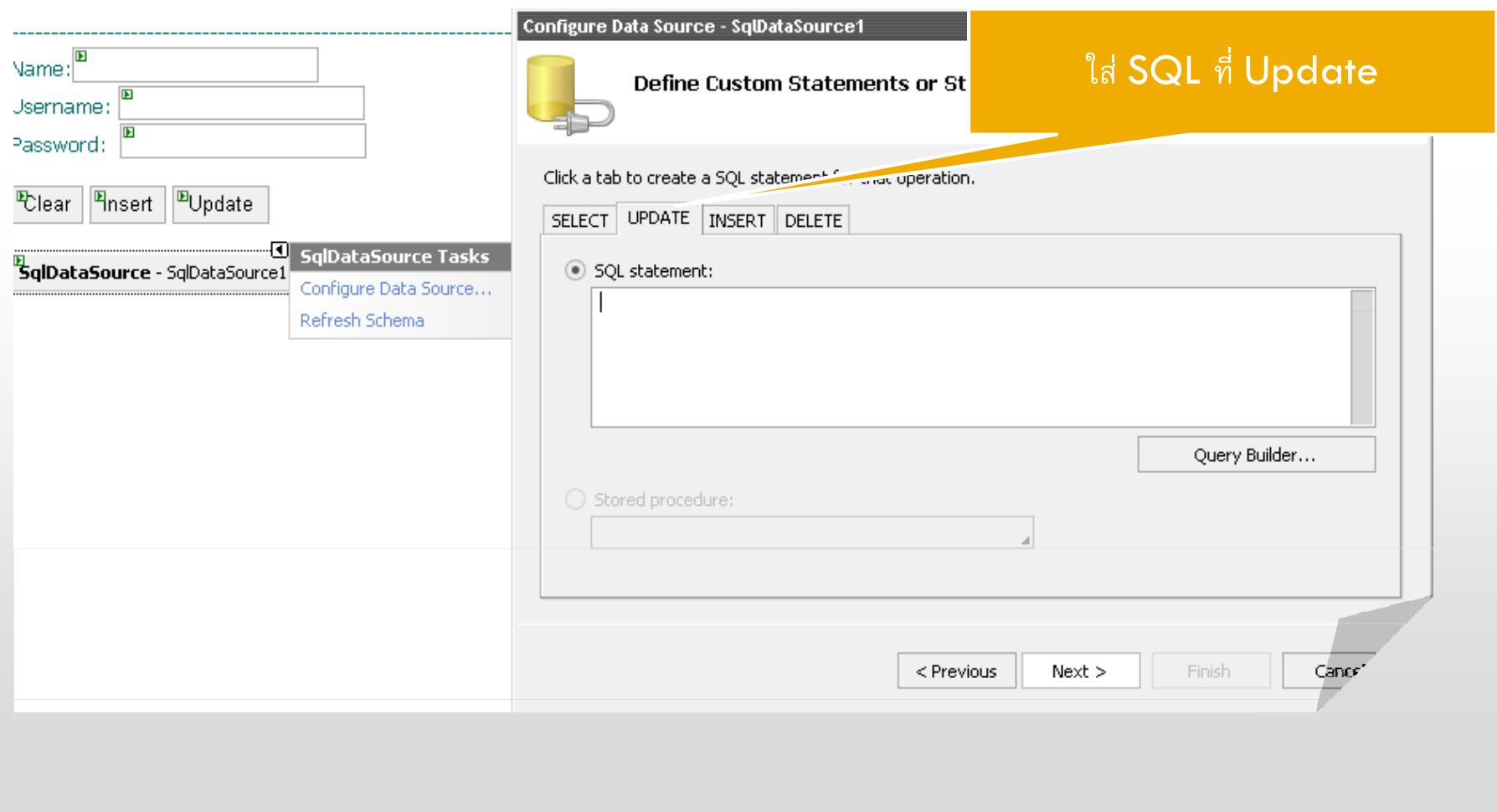
เอา DataView มารับข้อมูล

ถ่ายค่าลง Control เพื่อแสดงผล

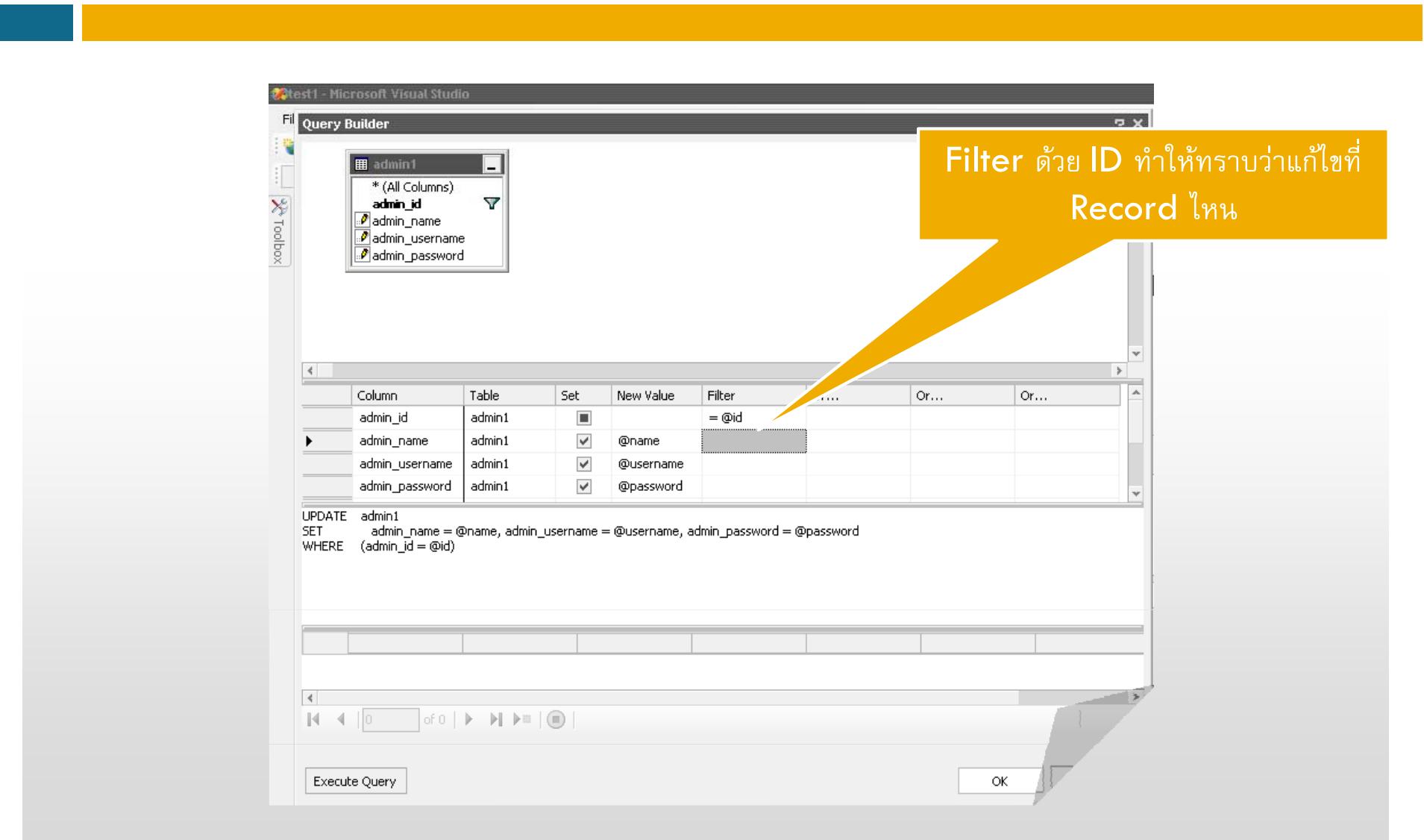
# Update Data



# Update Data



# Update Data



# Update Data

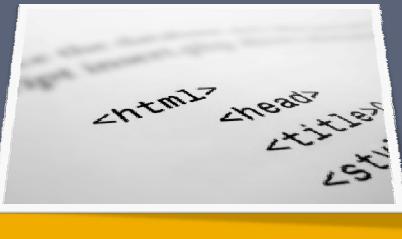
กำหนดค่าต่างๆ ให้ตรงกับ Field ที่  
กำหนดไว้

```
protected void Button4_Click(object sender, EventArgs e)
{
    SqlDataSource1.UpdateParameters["id"].DefaultValue = TextBox4.Text.ToString();
    SqlDataSource1.UpdateParameters["name"].DefaultValue = TextBox1.Text.ToString();
    SqlDataSource1.UpdateParameters["username"].DefaultValue = TextBox2.Text.ToString();
    SqlDataSource1.UpdateParameters["password"].DefaultValue = TextBox3.Text.ToString();
    SqlDataSource1.Update();
    cleardata();
}
```

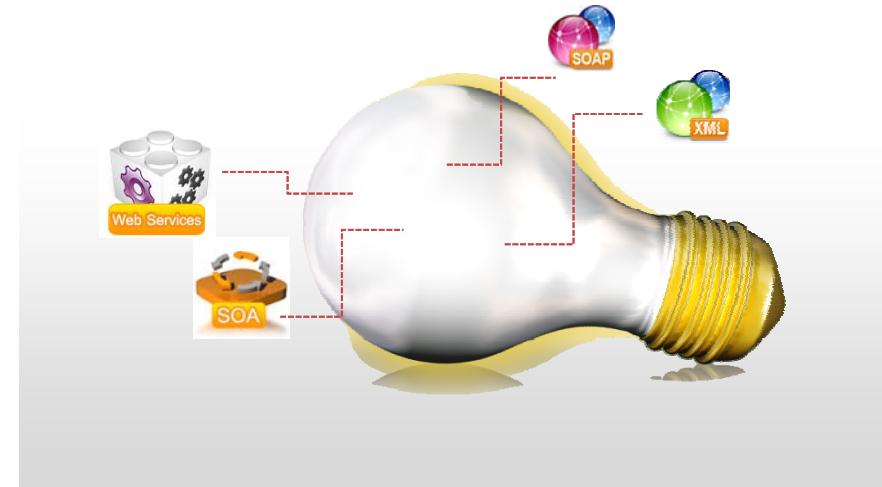
ใช้ Method Update ในการบันทึก

## ASP.NET WITH C# CLASS 7

อ.ณัฐภัทร แก้วรัตนภัทร



## Web Services



### เปรียบเทียบการทำงาน

- หากนักศึกษาเป็นเจ้าของเว็บไซต์และบริหารงานคนเดียวไม่มีคนช่วยงานเลย และทุกๆ วัน นักศึกษาต้องเข้าไปที่เว็บไซต์ของไทยรัฐ เพื่อคัดลอกข่าวมาใส่ในฐานข้อมูล จากนั้นหน้าเว็บของนักศึกษาจะอัพเดทข่าวใหม่ทุกวัน
- แล้วหากว่านักศึกษาไปพักผ่อนต่างประเทศ 1 อาทิตย์ โดยประเทศที่ไปใช้คอมพิวเตอร์และเครือข่ายอินเทอร์เน็ตไม่ได้ เว็บไซต์ของนักศึกษาจะอัพเดทข่าวได้เช่นไร?

## Web Services

- Web Services ไม่ใช่ Web Site
- Web Services คือ พังก์ชั่นที่เขียนขึ้นเพื่อให้ผู้ใช้ในเครือข่ายสามารถเรียกใช้พังก์ชั่นนี้ได้
- Web Services ทำให้เกิด Machine to Machine
- Web Services ใช้สถาปัตยกรรมแบบ SOA นั่นหมายความว่า เว็บเซอร์วิสต้องมีไฟล์ที่นิยามด้วยเว็บเซอร์วิสอง (WSDL) เพื่อที่จะให้ผู้ที่ต้องการใช้มันเข้าใจ

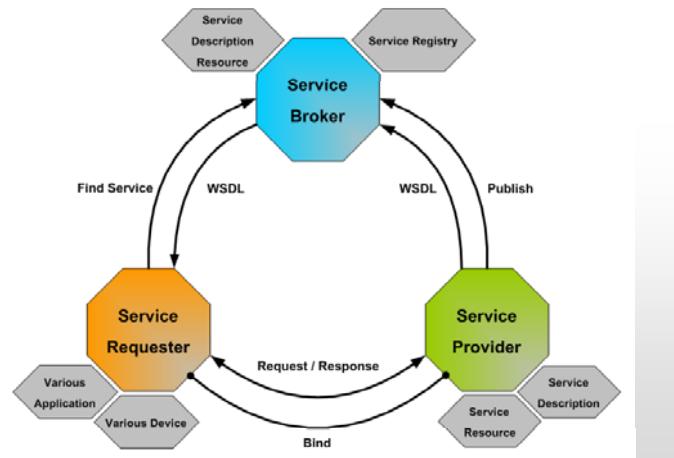
## Web Services

- แอพพลิเคชันที่เป็นเว็บเซอร์วิสสามารถทำงานในมาตรฐานระบบภาษาโปรแกรมที่ใช้สำหรับพัฒนา หรือ รูปแบบของระบบ (Platform) ของตนเอง แต่สามารถ “พูดคุย” หรือ ส่งข้อความ (Message) กับแอพพลิเคชันหรือเว็บเซอร์วิสอื่นๆ โดยไม่ยึดติดกับมาตรฐานและเทคโนโลยีของแต่ละฝ่าย (Platform Independence) ทั้งนี้เพื่อประโยชน์ในการแลกเปลี่ยนข้อมูลและบริการร่วมกันโดยไม่ต้องอาศัยมันมุชย์เพื่อแปลความหมายของการสนทนานั้น รวมถึงการขอใช้ฟังก์ชันโปรแกรมต่างๆ เรียกว่า “Remote Procedure Call: RPC” ทำให้เครื่องคอมพิวเตอร์สามารถสนทนาทำการแลกเปลี่ยนในเชิงของข้อมูลและการประมวลผลได้

## Web Services

- เว็บเซอร์วิสใช้แนวคิดจากสถาปัตยกรรมการให้บริการแบบเอสโซเอ (Service-Oriented Architecture: SOA) ซึ่งหมายถึง พังก์ชัน หรือแอพพลิเคชันที่ทำงานเฉพาะด้านขนาดไม่ใหญ่ แต่สมมูลเป็นซอฟต์แวร์ หรือพังก์ชันงานย่อยที่รอกองบกันเท่านั้น **ไม่ใช่ซอฟแวร์แพ็คเกจ (Package Software)** ที่มีขนาดใหญ่ แนวคิดของ SOA ทำให้เกิดการปรับเปลี่ยนแนวทางการพัฒนาระบบท่างๆ ที่ทำงานร่วมกับเครือข่ายอินเทอร์เน็ต

## Service Oriented Architecture :SOA



## ผู้ร้องขอใช้บริการ (Service Requester)

- ผู้ร้องขอใช้บริการ (Service Requester) หมายถึง ผู้ใช้ (User) อุปกรณ์ (Various Device) หรือ แอพพลิเคชัน (Application) ที่มีความต้องการใช้บริการใดๆ ที่สามารถตอบสนองต่อความต้องการของผู้ร้องขอได้จากผู้ตระเตรียมบริการ (Service Provider) ซึ่งผู้ร้องขอใช้บริการจะทำการค้นหา (Find) บริการได้จากนายหน้าผู้ให้บริการ (Service Broker) และจะได้รับเอกสาร WSDL เพื่อแสดงรายละเอียดที่จะใช้บริการนั้นๆ ทำให้ผู้ร้องขอใช้บริการส่งค่าพารามิเตอร์ (Parameters) ไปยังบริการของผู้ตระเตรียมบริการ และทำการรับผลลัพธ์ที่สนองต่อการร้องขอใช้บริการกลับคืนมา

## นายหน้าผู้ให้บริการ (Service Broker)

- นายหน้าผู้ให้บริการ (Service Broker) หมายถึง แหล่งที่บันทึกข้อมูลการลงทะเบียนบริการ (Service Registry) โดยนายหน้าผู้ให้บริการแทนด้วย UDDI ในเทคโนโลยีเว็บเซอร์วิส ซึ่งมีไว้สำหรับผู้ตระเตรียมบริการที่ต้องการเผยแพร่ (Publish) บริการการทำงานของตน เพื่อเป็นประโยชน์ต่อผู้พัฒนาระบบที่มีความต้องการใช้บริการนั้นๆ โดยนายหน้าผู้ให้บริการจะมีการเก็บข้อมูลเอกสาร WSDL ให้แก่ผู้ร้องขอใช้บริการเพื่อแสดงรายละเอียดของบริการนั้นๆ และเมื่อผู้ร้องขอใช้บริการทราบข้อมูลจากเอกสาร WSDL แล้ว จะทำการผูก (Bind) กับผู้ตระเตรียมบริการโดยตรง ซึ่งไม่จำเป็นต้องติดต่อกับนายหน้าผู้ให้บริการอีก ในการผูกเพื่อบริการนี้จะมีการใช้ SOAP เป็นโปรโตคอลในเทคโนโลยีเว็บเซอร์วิสและลักษณะการทำงานจะมีการร้องขอ (Request) ไปยังบริการ เมื่อได้ผลลัพธ์จะทำการสนองกลับ (Response) หมายง่ายๆ ขอใช้บริการ

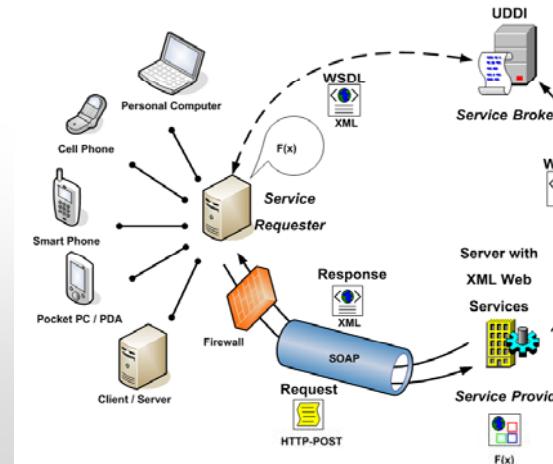
## ผู้ตระเตรียมบริการ (Service Provider)

- ผู้ตระเตรียมบริการ (Service Provider) หมายถึง ผู้ที่พัฒนาและสร้างบริการต่างๆ สำหรับใช้พัฒนาระบบงานของตนเองและผู้อื่น โดยมีแหล่งเก็บบริการและเอกสาร WSDL ของตนเอง ซึ่งผู้ตระเตรียมบริการจะมีการเผยแพร่บริการโดยการลงทะเบียนและใช้เอกสาร WSDL ฝากไว้กับนายหน้าผู้ให้บริการ เพื่อให้ผู้ร้องขอใช้บริการสืบค้นบริการต่างๆ ที่ต้องการได้

## WSDL SOAP UDDI

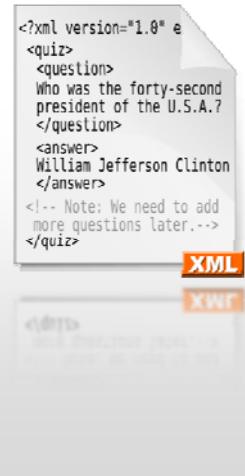
- ในการทำงานของแอปพลิเคชันทางด้านเว็บเซอร์วิสจะต้องมีการอธิบายเซอร์วิส หรือ พังก์ชันที่ให้บริการด้วยภาษาเอ็กซ์เชิร์ฟและโดยจะเขียนไว้ใน **ดับเบิลยูดีЭЛЭДЭЛ (Web Service Description Language: WSDL)** ทำให้สามารถเข้าใจการใช้เซอร์วิสนั้นได้ เว็บเซอร์วิสจะใช้ภาษาเอ็กซ์เชิร์ฟและอัลติಡต่อสื่อสารบนโครงสร้างพื้นฐานของโปรโตคอล HTTP ที่มีอยู่แล้ว โดยการรวมตัวกันระหว่างภาษาเอ็กซ์เชิร์ฟและ กับ โปรโตคอล HTTP มีชื่อเรียกว่า **เออสโอล็อกซ์ หรือ โซพ (Simple Object Access Protocol: SOAP)** และเมื่อมีการพัฒนาเว็บเซอร์วิสสำหรับใช้งานแล้วจะต้องมีการเผยแพร่เว็บเซอร์วิสเพื่อให้พันธมิตรหรือผู้ใช้งานสามารถเข้ามาขอใช้บริการได้ โดยการเผยแพร่นั้นมีการกระทำผ่านสองมาตรฐาน คือ **ยูดีดีไอ (Universal Description Discovery and Integration: UDDI)** และ **ดิสโคฟเวอรี่ (Discovery: DISCO)**

## Web Services System

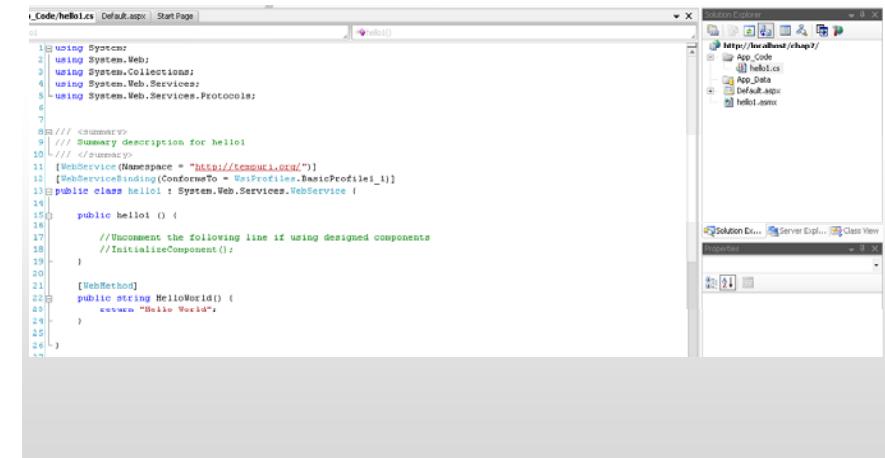


## eXtensible Markup Language : XML

- ภาษาเอ็กซ์เพ็มแอล (XML) คือ ภาษาโปรแกรมสำหรับนิยามข้อมูลที่ทำให้ผู้ใช้สามารถสร้างและจัดการเอกสารได้ โดยภาษาเอ็กซ์เพ็มแอลเป็นภาษาที่มีโครงสร้าง (Structured Documents) ภายในเอกสารจะบรรจุตัวอักษร (Plain Text) ซึ่งทำให้สามารถปรับเปลี่ยนการแสดงผล (Rendering) ในรูปแบบที่หลากหลาย จุดประสงค์หลักของเอ็กซ์เพ็มแอล คือ การแยกส่วนข้อมูลเพื่อประโยชน์ในการนำไปใช้ร่วมกับการพัฒนาหรือใช้ร่วมกับโปรแกรม หรือ แอ��พลิเคชัน อื่นๆ ได้



## Demo HelloWebServices



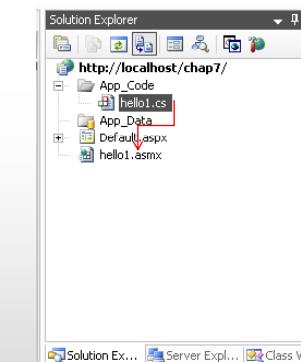
## Demo : Structure

- โครงสร้างของ Web Services คล้ายกับโครงสร้างของฟังก์ชัน

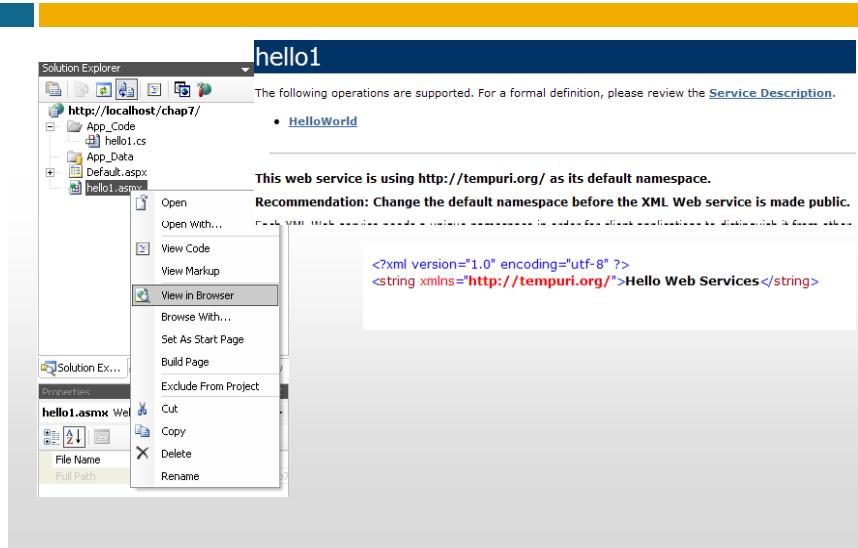
[WebMethod]

```
public string HelloWorld() {
    return "Hello Web Services";
}
```

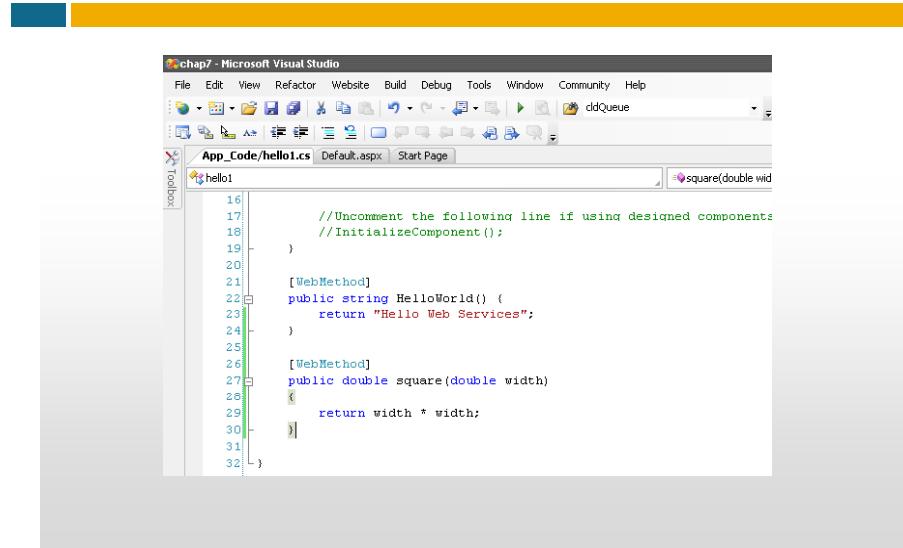
## WS Files



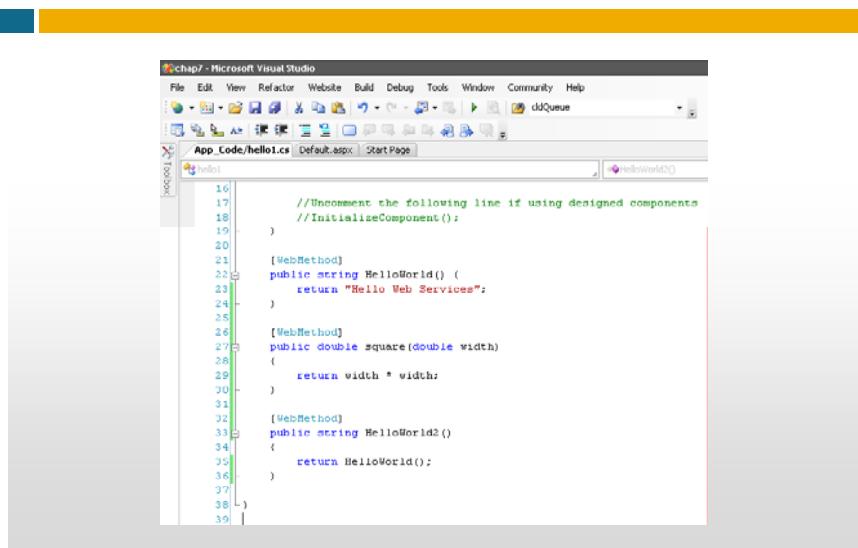
## RUN



## New Web Method

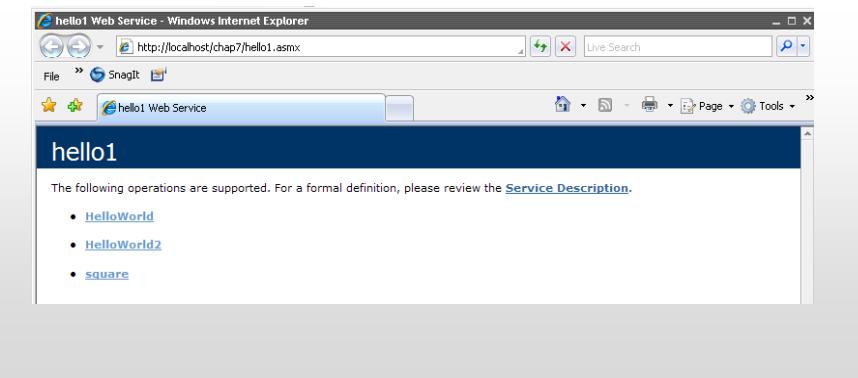


## Call other function



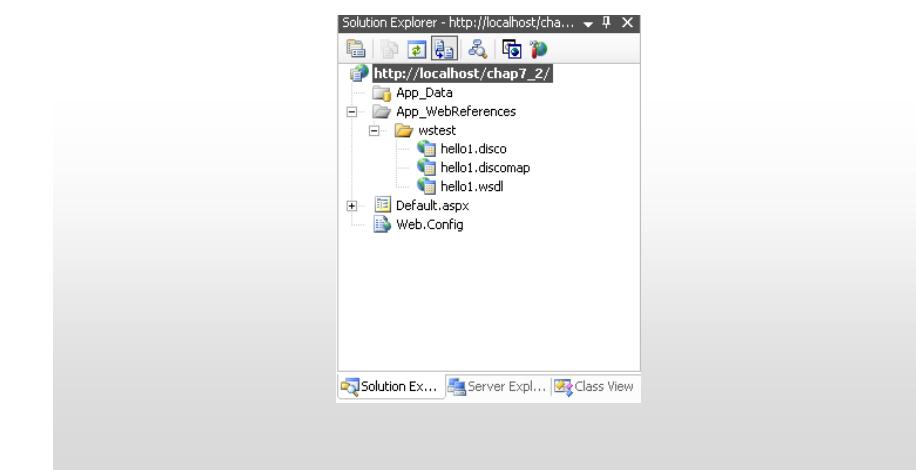
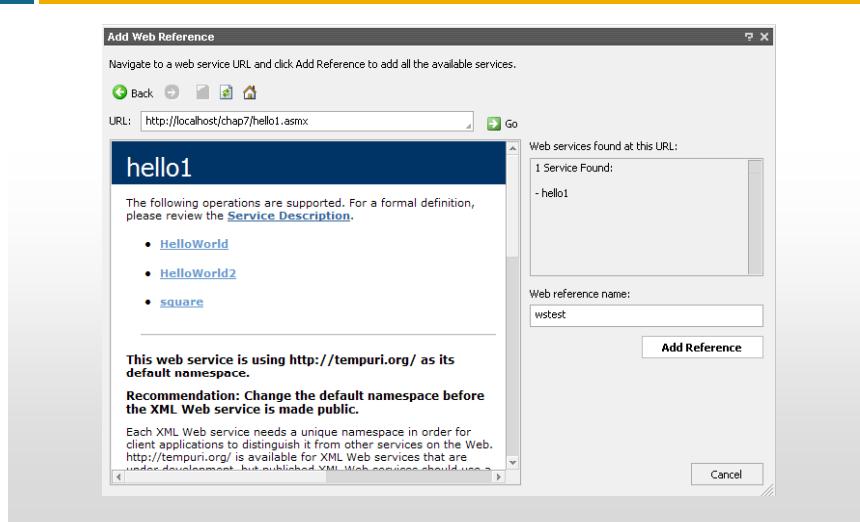
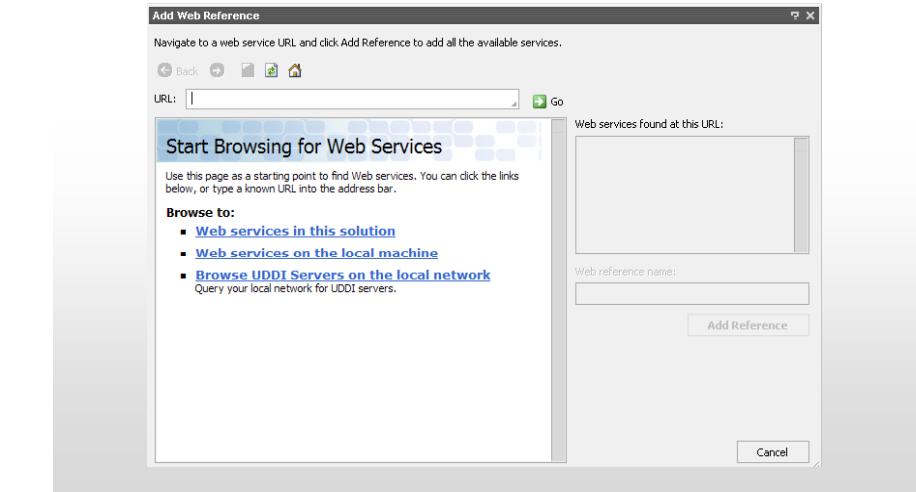
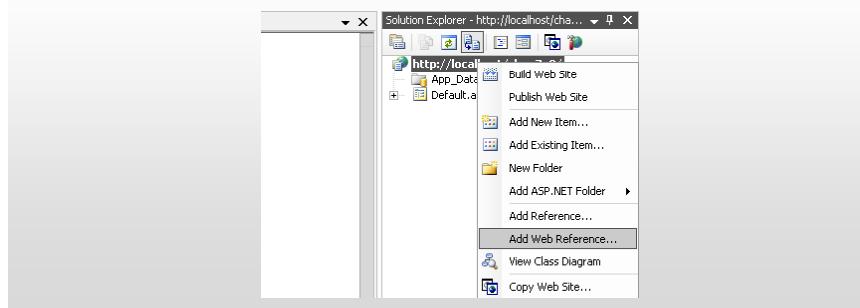
## การใช้ Services

- ให้นักศึกษา RUN ไฟล์ นามสกุล \*.asmx ไว้ โดยห้ามปิดขณะที่จะทำการเพิ่มลง Web References



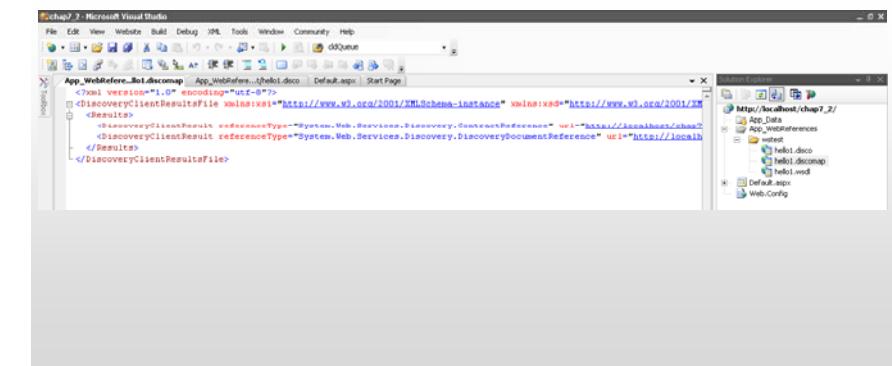
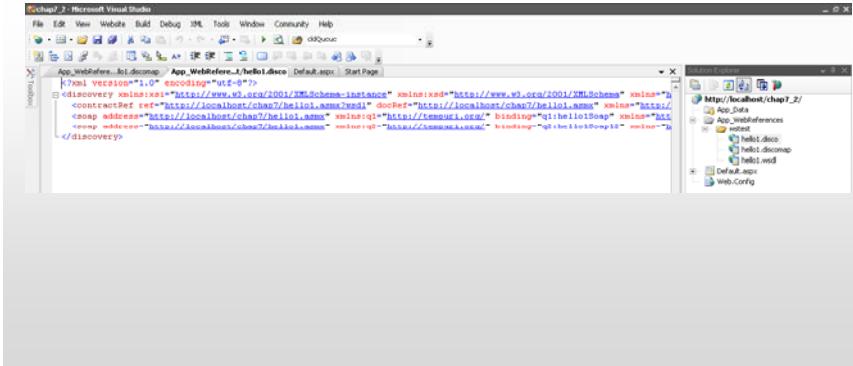
# การใช้ Services

- ให้นักศึกษาสร้าง Web Project ขึ้นมาใหม่ โดยปิด Project เดิมก่อน  
แบบไฟล์ \*.asmx ยังคงต้อง Run อยู่บน Web Browser
- เมื่อ Web Project พร้อมใช้ ให้ทำการ Add Web Reference  
ซึ่งเป็นการดึง WSDL เข้ามาในระบบงานใหม่

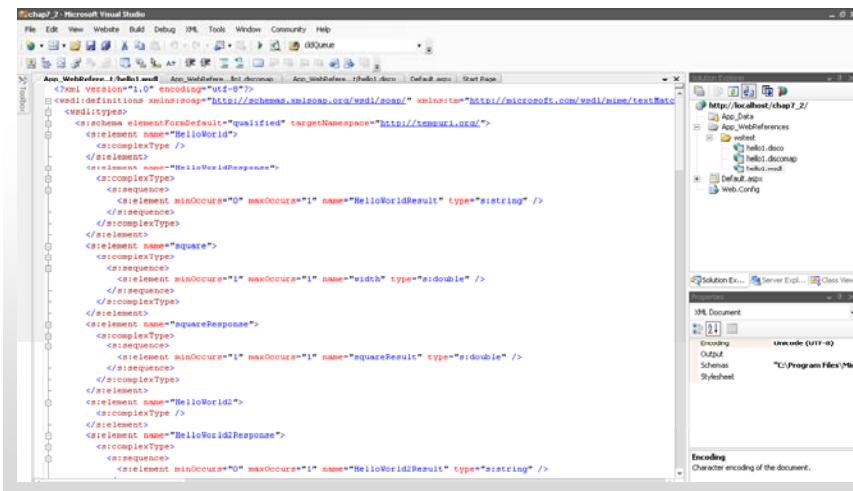
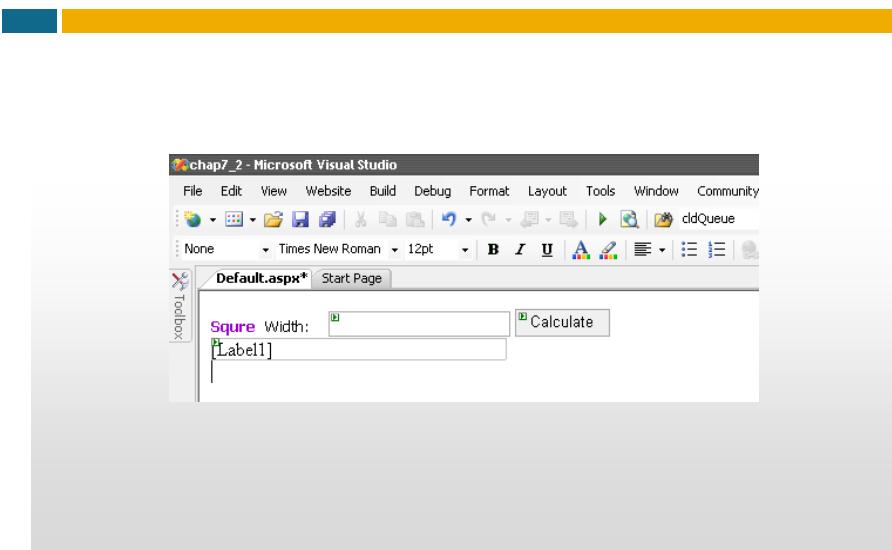


## Disco

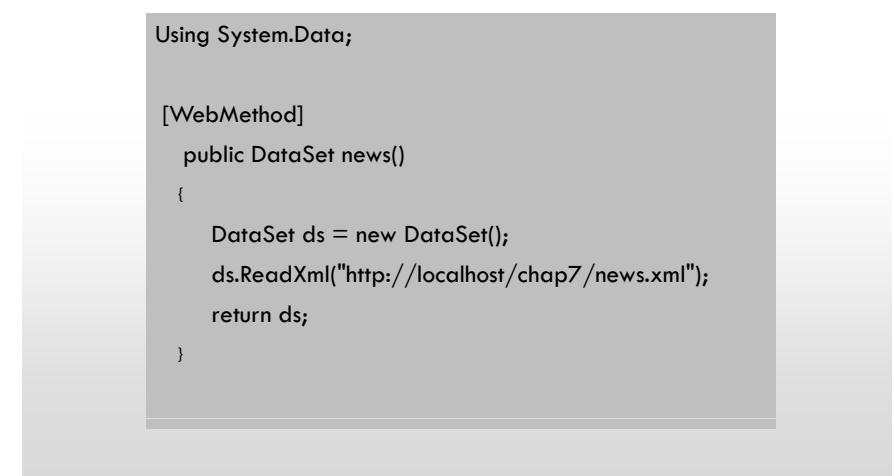
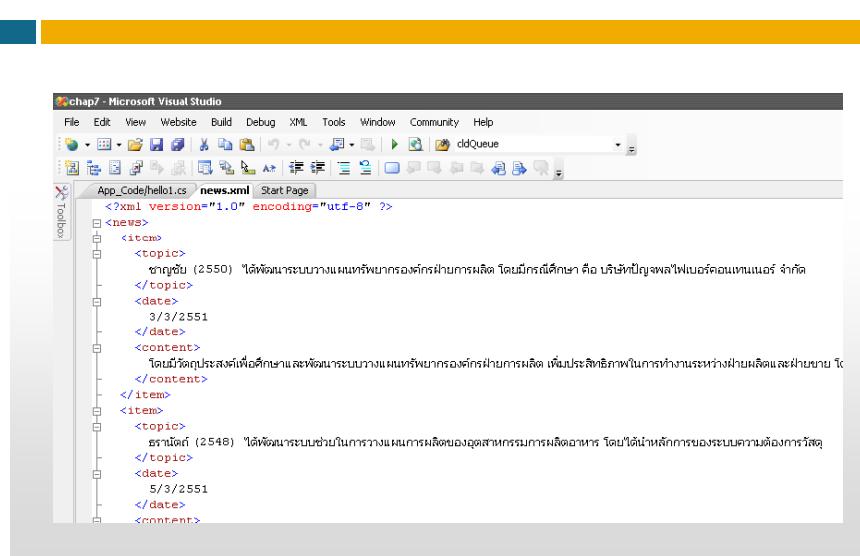
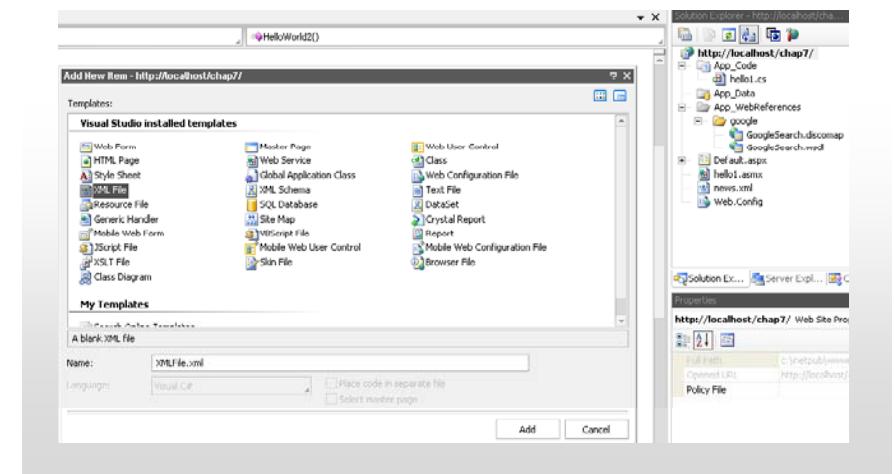
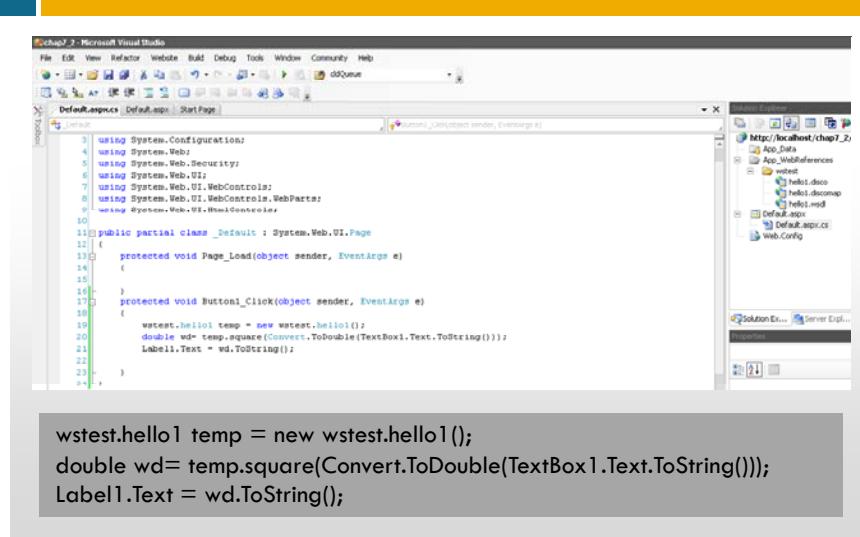
## Disco Map



## WSDL



## ทดลองสร้างไฟล์ XML



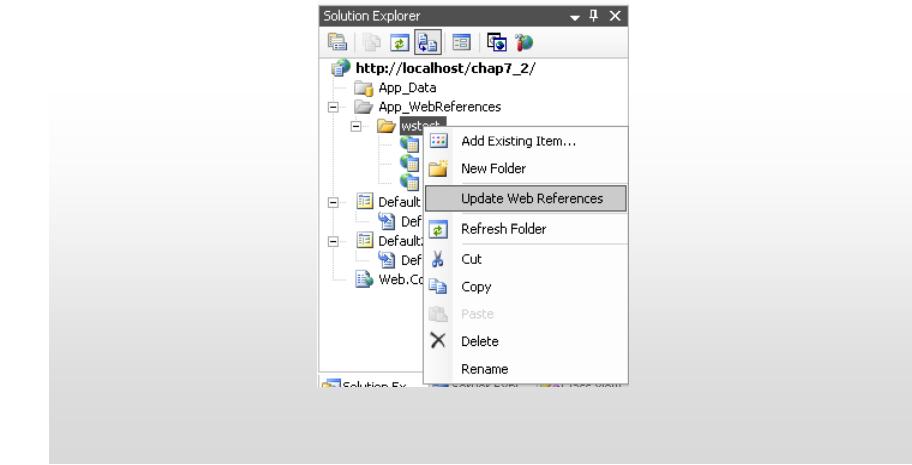
# Update Web References

The screenshot shows a Microsoft Internet Explorer window displaying XML code. The XML structure includes elements like <xs:element>, <xs:complexType>, and <xs:sequence>. A specific news item is highlighted with a yellow background, showing details such as date (3/3/2005), content (text about MySQL and VB.NET), and topic (PHP). The XML code is as follows:

```

<xs:element name="date" type="xs:string" minOccurs="0" />
<xs:element name="content" type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:complexType>
<xs:elements>
</xs:elements>
</xs:schema>
- <dfgr:dfgram smvs:metadata="urn:schemas-microsoft-com:zend-msdata" xmlns:dfgr="urn:schemas-microsoft-com:xml-diffgram2" xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  - <news smvs:>
    - <topic>ภาษา PHP</topic>
    - <date>3/3/2005</date>
    - <content>เรียนรู้การเขียนโปรแกรมภาษา PHP สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา PHP ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา PHP ในการพัฒนาเว็บไซต์ แต่เป็นการเขียนภาษา MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ และค่าที่มาจาก MySQL จะถูกนำมาใช้ใน PHP ได้โดยตรง</content>
    - <date>3/3/2005</date>
    - <item dfgr:id="Item2" metadata:rowOrder="1" dfgr:hasChanges="Inserted">
      <topic>ภาษา VB</topic>
      - <content>เรียนรู้การเขียนภาษา VB สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา VB ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา VB ในการพัฒนาเว็บไซต์ แต่เป็นการเขียนภาษา MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ และค่าที่มาจาก MySQL จะถูกนำมาใช้ใน VB</content>
    - <item dfgr:id="Item3" metadata:rowOrder="2" dfgr:hasChanges="Inserted">
      <topic>ภาษา VB 2005 และ MySQL</topic>
      - <date>4/5/2005</date>
      - <content>เรียนรู้การเขียนภาษา VB 2005 และ MySQL สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา VB 2005 และ MySQL ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา VB 2005 และ MySQL ในการพัฒนาเว็บไซต์และแอปพลิเคชัน แต่เป็นการเขียนภาษา VB 2005 และ MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ</content>
  - </news>
</dfgr:dfgram>

```



The screenshot shows Microsoft Visual Studio with a grid view containing three columns: Column0, Column1, and Column2, each with the value 'abc'. Below the grid, a C# code editor shows the 'Default2.aspx.cs' file. The code defines a partial class 'Default2' that inherits from 'System.Web.UI.Page'. It contains a 'Page\_Load' event handler that creates a new instance of 'wtest.hello1' and binds its data source to a 'GridView1' control.

```

using System.Data;
public partial class Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        wtest.hello1 temp = new wtest.hello1();
        GridView1.DataSource = temp.news();
        GridView1.DataBind();
    }
}

```

The screenshot shows a Microsoft Internet Explorer window displaying the output of the 'Default2.aspx' page. The page lists news items with columns for topic, date, and content. The first news item is about learning PHP and MySQL, the second about VB.NET and MySQL, and the third about VB 2005 and MySQL. The content of the news items is identical to the XML shown in the first screenshot.

topic	date	content
ภาษา PHP (2550) เรียนรู้การเขียนภาษา PHP สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา PHP ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา PHP ในการพัฒนาเว็บไซต์ แต่เป็นการเขียนภาษา MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ และค่าที่มาจาก MySQL จะถูกนำมาใช้ใน PHP	3/3/2005	เรียนรู้การเขียนภาษา PHP สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา PHP ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา PHP ในการพัฒนาเว็บไซต์ แต่เป็นการเขียนภาษา MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ และค่าที่มาจาก MySQL จะถูกนำมาใช้ใน PHP
ภาษา VB (2548) เรียนรู้การเขียนภาษา VB สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา VB ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา VB ในการพัฒนาเว็บไซต์ แต่เป็นการเขียนภาษา MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ และค่าที่มาจาก MySQL จะถูกนำมาใช้ใน VB	4/5/2005	เรียนรู้การเขียนภาษา VB สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา VB ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา VB ในการพัฒนาเว็บไซต์ แต่เป็นการเขียนภาษา MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ และค่าที่มาจาก MySQL จะถูกนำมาใช้ใน VB
ภาษา VB 2005 และ MySQL (2550) เรียนรู้การเขียนภาษา VB 2005 และ MySQL สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา VB 2005 และ MySQL ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา VB 2005 และ MySQL ในการพัฒนาเว็บไซต์และแอปพลิเคชัน แต่เป็นการเขียนภาษา VB 2005 และ MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ</td>	4/5/2005	เรียนรู้การเขียนภาษา VB 2005 และ MySQL สำหรับผู้เริ่มต้น ที่สามารถใช้ภาษา VB 2005 และ MySQL ในการพัฒนาเว็บไซต์และแอปพลิเคชัน ไม่ใช่แค่การเขียนภาษา VB 2005 และ MySQL ในการพัฒนาเว็บไซต์และแอปพลิเคชัน แต่เป็นการเขียนภาษา VB 2005 และ MySQL เพื่อจัดการข้อมูล 1 และ 2 แบบ</td>

# ทดลองดึง RSS จากแหล่งบนอินเทอร์เน็ต

การเมือง

<http://www.innnews.co.th/RSS/politic.xml>

เศรษฐกิจ

<http://www.innnews.co.th/RSS/biz.xml>

กีฬา

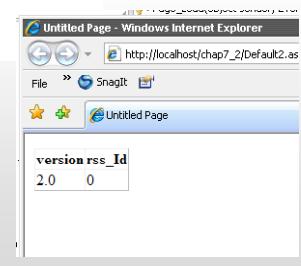
<http://www.innnews.co.th/RSS/sport.xml>

บันเทิง

<http://www.innnews.co.th/RSS/entertain.xml>

```
34 [WebMethod]
35 public string HelloWorld2()
36 {
37     return HelloWorld();
38 }
39 [WebMethod]
40 public DataSet news()
41 {
42     DataSet ds = new DataSet();
43     ds.ReadXml("http://localhost/chap7/news.xml");
44     return ds;
45 }
46 [WebMethod]
47 public DataSet newssport()
48 {
49     DataSet ds = new DataSet();
50     XmlTextReader x = new XmlTextReader("http://www.innnews.co.th/RSS/politic.xml");
51     ds.ReadXml(x);
52     return ds;
53 }
54 }
```

```
13
14 public partial class Default2 : System.Web.UI.Page
15 {
16     protected void Page_Load(object sender, EventArgs e)
17     {
18         wstest.hello1 temp = new wstest.hello1();
19         GridView1.DataSource = temp.newssport();
20         GridView1.DataBind();
21     }
22 }
23 }
```



```
13
14 public partial class Default2 : System.Web.UI.Page
15 {
16     protected void Page_Load(object sender, EventArgs e)
17     {
18         wstest.hello1 temp = new wstest.hello1();
19         DataSet ds = new DataSet();
20         ds = temp.newssport();
21         GridView1.DataSource = ds.Tables[3];
22         GridView1.DataBind();
23 }
24 }
```



## ทดสอบการใช้ Services บนอินเทอร์เน็ต

□ [http://thai-uddi.net/uddi/index.jsp?content=stat\\_rec\\_most.jsp](http://thai-uddi.net/uddi/index.jsp?content=stat_rec_most.jsp)

The screenshot shows a search interface for services. The search criteria are set to 'Between' two dates (YYYY-MM-DD) and 'Rating from 1.0 To 5.0'. The results table lists various services with their descriptions, total votes, and average rating. The services listed include check, check, CmmService, PowerUnit, RIASEcondaryStruct, WebSearchWS, ProfileAPService, GetEntry, RoadDistance, CurrencyConvotor, ValidateEmail, MFundsService, GlobalWeather, GlobalWeather, GlobalWeather, CurrencyConvotor, UltraML, and CurrencyConvotor.

Services name	Description	Total Votes	Rating
check	This	2	5
check	This	1	5
CmmService		1	5
PowerUnit		1	5
RIASEcondaryStruct		1	5
WebSearchWS	asdfasdfoiajedifjaewoi j.o	1	5
ProfileAPService	FTFUTJU	3	4.67
GetEntry	hjhbnjknbkjbjbjk	5	4.6
RoadDistance		4	4.5
CurrencyConvotor		2	4.5
ValidateEmail	View a service	1	4
MFundsService		3	3.33
GlobalWeather		7	3.29
GlobalWeather		5	3
GlobalWeather		2	3
CurrencyConvotor	Currency Convotor	3	2.67
UltraML	WebX	1	2
CurrencyConvotor		1	1

The screenshot shows a login page for the Thailand Public Web Services Registry. It displays binding template information with the following details:

- Binding key : BE4D9570-241D-11DD-BFC2-82113BB23CE3
- Binding description :
- Service key : BE499DD0-241D-11DD-BFC2-A317A47C135C
- Access point : <http://www.webservicex.com/ValidateEmail.asmx>
- tModel Instance(s) : uid:BE4223C0-241D-11DD-BFC2-AD05780DB322

The screenshot shows the 'Add Web Reference' dialog in Visual Studio. The URL is set to <http://www.webservicex.com/ValidateEmail.asmx>. The 'ValidateEmail' service is found at this URL. The 'Web reference name:' field is set to 'email' and the 'Add Reference' button is visible.

```

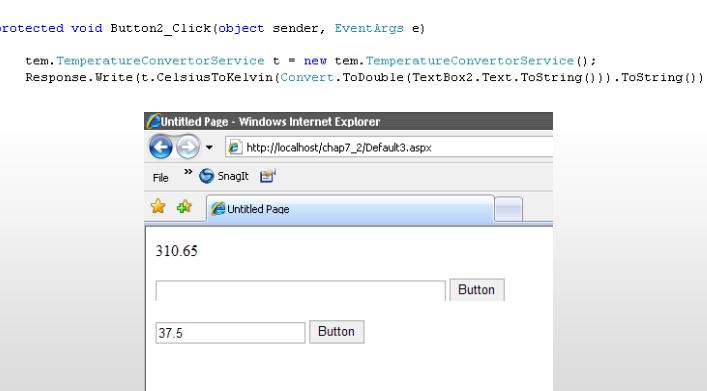
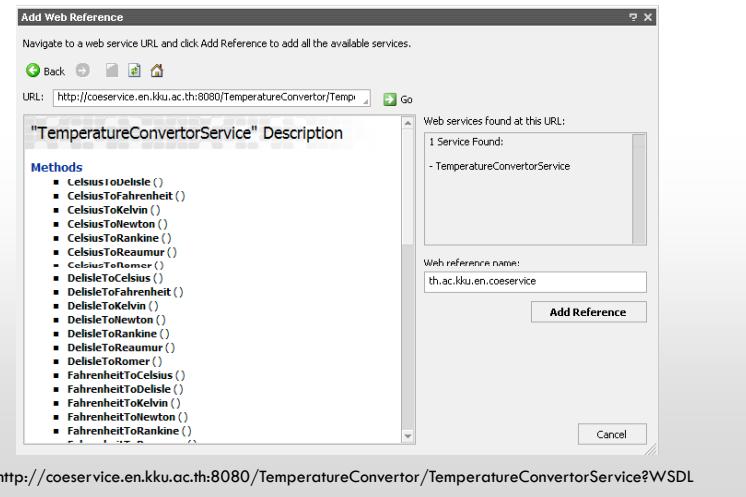
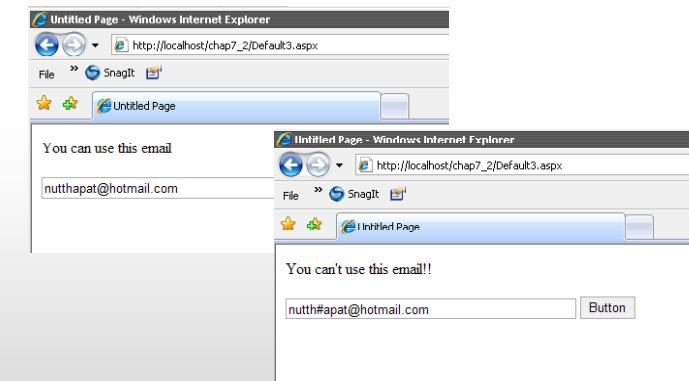
39 [WebMethod]
40 public DataSet news()
41 {
42     DataSet ds = new DataSet();
43     ds.ReadXml("http://localhost/chap7/news.xml");
44     return ds;
45 }
46 [WebMethod]
47 public DataSet newsport()
48 {
49     DataSet ds = new DataSet();
50     XmlTextReader x = new XmlTextReader("http://www.innnncwo.co.th/RSS/politic.xml");
51     ds.ReadXml(x);
52     return ds;
53 }
54 [WebMethod]
55 public bool checkemail(string str)
56 {
57     email.ValidateEmail temp = new email.ValidateEmail();
58     bool a= temp.IsValidEmail(str.ToString());
59     return a;
60 }

```

```

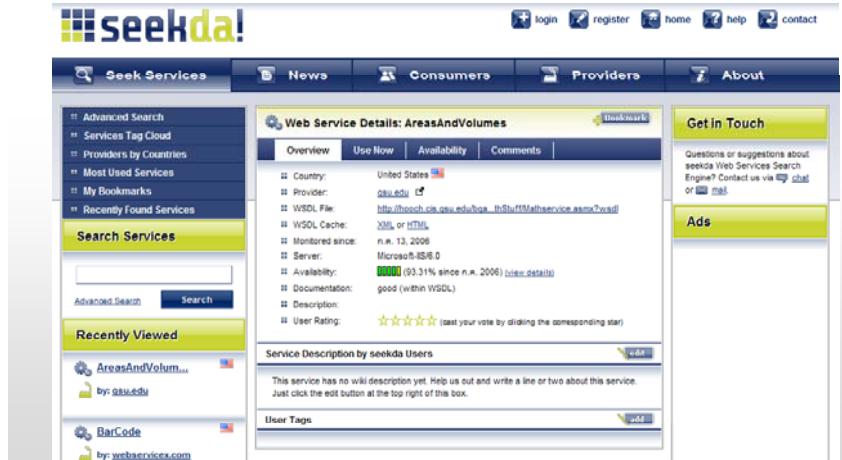
12 public partial class Default3 : System.Web.UI.Page
13 {
14     protected void Page_Load(object sender, EventArgs e)
15     {
16     }
17     protected void Button1_Click(object sender, EventArgs e)
18     {
19         wtest.hello1 temp = new wtest.hello1();
20         bool a = temp.checkemail(TextBox1.Text.ToString());
21         if (a)
22         {
23             Response.Write("You can use this email");
24         }
25         else
26         {
27             Response.Write("You can't use this email!!!");
28         }
29     }
30 }
31
32

```



## Test WebServices From UDDI

- <http://seekda.com/>
- <http://seekda.com/goto?uri=http%3A%2F%2Fhome.cis.gsu.edu%2Fbgates%2FMathStuff%2FMathservice.asmx>
- GOOGLE :  
<http://api.google.com/GoogleSearch.wsdl>
- ແກລກາຫາ
- <http://www.webservicex.com/TranslateService.asmx?WSDL>



The screenshot shows the seekda! website interface. At the top, there's a navigation bar with links for login, register, home, help, and contact. The main content area has a sidebar on the left with links for Advanced Search, Services Tag Cloud, Providers by Countries, Most Used Services, My Bookmarks, and Recently Found Services. Below this is a search bar labeled 'Search Services' with 'Advanced Search' and 'Search' buttons. To the right of the sidebar is a detailed service listing for 'AreasAndVolumes'. The listing includes:

- Country: United States
- Provider: gsu.edu
- WSDL File: http://home.cis.gsu.edu/bga...\_hStuff/Mathservice.asmx?wsdl (XML or HTML)
- Monitored since: n/a, 13. 2006
- Server: Microsoft-IIS/6.0
- Availability: 93.31% since n/a, 2006 (view details)
- Documentation: good (within WSDL)
- User Rating: 4.5 (based on 10 votes)

A 'Recently Viewed' section lists other services: 'AreasAndVolume...' by gsu.edu and 'Barcode' by webservicex.com. On the right side, there's a 'Get in Touch' box for user feedback and an 'Ads' section.

## เทคโนโลยีเว็บเซอร์วิส

ในการพัฒนาระบบสธรายงานและบุคลากรผ่านเทคโนโลยีเว็บเซอร์วิสนี้ได้ทำการศึกษาทฤษฎี และการประยุกต์ใช้เทคโนโลยีเว็บเซอร์วิสที่เกี่ยวข้องกับการพัฒนาระบบ ซึ่งมีรายละเอียดดังต่อไปนี้

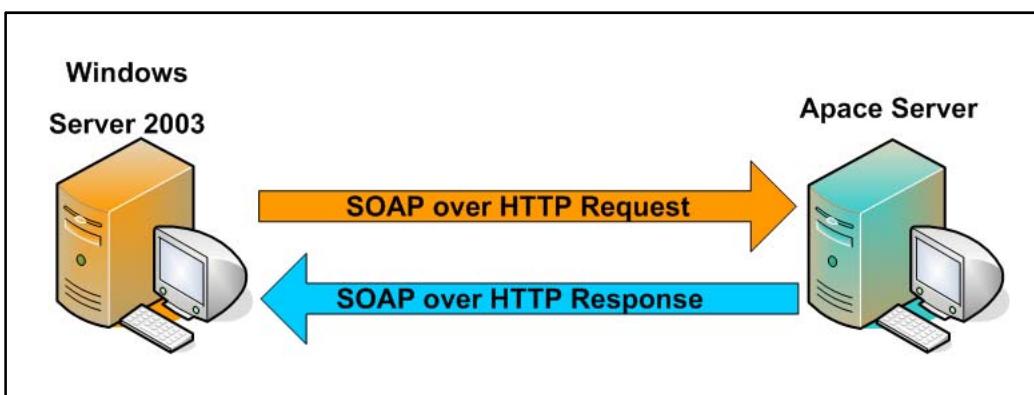
เครือข่ายอินเทอร์เน็ต (Internet) คือ เครือข่ายที่เชื่อมโยงคอมพิวเตอร์ เครื่องแม่ข่าย (Server) และอุปกรณ์ประมวลผลต่างๆ เข้าไว้ด้วยกัน โดยมีการใช้โปรโตคอลที่ซีพีไอพี (Transmission Control Protocol/Internet Protocol: TCP/IP) เป็นมาตรฐานที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ การใช้งานอินเทอร์เน็ตในช่วงแรกเป็นเพียงการรับและส่งไฟล์ข้อมูล หรือ อีเมลล์ (e-Mail) เท่านั้น อินเทอร์เน็ตได้รับการพัฒนาต่อทำให้มีบริการเวิลด์เว็บ (World Wide Web: WWW) เกิดขึ้น เว็บไซต์ต่างๆ จำนวนมากถูกสร้างขึ้นและให้บริการข้อมูลตัวอักษร (Text) ภาพ (Image) และเสียง (Sound) โดยผ่านโปรโตคอลเอชทีพี (Hyper Text Transfer Protocol: HTTP) ที่ทำงานบนโปรโตคอล TCP/IP ตามพอร์ตที่กำหนด (ปกติใช้พอร์ต 80)

การพัฒนาอินเทอร์เน็ตมีการดำเนินต่อไป จากเว็บเพจที่แสดงภาพและตัวอักษรธรรมชาติ ได้มีการพัฒนาให้สามารถประมวลผลด้านระบบฐานข้อมูลได้ ซึ่งทำให้เว็บไซต์ธรรมชาติ (Static Web Site) กลายเป็นเว็บแอพพลิเคชัน (Web Application) ที่สามารถแสดงข้อมูลได้แบบพลวัต (Dynamic Web Site) ทำให้เกิดการพัฒนาเว็บแอพพลิเคชันจำนวนมาก เช่น มีการทำพาณิชย์อิเล็กทรอนิกส์ (e-Commerce) เป็นต้น อุปกรณ์ต่างๆ มีการปรับตัวโดยพัฒนาให้ทำงานร่วมกับเครือข่ายอินเทอร์เน็ตได้ เช่น โทรศัพท์เคลื่อนที่ เครื่องพ็อกเก็ตพีซี (Pocket PC) และอุปกรณ์ต่างๆ จำนวนมาก

เว็บแอพพลิเคชันในปัจจุบันถูกพัฒนาขึ้นจำนวนมาก โดยที่แต่ละเว็บไซต์มีการใช้ระบบฐานข้อมูลของตนเอง แต่เมื่อความต้องการแลกเปลี่ยนข้อมูลเกิดขึ้น ปัญหาการส่งถ่ายข้อมูลจากเว็บไซต์หนึ่งไปยังอีกเว็บไซต์หนึ่งไม่สามารถทำได้สะดวกรวดเร็ว เช่น เว็บไซต์ท่องเที่ยวต้องการข้อมูลเกี่ยวกับราคา และเวลาการเดินทางของสายการบิน ในอดีตต้องใช้พนักงานในการเข้าไปยังเว็บไซต์สายการบินและทำการคัดลอกข้อมูลตารางเวลาและราค่าโดยสาร เพื่อนำมาจัดเก็บในเว็บไซต์ของบริษัทหรืออาจจะใช้วิธีการที่เรียกว่าสกรีนแสคริปปิ้ง (Screen Scraping) ซึ่งเป็นวิธีในการคัดลอก (Copy) ข้อมูลจากหน้าเว็บเพจที่ต้องการ และทำการโปรแกรมให้ตัดป้ายระบุ (Tag) ของภาษา HTML ออกไป เพื่อเหลือส่วนของข้อมูลที่ต้องการ แต่วิธีการเหล่านี้เสียเวลา และเกิดความเสี่ยงที่อาจจะเกิดขึ้นจากมนุษย์ได้ (ธาริน, 2537) ดังนั้นองค์กรทางด้านเทคโนโลยีคอมพิวเตอร์ จึงมีการออกแบบและพัฒนาระบบสำหรับการแลกเปลี่ยนและประมวลผลข้อมูลบนเครือข่ายอินเทอร์เน็ตขึ้น โดยเรียกว่า เว็บเซอร์วิส (Web Services) ทำให้สามารถแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์ส่วนบุคคล พ็อกเก็ตพีซีหรือ โทรศัพท์เคลื่อนที่ได้อย่างสะดวกรวดเร็ว

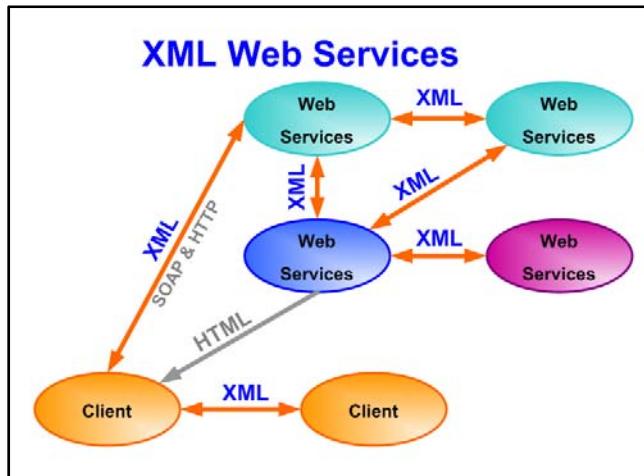
2.2.1 ประวัติความเป็นมาของเว็บเซอร์วิสเริ่มจากเว็บเซอร์วิสที่กำหนดเป็นมาตรฐานโดยโอเออีสไอเอส (Organization for The Advancement of Structured Information Standards: OASIS) และดับเบิลยูทีซี (World Wide Web Consortium: W3C) โดยมีหลักการทำงานเป็นแอพพลิเคชันออนไลน์สำหรับธุรกิจและผู้บริโภค (Consumer) ที่ต้องการประมวลผลและแลกเปลี่ยนข้อมูลผ่านเครือข่าย

อินเทอร์เน็ต โดยผู้ใช้งานสามารถเลือกใช้และผสมผสานเว็บเซอร์วิสเข้าด้วยกันผ่านอุปกรณ์ได้เกือบทุกชนิดตั้งแต่คอมพิวเตอร์ส่วนบุคคล (Personal Computer: PC) จนไปถึงโทรศัพท์เคลื่อนที่ที่รองรับการทำงานเว็บเซอร์วิส เนื่องจากคุณสมบัติที่สำคัญของเว็บเซอร์วิส คือ แอพพลิเคชันที่เป็นเว็บเซอร์วิส สามารถทำงานในมาตรฐานระบบ ภาษาโปรแกรมที่ใช้สำหรับพัฒนา หรือ รูปแบบของระบบ (Platform) ของตนเอง แต่สามารถ “พูดคุย” หรือ ส่งข้อความ (Message) กับแอพพลิเคชันหรือเว็บเซอร์วิสอื่นๆ โดยไม่ยึดติดกับมาตรฐานและเทคโนโลยีของแต่ละฝ่าย (Platform Independence) ทั้งนี้เพื่อประโยชน์ในการแลกเปลี่ยนข้อมูลและบริการร่วมกันโดยไม่ต้องอาศัยมนุษย์เพื่อแปลความหมายของการสนทนานั้น รวมถึงการขอใช้ฟังก์ชันโปรแกรมต่างๆ เรียกว่า ไกเซ่นนี้ว่า อาร์พีซี (Remote Procedure Call: RPC) ทำให้เครื่องคอมพิวเตอร์สามารถสนทนาทำการแลกเปลี่ยนในเชิงของข้อมูลและการประมวลผลได้ (Machine to Machine) โดยสามารถแสดงการร้องขอรับการ (Request) และการตอบสนอง (Response) ข้ามรูปแบบของระบบ (Platform) ดังภาพที่ 2-1



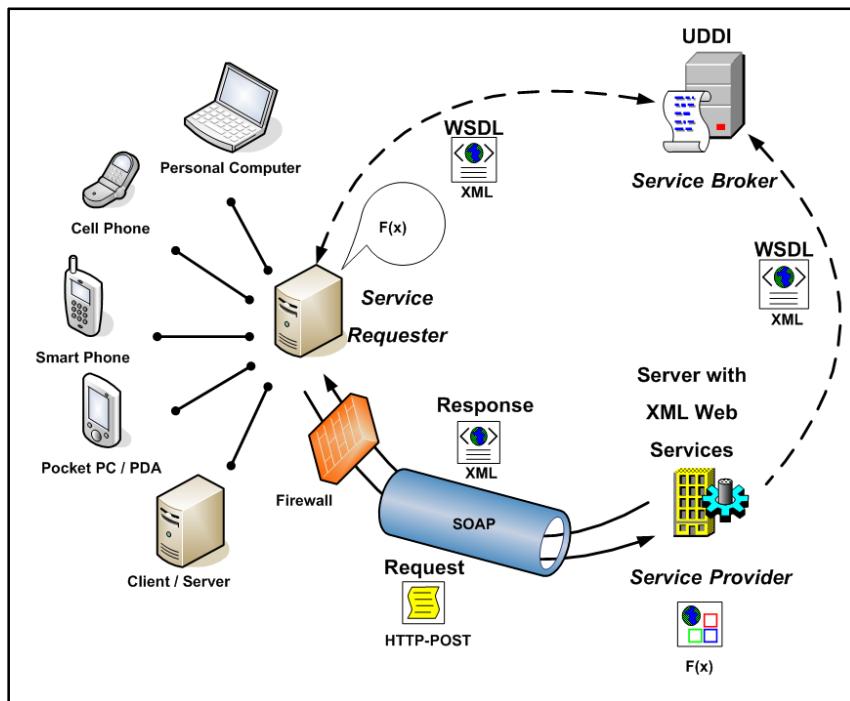
ภาพที่ 2-1 การทำงานข้ามรูปแบบของระบบโดยใช้ SOAP บนโปรโตคอล HTTP

หลักการทำงานเว็บเซอร์วิสทำให้เห็นถึงแนวโน้มในการช่วยให้คอมพิวเตอร์สนทนากับอีกฝ่ายได้ง่ายขึ้น การสนทนา กันระหว่างเครื่องคอมพิวเตอร์จะเกิดขึ้นผ่านมาตรฐานใหม่ของอินเทอร์เน็ตและโปรโตคอลที่ยอมให้แอพพลิเคชันทำงานร่วมกันได้ โดยภาษาสำหรับสนทนาที่ถูกเลือกให้เป็นตัวกลาง เชื่อมโยงระหว่างเครื่องต่อเครื่องนี้ คือ ภาษาเอ็กซ์เพล็กซ์ (eXtensible Markup Language: XML) โดยภาษาเอ็กซ์เพล็กซ์มีการแยกข้อมูลออกจากการแสดงผลทำให้ข้อมูลเป็นอิสระต่ออุปกรณ์หรือภาษาที่ใช้ในการแสดงผล ซึ่งภาษาเอ็กซ์เพล็กซ์จะมีป้ายระบุ (Tag) ที่จัดเก็บเนื้อหาดิจิตอลตามรูปแบบมาตรฐาน ซึ่งเมื่อคอมพิวเตอร์ถูกเชื่อมโยงกันด้วยวิธีการที่เป็นมาตรฐานเดียวกันแล้ว ทำให้สามารถใช้ภาษา มาตรฐานนี้แลกเปลี่ยนเพื่อประมวลผลข้อมูลกันได้สะดวก ธุรกิจหรือองค์กรจะสามารถเชื่อมต่อการทำงานกับพันธมิตรได้อย่างรวดเร็ว อัตโนมัติ และประหยัดค่าใช้จ่ายขององค์กรได้ดังภาพที่ 2-2



ภาพที่ 2-2 การแลกเปลี่ยนข้อมูลด้วยภาษาเอ็กซ์เพล็อกซ์ (XML)

ในการทำงานของแอพพลิเคชันทางด้านเว็บเซอร์วิสจะต้องมีการอธิบายเซอร์วิสหรือ พังก์ชันที่ให้บริการด้วยภาษาเอ็กซ์เพล็อกซ์ โดยจะเขียนไว้ในดับเบลยูดีสเดสก์เพล็อก (Web Service Description Language: WSDL) ทำให้สามารถเข้าใจการใช้เว็บเซอร์วิสนั้นได้ เว็บเซอร์วิสจะใช้ภาษาเอ็กซ์เพล็อกซ์และต่อสื่อสารบนโครงสร้างพื้นฐานของโปรโตคอล HTTP ที่มีอยู่แล้ว โดยการรวมตัวกันระหว่างภาษาเอ็กซ์เพล็อกซ์ กับ โปรโตคอล HTTP มีชื่อเรียกว่าเอสโอลปี หรือ โซฟ (Simple Object Access Protocol: SOAP) และเมื่อมีการพัฒนาเว็บเซอร์วิสสำหรับใช้งานแล้วจะต้องมีการเผยแพร่เว็บเซอร์วิส เพื่อให้พันธมิตรหรือผู้ใช้งานสามารถเข้ามาขอใช้บริการได้ โดยการเผยแพร่นั้นมีการกระทำผ่านสองมาตรฐาน คือ ยูดีดีไอ (Universal Description Discovery and Integration: UDDI) และ ดิสโคฟเวอรี่ (Discovery: DISCO) โดยสามารถแสดงภาพรวมการทำงานของเทคโนโลยีเว็บเซอร์วิสดังภาพที่ 2-3



ภาพที่ 2-3 การทำงานของเทคโนโลยีเว็บเซอร์วิสกับหลักหลายอุปกรณ์

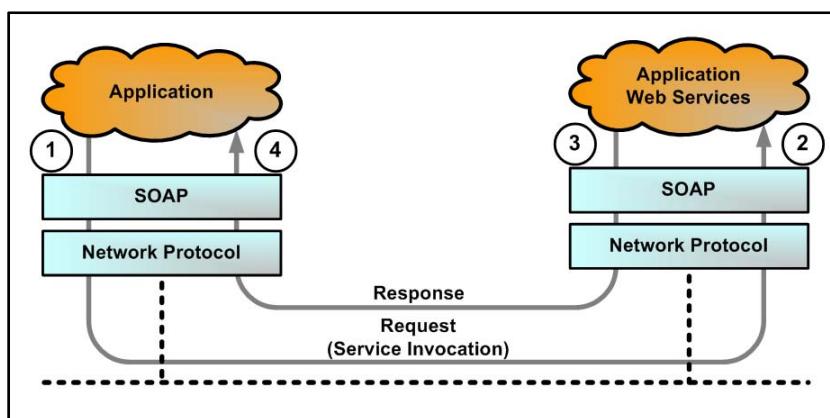
จากภาพที่ 2-3 แสดงให้เห็นถึงหลักการทำงานโดยรวมของเทคโนโลยีเว็บเซอร์วิส โดยจะสังเกตได้ว่าส่วนของผู้ร้องขอใช้บริการ (Service Requester) จะมาจากอุปกรณ์ใดก็ได้ที่สนับสนุนการทำงานของเทคโนโลยีเว็บเซอร์วิส ส่วนผู้ร้องขอใช้บริการจะทำหน้าที่ร้องขอการทำงานหรือฟังก์ชันจากเว็บเซอร์วิสที่สนใจ ซึ่งผู้ร้องขอใช้บริการจะตรวจสอบเว็บเซอร์วิสที่ลงทะเบียนไว้กับนายหน้าผู้ให้บริการ (Service Broker) หรือ UDDI เมื่อพบบริการที่ต้องการแล้วจะตรวจสอบเอกสาร WSDL ที่ใช้สำหรับอธิบายบริการหรือฟังก์ชันทำให้ทราบถึงตัวแปร (Variables) และชนิดของตัวแปร (Type of Variables) เพื่อส่งค่าตัวแปรไปให้เว็บเซอร์วิสทำงานได้อย่างถูกต้องและในส่วนของผู้ตระเตรียมบริการ (Service Provider) เมื่อทำการพัฒนาบริการหรือฟังก์ชันแล้วจะต้องทำการลงทะเบียนเพื่อส่งเอกสาร WSDL ไปยังส่วนของ UDDI เมื่อผู้ร้องขอใช้บริการทำการตรวจสอบเอกสารและการบริการต่างๆ แล้วติดต่อกับผู้ตระเตรียมบริการโดยตรงผ่านโปรโตคอล SOAP ที่ทำงานบนโปรโตคอล HTTP จากนั้นผู้ร้องขอใช้บริการจะส่งค่าตัวแปรไปยังผู้ตระเตรียมบริการเพื่อทำงานตามฟังก์ชันที่ร้องขอ เมื่อทำงานตามฟังก์ชันเสร็จสิ้นจะส่งค่ากลับ (Return) ไปยังผู้ร้องขอใช้บริการโดยข้อมูลที่ใช้ในการส่งถ่ายจะอยู่ในรูปของเอกสารเอกซ์เพล็กซ์ (XML) ได้โดยไม่ติดปัญหาการขวางกั้นข้อมูลบนเครือข่าย

2.2.2 องค์ประกอบหลักที่สำคัญของเว็บเซอร์วิสที่จะสามารถทำงานประสานความสัมพันธ์การให้บริการและการรับบริการ มีดังต่อไปนี้

2.2.2.1 ภาษาเอกซ์เพล็กซ์ (eXtensible Markup Language: XML) คือ ภาษามาตรฐานที่ใช้ในการแลกเปลี่ยนข้อมูลบนเครือข่ายคอมพิวเตอร์ โดยเป็นอิสระจาก

รูปแบบของระบบและส่วนแสดงผล ซึ่งลักษณะโครงสร้างของภาษาเอ็กซ์เพรสส์เป็นป้ายชื่อ (Tag) ที่สามารถนิยามความหมายของได้ทำให้สามารถเข้าถึงและถ่ายเทข้อมูลโดยอาศัย Tag ต่างๆ

2.2.2.2 โปรโตคอลเอสโอลเอฟ (Simple Object Access Protocol: SOAP) หรือ โซฟ คือ โปรโตคอลที่ทำให้สามารถเรียกใช้ส่วนประกอบ (Component) ต่างๆ ที่ทำงานอยู่บนรูปแบบของระบบที่ต่างกันได้ โดยโปรโตคอล SOAP เป็นโปรโตคอลที่ทำงานในระดับชั้นแอพพลิเคชั่น (Application Layer) ผ่านอินเทอร์เน็ตโปรโตคอล เช่น HTTP ที่ใช้เป็นโปรโตคอลในการส่งข้อมูลที่หลากหลาย (Hyper Text) แนวคิดการเรียกใช้เว็บเซอร์วิสผ่านโปรโตคอล SOAP โดยจำแนกเป็นระดับชั้นทางเครือข่ายสามารถแสดงได้ดังภาพที่ 2-4



ภาพที่ 2-4 การร้องขอและการตอบสนองการบริการผ่าน SOAP Message

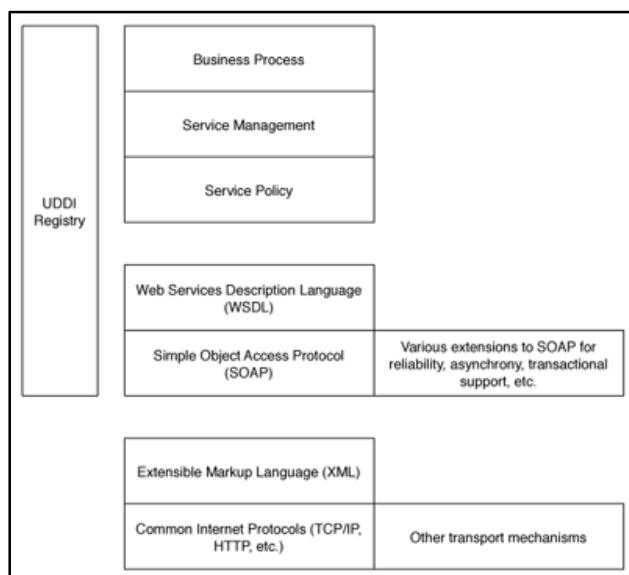
จากภาพ 2-4 สามารถอธิบายขั้นตอนการทำงานของโปรโตคอล SOAP ได้ดังนี้

- แอพพลิเคชั่นของผู้ร้องขอบริการทำการสร้าง SOAP Message เพื่อส่งผ่านสื่อไปใช้บริการจากเว็บเซอร์วิส
- เว็บเซอร์วิสผู้ให้บริการจะได้รับ SOAP Message จากผู้ร้องขอบริการ โดยจะอยู่ในรูปแบบภาษาเอ็กซ์เพรสส์แล้ว
- เว็บเซอร์วิสทำการประมวลผลตามบริการที่ร้องขอ และส่งผลลัพธ์กลับไปโดยจะอยู่ในรูปแบบของ SOAP Message ก่อนการส่งถ่ายกลับไปยังผู้ขอรับบริการ
- แอพพลิเคชั่นของผู้ขอรับบริการจะได้รับ SOAP Message กลับมา และถูกแปลงให้อยู่ในรูปแบบข้อมูลของผลลัพธ์ที่ต้องการเพื่อนำไปใช้งานต่อไป

2.2.2.3 ดับเบิลยูดีแอล (Web Services Description Language: WSDL) เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของเว็บเซอร์วิสและวิธีการติดต่อกับเว็บเซอร์วิส WSDL เป็นสิ่งที่จำเป็นในการพัฒนาระบบกระจาย (Distributed System) เพื่อใช้อธิบายความหมายและวิธีการติดต่อ (Interface Definition Language: IDL) โดยใช้ภาษา XML ซึ่ง WSDL เกิดจากการรวมแนวคิดของเอ็นเออสแอล (The Network Accessible Service Specification Language: NASSL) และดับเบิลยูดีแอล (Well-Defined Services: WDS) ของบริษัทไอบีเอ็ม กับ เอสดีแอล (The Service

Description Language: SDL) และเอกสารชีแอล (The SOAP Contract Language: SCL) ของบริษัทไมโครซอฟท์ ปัจจุบัน WSDL เป็นภาษาที่อยู่ในการดูแลของ W3C (World Wide Web Consortium)

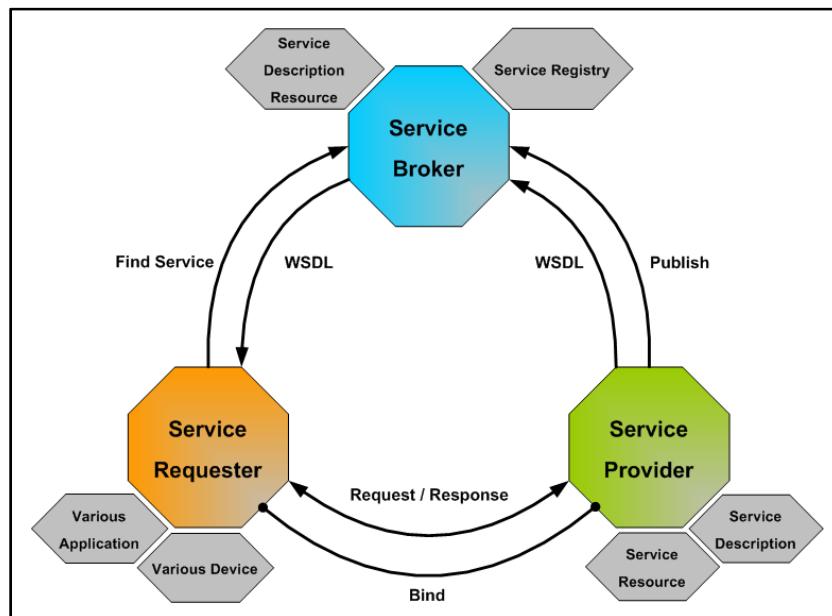
2.2.2.4 ยูดีดีไอ (Universal Description Discovery and Integration: UDDI) เป็นสมุดโทรศัพท์ UDDI ที่ตระเตรียมไว้สำหรับ XML Web Services ทำให้ผู้ที่ต้องการใช้เว็บเซอร์วิสสามารถทำการค้นหาเว็บเซอร์วิสที่ต้องการใช้งานเพื่อสนองต่อความต้องการของระบบที่พัฒนาโดยผู้ที่พัฒนาหรือสร้างเว็บเซอร์วิสขึ้นสามารถที่จะนำเว็บเซอร์วิสของตนเองไปลงทะเบียนไว้กับ UDDI เนื่องจาก UDDI เป็นตัวกลางในการเชื่อมโยงเว็บเซอร์วิสเข้ากับผู้ที่ต้องการขอใช้บริการ โดยสามารถแสดงลำดับความสัมพันธ์ของ UDDI กับเทคโนโลยีเว็บเซอร์วิสได้ดังภาพที่ 2-5 (UDDI XML.ORG and OASIS, 2007)



ภาพที่ 2-5 ความสัมพันธ์ระหว่าง UDDI กับเทคโนโลยีเว็บเซอร์วิส

2.2.3 สถาปัตยกรรมแบบเอสโอเอ (SOA) กับเทคโนโลยีเว็บเซอร์วิส (Web Services Technology) เว็บเซอร์วิสใช้แนวคิดจากสถาปัตยกรรมการให้บริการแบบเอสโอเอ (Service-Oriented Architecture: SOA) ซึ่งคำว่า Service-Oriented หมายถึง พังก์ชันหรือแอพพลิเคชันที่ทำงานเฉพาะด้านขนาดใหญ่ แต่สมมุติเป็นซอฟต์แวร์ร่ายอยู่ที่รอการประกอบกันเท่านั้นไม่ใช่ซอฟแวร์แพ็คเกจ (Package Software) ที่มีขนาดใหญ่ ส่วนคำว่า Architecture หมายถึง สถาปัตยกรรมการออกแบบระบบขององค์กรโดยแสดงถึงความต้องการบริการที่จะใช้กับระบบขององค์กร และมีการออกแบบให้ออกมาในรูปแบบของพังก์ชันงานย่อยมีความสามารถในการทำงานเฉพาะอย่างและรอการประกอบกันของพังก์ชันเท่านั้น แนวคิดของ SOA ทำให้เกิดการปรับเปลี่ยนแนวทางการพัฒนาระบบท่างๆ ที่ทำงานร่วมกับเครือข่ายอินเทอร์เน็ต เกิดการพัฒนาระบบในแบบโซลูชัน (Software as a Service: SaaS) ซึ่งเป็นการพัฒนาระบบที่ใช้การประกอบกันของบริการผ่านเครือข่ายโดยผู้ที่ใช้บริการไม่จำเป็นต้องพัฒนาบริการขึ้นมาใช้งานเอง แต่สามารถขอใช้บริการหรือขอเช่าใช้งานได้ ดังนั้นแนวคิดของ SOA จึงเป็นส่วนหนึ่งในการประมวลผลแบบกระจาย (Distributed Computing Concepts) ที่มีประโยชน์ในการ

ใช้ทรัพยากรระบบอย่างไม่สิ้นเปลือง ลดความซ้ำซ้อนของข้อมูลและฟังก์ชันงาน รวมถึงเพิ่มประสิทธิภาพการประมวลผลกรณีที่ต้องใช้การประมวลผลสูง โดยสถาปัตยกรรมแบบ SOA สามารถแสดงได้ดังภาพที่ 2-6



ภาพที่ 2-6 สถาปัตยกรรมแบบเอสโอเอ (Service-Oriented Architecture: SOA)

จากภาพที่ 2-6 แสดงให้เห็นถึงสถาปัตยกรรมแบบเอสโอเอที่เป็นแนวคิดหลักของเทคโนโลยีเว็บเซอร์วิส โดยจะประกอบด้วยกัน 3 ส่วน สามารถอธิบายหลักการและความสัมพันธ์ของแต่ละส่วนได้ดังนี้

2.2.3.1 ผู้ร้องขอใช้บริการ (Service Requester) หมายถึง ผู้ใช้ (User) อุปกรณ์ (Various Device) หรือ แอพพลิเคชัน (Application) ที่มีความต้องการใช้บริการใดๆ ที่สามารถตอบสนองต่อความต้องการของผู้ร้องขอได้จากผู้ตระเตรียมบริการ (Service Provider) ซึ่งผู้ร้องขอใช้บริการจะทำการค้นหา (Find) บริการได้จากนายหน้าผู้ให้บริการ (Service Broker) และจะได้รับเอกสาร WSDL เพื่อแสดงรายละเอียดที่จะใช้บริการนั้นๆ ทำให้ผู้ร้องขอใช้บริการส่งค่าพารามิเตอร์ (Parameters) ไปยังบริการของผู้ตระเตรียมบริการ และทำการรอรับผลลัพธ์ที่สนองต่อการร้องขอใช้บริการกลับคืนมา

2.2.3.2 นายหน้าผู้ให้บริการ (Service Broker) หมายถึง แหล่งที่บันทึกข้อมูลการลงทะเบียนบริการ (Service Registry) โดยนายหน้าผู้ให้บริการแทนด้วย UDDI ในเทคโนโลยีเว็บเซอร์วิส ซึ่งมีไว้สำหรับผู้ตระเตรียมบริการที่ต้องการเผยแพร่ (Publish) บริการการทำงานของตน เพื่อเป็นประโยชน์ต่อผู้พัฒนาระบบที่มีความต้องการใช้บริการนั้นๆ โดยนายหน้าผู้ให้บริการจะมีการเก็บข้อมูลเอกสาร WSDL ให้แก่ผู้ร้องขอใช้บริการเพื่อแสดงรายละเอียดของบริการนั้นๆ และเมื่อผู้ร้องขอใช้บริการทราบข้อมูลจากเอกสาร WSDL และจะทำการผูก (Bind) กับผู้ตระเตรียมบริการโดยตรง ซึ่งไม่จำเป็นต้องติดตอกับนายหน้าผู้ให้บริการอีก ในการผูกเพื่อบริการนี้จะมีการใช้ SOAP เป็นโปรโตคอลใน

เทคโนโลยีเว็บเซอร์วิสและลักษณะการทำงานจะมีการร้องขอ (Request) ไปยังบริการ เมื่อได้ผลลัพธ์จะทำการสนองกลับ (Response) มายังผู้ขอใช้บริการ

2.2.3.3 ผู้ตระเตรียมบริการ (Service Provider) หมายถึง ผู้ที่พัฒนาและสร้างบริการต่างๆ สำหรับใช้พัฒนาระบบงานของตนเองและผู้อื่น โดยมีแหล่งเก็บบริการและเอกสาร WSDL ของตนเอง ซึ่งผู้ตระเตรียมบริการจะมีการเผยแพร่บริการโดยการลงทะเบียนและใช้อเอกสาร WSDL ฝ่ากไว้กับนายหน้าผู้ให้บริการ เพื่อให้ผู้ร้องขอใช้บริการสืบค้นบริการต่างๆ ที่ต้องการได้

## ภาษาเอ็กซ์เอ็มแอล

ในปัจจุบันการแสดงผลของเว็บไซต์ทั่วไปจะใช้ภาษาเอชทีเอ็มแอล (Hypertext Markup Language: HTML) โดยภาษา HTML ได้รับการพัฒนาโดย Tim Bernes-Lee ในปี ค.ศ. 1991 ซึ่งเป็นภาษาที่ได้รับความนิยมในการพัฒนาส่วนการแสดงผลของเว็บเพจอย่างกว้างขวางและเป็นภาษามาตรฐานหนึ่งของเอสจีเอ็มแอล (Standard Generalized Markup Language: SGML, ISO 8879) โดย SGML เป็นมาตรฐานที่ทำเครื่องหมายไว้ในเอกสารเป็นโครงสร้างสัญลักษณ์ เช่น ป้ายชื่อ (Tag) ในภาษา HTML เป็นต้น (Jon Bosak, 1997) ภาษา HTML ถูกควบคุมจัดการโดยดับเบิลยูทรีซี (World Wide Web Consortium: W3C) ในปัจจุบัน W3C ผลักดันรูปแบบของภาษา HTML แบบใหม่ที่เรียกว่า เอ็กซ์เอชทีเอ็มแอล (eXtensible HTML: XHTML) ซึ่งเป็นลักษณะของโครงสร้าง XML แบบหนึ่งที่มีหลักเกณฑ์ในการกำหนดโครงสร้างของโปรแกรมที่มีรูปแบบที่เป็นมาตรฐานกว่ามาตรฐานเดิมที่ใช้ HTML รุ่น 4.01 ที่ใช้กันอยู่ในปัจจุบัน (วิกิพีเดียสารานุกรมเสรี, 2550)

ภาษา HTML เป็นภาษามาตรฐานที่ใช้ในการแสดงผลและจัดโครงสร้างหลักหน้าเว็บเพจ เช่น ส่วนหัวของเว็บเพจ (Header) ส่วนของตัวเว็บเพจ (Body) เท่านั้น โดยสามารถจัดวางข้อมูลได้หลากหลาย (Hypertext) เช่น ข้อความ (Text) รูปภาพ (Image) รวมถึงไฟล์มัลติมีเดีย (Multimedia) อื่นๆ ที่ภาษา HTML รองรับการแสดงผลได้

### 2.3.1 คุณลักษณะของภาษา HTML สามารถสรุปได้ดังนี้

2.3.1.1 ภาษา HTML เป็นภาษาที่เน้นการแสดงผลบนเว็บเบราว์เซอร์ (Web Browser) เป็นหลัก

2.3.1.2 ภาษา HTML ไม่สามารถแสดงผลบนหน้าจอในอุปกรณ์อื่นๆ ได้อย่างสะดวก เช่น โทรศัพท์เคลื่อนที่และ PDA เป็นต้น แต่การใช้ภาษา XHTML ที่พัฒนาขึ้นใหม่มีการใช้หลักการของภาษาเอ็กซ์เอ็มแอลทำให้สามารถแสดงผลได้

2.3.1.3 ภาษา HTML ไม่สามารถส่งถ่ายเพื่อแปลงไฟล์ได้อย่างสะดวก เพราะความไม่ยืดหยุ่นในตัวภาษา เช่น ใช้โปรแกรมไมโครซอฟท์เวิร์ดสร้างไฟล์ HTML แต่กรณีที่แปลงย้อนกลับไปเป็นไฟล์เอกสารเดิมที่ใช้สร้างไม่สามารถทำได้เหมือนต้นฉบับ

2.3.1.4 ข้อมูลที่อยู่ในป้ายชื่อ (Tag) ภาษา HTML ไม่สามารถส่งถ่ายข้อมูลให้กับแอพพลิเคชันอื่นๆ ได้ เนื่องจากภาษา HTML เป็นภาษาที่ใช้แสดงผลเท่านั้น กรณีที่จะทำได้จะต้องทำการสำเนาไฟล์ HTML และใช้ฟังก์ชันพิเศษในการตัดป้ายชื่อภาษา HTML ออกไป แต่การทำเช่นนี้อาจจะเกิดความบกพร่องของข้อมูลที่ได้รับ เพราะไม่สามารถเข้าใจป้ายชื่อต้นทางที่เก็บข้อมูลกับความต้องการบางข้อมูลที่อยู่ในบางป้ายชื่อได้ทำให้ขาดความเชื่อถือ

2.3.1.5 ภาษา HTML เป็นภาษาที่ไม่มีความเป็นอัตโนมัติ เนื่องจากการทำงานในลักษณะการแสดงผลเพียงอย่างเดียวไม่สามารถทำงานร่วมกับโปรแกรมอื่นๆ ได้อย่างสะดวก ไม่มีการนิยามความหมายของแต่ละป้ายชื่อได้อย่างยึดหยุ่น การนำไปใช้จึงมีความลำบากไม่สะดวกและรวดเร็ว

จากคุณลักษณะของภาษา HTML ทำให้ทราบถึงความไม่ยึดหยุ่นของภาษา ข้อมูลที่อยู่ในป้ายชื่อ (Tag) ของภาษา HTML จะไม่สามารถถ่ายได้อย่างสะดวกรวดเร็ว รวมถึงการใช้ข้อมูลภายใต้ป้ายชื่อจะไม่สามารถทำได้อย่างถูกต้อง ทั้งนี้เนื่องจากมาตรฐานของภาษา HTML ใช้สำหรับแสดงผลเพียงอย่างเดียวเท่านั้น

ดังนั้นจึงมีการพัฒนาภาษามาตรฐานภาษาหนึ่งขึ้นมาเพื่อแก้ไขความไม่ยึดหยุ่นของการส่งถ่ายข้อมูลที่ยึดติดกับส่วนการแสดงผลโดยในปี พ.ศ. 2539 John Bosak ได้เสนอแนวคิดต่อ W3C โดยนำเสนอภาษาใหม่โดยยังคงรักษาโครงสร้างของเอสจีเอ็มแอล (Standard Generalized Markup Language: SGML, ISO 8879) แต่มีการปรับปรุงให้แตกต่างไป เรียกว่ามาตรฐานใหม่นี้ว่า ภาษาเอ็กซ์เอ็มแอล (eXtensible Markup Language: XML) ซึ่งใช้ในการบรรจุข้อมูลเท่านั้นจะไม่มีส่วนของการแสดงผลเข้ามาในเอกสาร XML นี้ แต่จะมีป้ายชื่อเพื่อใช้แยกข้อมูลในลักษณะของความหมาย ทำให้สามารถนำเอกสารอิเล็กทรอนิกส์ไปใช้กับแอพพลิเคชันอื่นได้ เนื่องจากแอพพลิเคชันที่นำเอกสาร XML ไปใช้สามารถถอดข้อมูลออกจากเอกสารโดยเข้าถึงจากป้ายชื่อของข้อมูลซึ่งจะพบข้อมูล แต่การที่จะเข้าถึงป้ายชื่อนั้นจำเป็นที่จะต้องทราบความหมายของแต่ละป้ายชื่อเพื่อเลือกใช้ข้อมูลได้อย่างถูกต้อง และภาษา XML สามารถให้ทำการปรับปรุงกำหนดป้ายชื่อได้เองเพื่อให้ความหมายที่ยึดหยุ่นของผู้พัฒนา

```
<?xml version="1.0" e
<quiz>
  <question>
    Who was the forty-second
    president of the U.S.A.?
  </question>
  <answer>
    William Jefferson Clinton
  </answer>
  <!-- Note: We need to add
       more questions later.-->
</quiz>
```

**XML**

## ภาพที่ 2-7 ตัวอย่างเอกสารเอ็กซ์เพ็มแอล (XML Document)

2.3.2 ความหมายของภาษาเอ็กซ์เพ็มแอล (XML) มีผู้เชี่ยวชาญด้านคอมพิวเตอร์และเครือข่ายได้ให้ความหมายของภาษาเอ็กซ์เพ็มแอลไว้หลายท่าน โดยผู้วิจัยได้นำความหมายของภาษาเอ็กซ์เพ็มแอลจาก Richard Baldwin ซึ่งได้อธิบายความหมายของเอ็กซ์เพ็มแอลไว้ดังนี้ ภาษาเอ็กซ์เพ็มแอลทำให้ผู้ใช้สามารถสร้างและดูแลเอกสารได้ โดยภาษาเอ็กซ์เพ็มแอลเป็นภาษาที่มีโครงสร้างเอกสารที่สร้างโดยภาษาเอ็กซ์เพ็มแอลจึงมีโครงสร้าง (Structured Documents) ภายในเอกสารจะบรรจุตัวอักษร (Plain Text) ซึ่งทำให้สามารถปรับเปลี่ยนการแสดงผล (Rendering) ในรูปแบบที่หลากหลาย จุดประสงค์หลักของเอ็กซ์เพ็มแอล คือ การแยกส่วนข้อมูลเพื่อประโยชน์ในการแสดงผลและการนำข้อมูลไปใช้ต่อ (เกียรติศักดิ์, 2549) จากการอธิบายความหมายของ Richard Baldwin ทำให้เกิดคำศัพท์ที่สำคัญดังนี้

2.3.2.1 เอกสารที่มีโครงสร้าง (Structured Documents) สามารถจำแนกได้เป็นสองประเภทดังนี้ (Text, 2007)

ก) โครงสร้างที่มองเห็น (Visible Structure) คือ เอกสารที่มีข้อมูลต่างๆ เช่น อักษร รูปภาพ ตาราง ลายเส้น เป็นต้น โดยภายในเอกสารที่มีโครงสร้างที่มองเห็นจะเป็นข้อมูลนั้นๆ อยู่ในสภาพแสดงให้เห็นจริงตามขนาด ทิศทาง และสี ซึ่งเอกสารลักษณะนี้จะเหมาะสมสำหรับการทำความเข้าใจของมนุษย์ได้ง่ายไม่ต้องมีการถอดข้อมูลนั้นก่อนการแสดงผล

ข) โครงสร้างที่มองไม่เห็น (Invisible Structure) คือ เอกสารที่มีข้อมูลในรูปแบบอิเล็กทรอนิกส์ โดยได้มีการทำเครื่องหมายต่างๆ (Electronic Marker) ไว้กับข้อมูล เพื่อใช้ในการแสดงผลบนอุปกรณ์ หรือ ใช้ในการทำความเข้าใจของเครื่องจักรและเครื่องคำนวณต่างๆ โดยเอกสารลักษณะนี้จะไม่เห็นโครงสร้างของข้อมูลโดยตรงแต่มีการใช้เครื่องหมายหรืออักษรบางอย่างที่จะบอกถึงความหมาย ความสัมพันธ์ หรือรูปแบบที่จะใช้ในการแสดงผลข้อมูลนั้นๆ ดังนั้นการแสดงผลให้ผู้ใช้ที่เป็นมนุษย์เข้าใจหรือรับทราบข้อมูล จะต้องมีการแปลหรือประมวลผลเอกสารนั้นให้ออกมาก่อนเสมอ เพื่อให้ข้อมูลอยู่ในรูปแบบที่มีโครงสร้างที่มองเห็นได้ เช่น โครงสร้างตารางของระบบฐานข้อมูล และ โครงสร้างแบบ SGML/XML Document Type Definitions เป็นต้น

ดังนั้นความหมายของ Richard Baldwin ที่กล่าวถึงเอกสารที่มีโครงสร้าง (Structured Documents) ก็ควรจะหมายถึงเอกสารแบบโครงสร้างที่มองไม่เห็น ซึ่งหากพิจารณาเอกสารเอ็กซ์เพ็มแอลแล้วการที่เป็นเอกสารแบบโครงสร้างที่มองไม่เห็นจะมีการใส่ป้ายชื่อ (Tag) กับข้อมูลเพื่อเป็นการนิยามข้อมูลนั้นไม่ใช้การแสดงผลโดยตรงและการใส่ป้ายชื่อ (Tag) ภายในเอกสารเอ็กซ์เพ็มแอลนั้น จะต้องยึดตามหลักไวยกรรมและโครงสร้างของภาษาด้วย

2.3.2.2 ตัวอักษรพื้นฐาน (Plain Text) คือ อักษรอักษรพื้นฐานที่สามารถพิมพ์ได้ (Alphabet and Punctuation) โดยในปัจจุบันจะเป็นอักษรอักษรที่เข้ารหัสแบบแอลกิฟ (American Standards Committee on Information Interchange: ASCII) ซึ่งเป็นมาตรฐานที่ควบคุมลักษณะของ

อักษร อักษร ดังนั้นเอกสารที่สร้างจากภาษาเอ็กซ์เพลจิ้งสามารถใช้กับระบบต่างๆ ได้โดยไม่ขึ้นอยู่ กับระบบหรือรูปแบบของระบบใดๆ

**2.3.2.3 การปรับเปลี่ยนแสดงผล (Rendering)** คือ การแสดงผลข้อมูลต่างๆ ให้อยู่ในรูปแบบที่มนุษย์สามารถเข้าใจได้ โดยข้อมูลที่นำมาแสดงนั้นจะถูกแปลมาจากข้อมูลดิบที่อยู่ในรูปแบบของเครื่องจักรหรือเครื่องคอมพิวเตอร์เข้าใจ ถ้าประมวลจากการแปลแล้วมนุษย์จะไม่สามารถที่จะเห็นภาพหรือข้อมูลในลักษณะธรรมชาติได้เลย แต่จะเห็นข้อมูลดิบที่ยากจะเข้าใจเมื่อเอกสารเอ็กซ์เพลจิ้งเป็นเอกสารแบบโครงสร้างที่มองไม่เห็นและใช้อักษรพื้นฐานภายในเอกสารทำให้จะต้องมีการปรับเปลี่ยนการแสดงผลได้ โดยจะขึ้นอยู่กับการพัฒนาโปรแกรมหรือแอพพลิเคชันที่จะนำไปใช้ในการนำไปใช้สามารถที่จะมีการปรับปรุงรูปแบบ (Format) ของข้อมูลได้ เช่น การทำตัวหนาหรือการปรับปรุงสีให้แก่ตัวอักษรและข้อมูล รวมถึงการสร้างรูปแบบกราฟฟิกจากข้อมูลดิบได้ด้วย ตัวอย่างเช่น ภาษา HTML ที่มีความสามารถด้านการแสดงผล ดังนั้นเอกสารภาษาเอ็กซ์เพลจิ้งเป็นภาษาที่หมายความที่ใช้เป็นเสียงภาษากลางในการติดต่อและถ่ายเทข้อมูลให้แก่ระบบคอมพิวเตอร์ได้เป็นอย่างดี

### 2.3.3 คุณลักษณะของภาษาเอ็กซ์เพลจิ้ง สามารถสรุปได้ดังนี้

**2.3.3.1 ภาษาเอ็กซ์เพลจิ้งและสามารถสร้างเอกสารอิเล็กทรอนิกส์ที่ภายในบรรจุข้อมูลต่างๆ โดยมีการใช้ป้ายชื่อเชิงความหมาย (Meta Data) เพื่อแยกข้อมูล**

**2.3.3.2 ภาษาเอ็กซ์เพลจิ้งและออกแบบให้มีการใช้งานที่สะดวกสามารถกำหนดป้ายชื่อได้เองทำให้ทราบถึงความหมายเฉพาะของแต่ละป้ายชื่อทำให้มนุษย์สามารถที่จะเข้าใจได้เมื่อยกกรณีที่ใช้ภาษาธรรมชาติ การที่เข้าใจป้ายชื่อข้อมูลไม่ยากทำให้การพัฒนาระบบสามารถเข้าถึงและดึงข้อมูลที่ตอบสนองต่อความต้องการอกรมาได้มีความสะดวกและไม่ผิดพลาดหรือบกพร่อง**

**2.3.3.3 ภาษาเอ็กซ์เพลจิ้งและเป็นภาษามาตรฐานที่อยู่ในระดับต่ำ (Low Level) สามารถนำไปใช้ร่วมกับการพัฒนาหรือใช้ร่วมกับโปรแกรมหรือแอพพลิเคชันอื่นๆ ได้**

**2.3.3.4 เมื่อภาษาเอ็กซ์เพลจิ้งและอยู่ในระดับต่ำทำให้สามารถสร้างเอกสาร XML ได้จากตัวแก้ไขอักษรทั่วไป (Text Editor) และในปัจจุบันมีการพัฒนาโปรแกรมสำหรับช่วยสร้างเอกสารนี้ให้สะดวกรวดเร็วมากขึ้น**

**2.3.3.5 ภาษาเอ็กซ์เพลจิ้งและจะบรรจุข้อมูลโดยไม่มีส่วนของการแสดงผลภายในเอกสาร ทำให้โปรแกรมหรือแอพพลิเคชันสำหรับแสดงผลสามารถดึงข้อมูลไปใช้ได้อย่างสะดวก**

**2.3.3.6 ภาษาเอ็กซ์เพลจิ้งและเป็นภาษาที่มีโครงสร้าง (Structured Documents) ทำให้สะดวกต่อการทำความเข้าใจและสามารถนำไปใช้ประยุกต์และพัฒนาระบบงานต่างๆ ได้อย่างหลักหลา**

### 2.3.4 โครงสร้างหลักของภาษาเอ็กซ์เพลจิ้งและจำแนกออกเป็น 3 ส่วนดังนี้

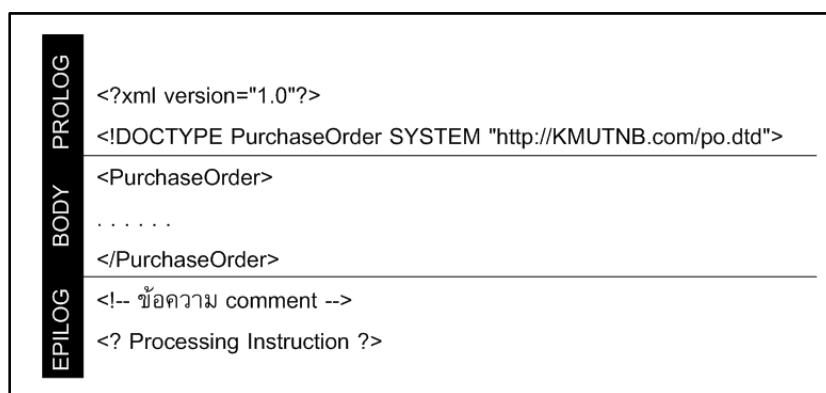
**2.3.4.1 โปรดล็อก (Prolog) คือ ส่วนหัวในการประกาศ (Declaration) ซึ่งจะแยกได้เป็น 2 ส่วนคือ ส่วนการประกาศรุ่น (Version) ของภาษาเอ็กซ์เพลจิ้งเพื่อการแปลผลที่ถูกต้องตามรุ่นของภาษา นี้ และส่วนที่สองคือ ส่วนของการประกาศกฎเกี่ยวกับเอกสารเอ็กซ์เพลจิ้งและนั้นๆ (Document**

Type Declaration) ซึ่งสามารถใส่ก្នុងរាយในเอกสารรวมกับภาษาเอ็กซ์เอ็มแอลได้เลย หรือจะทำการดึงไฟล์นิยามชนิดเอกสาร (Document Type Definition: DTD) เข้ามาใช้ได้เช่นกัน

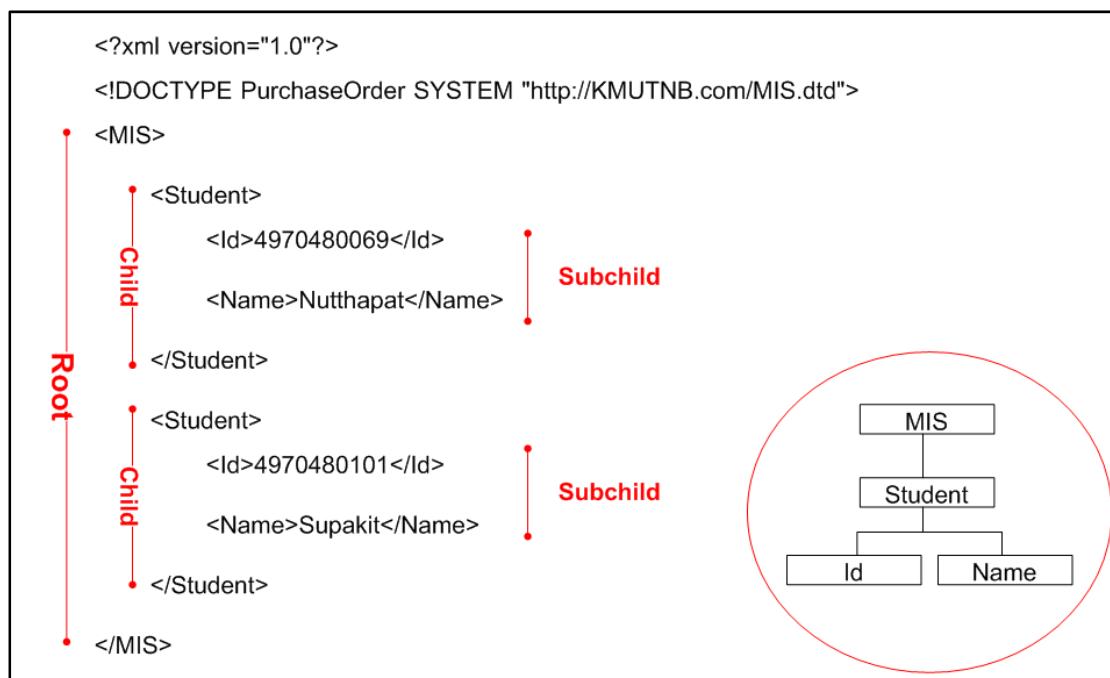
2.3.4.2 บอดี้ (Body) คือ ส่วนของเนื้อหาที่เป็นข้อมูลจริงและป้ายชื่อ (Tag) โดยในส่วนนี้จะบรรจุข้อมูลที่ถูกป้ายชื่อ (Tag) ครอบไว้เพื่อให้นิยามแก่ข้อมูลนั้นๆ ซึ่งมีการใช้โครงสร้างเอกสารตามหลักการและไวยกรณ์ของภาษาเอ็กซ์เอ็มแอล

2.3.4.3 อิพิล็อก (Epilog) คือ ส่วนเพิ่มเติมที่ทำหน้าที่เป็นข้อความวิจารณ์หรือเป็นข้อความคำอธิบายชนิดหมายเหตุ (Comments) และส่วนที่เป็นโปรดเซสชิงอินสตรัคชัน (Processing Instruction: PI) ส่วนนี้ไม่จำเป็นต้องอยู่บริเวณล่างสุดของเอกสาร แต่สามารถแทรกอยู่ในส่วนบอดี้ได้

จากคำอธิบายโครงสร้างหลักของเอกสารที่สร้างโดยภาษาเอ็กซ์เอ็มแอลทั้ง 3 ส่วนสามารถแสดงได้ดังภาพที่ 2-8 และ 2-9



ภาพที่ 2-8 โครงสร้างหลักของเอกสารภาษาเอ็กซ์เอ็มแอล



## ภาพที่ 2-9 ตัวอย่างป้ายชื่อและขอบเขตข้อมูลด้วยภาษาเอ็กซ์เอ็มแอล

2.3.5 รูปแบบและข้อกำหนดในการสร้างเอกสารข้อมูลด้วยภาษาเอ็กซ์เอ็มแอลจะมีหลักไว้ยกรณ์ทางภาษาคอมพิวเตอร์ ดังนั้นจำเป็นต้องทราบเกี่ยวกับกฎข้อบังคับในการสร้าง โดยสามารถสรุปได้เป็นหัวข้อดังนี้

2.3.5.1 ภาษาเอ็กซ์เอ็มแอลจะอยู่ในรูปแบบของป้ายชื่อ (Tag) ขั้นข้อมูลจริง โดยป้ายชื่อจะอยู่ในรูปแบบ “`<tag> ... </tag>`” สามารถแสดงได้ดังภาพที่ 2-9

2.3.5.2 เมื่อสร้างป้ายชื่อ (Tag) แล้ว ป้ายชื่อดังกล่าวจะต้องมีทั้งป้ายชื่อเปิด และป้ายชื่อปิดเสมอเพื่อเป็นการระบุขอบเขตของป้ายชื่อนั้นๆ กับข้อมูลที่ต้องการนิยาม

2.3.5.3 ในไวยกรณ์ภาษาเอ็กซ์เอ็มแอลจะถือว่าตัวอักษรเล็กกับตัวอักษรใหญ่มีความแตกต่างกัน (Case Sensitive)

2.3.5.4 การใส่ป้ายชื่อจะต้องใส่ให้ถูกต้องตามลำดับขั้นของข้อมูล เนื่องจากภาษาเอ็กซ์เอ็มแอลสามารถสร้างลำดับขั้นของข้อมูลเพื่อแสดงถึงความสัมพันธ์ของข้อมูลได้ ดังนั้นการเปิดป้ายชื่อและปิดป้ายชื่อจึงมีความสำคัญกับนิยามและขอบเขตของข้อมูล

2.3.6 หมวดหมู่ที่เกี่ยวกับเอ็กซ์เอ็มแอล (XML Categories) ภาษาเอ็กซ์เอ็มแอลได้ถูกพัฒนาขึ้นเพื่อใช้นิยามข้อมูลและเป็นตัวกลางในการแลกเปลี่ยนเท่านั้น แต่ภาษาเอ็กซ์เอ็มแอลยังมีเทคโนโลยีอื่นสนับสนุนการทำงานดังนี้ (Mark, 2003)

2.3.6.1 เอ็กซ์เอสแอล (eXtensible Stylesheet Language: XSL) เป็นภาษาที่พัฒนาขึ้นเพื่อทำหน้าที่ควบคุมการแสดงผลให้กับข้อมูลที่ใช้ภาษาเอ็กซ์เอ็มแอลโดยตรง และ XSL สามารถใช้งานร่วมกับภาษา HTML และ CSS ได้เป็นอย่างดี โดยภาษา XSL ประกอบไปด้วยส่วนประกอบ (Elements) เช่นเดียวกับภาษา HTML แต่มีความแตกต่างกัน เช่น ภาษา XSL สามารถทำงานแบบมีเงื่อนไขได้ เป็นต้น

2.3.6.2 ดีเอสโอ (Data Source Object: DSO) เป็นเทคโนโลยีที่ถูกพัฒนาจากบริษัทไมโครซอฟท์ซึ่งเป็นแหล่งข้อมูลเสมือนฐานข้อมูลขนาดเล็ก ภาษา HTML จะมองข้อมูลจากเอ็กซ์เอ็มแอลเป็นกลุ่มของข้อมูล สามารถใช้ภาษา HTML ร่วมกับเอ็กซ์เอ็มแอลเพื่อผูกกันทำให้สามารถใช้ป้ายชื่อของภาษา HTML แสดงผลข้อมูลที่ป้ายชื่อนั้นๆ ได้

2.3.6.3 ดีโอม (Document Object Model: DOM) คือ แบบจำลองของเอกสารบนเว็บ เพื่อให้แต่ละภาษาที่ใช้พัฒนาสามารถนำข้อมูลที่ใช้เทคโนโลยีเอ็กซ์เอ็มแอลไปใช้ได้ โดยการเข้าถึงหรือการประมวลผลข้อมูลที่อยู่ในเอ็กซ์เอ็มแอลสามารถทำได้โดยไวยกรณ์ของภาษาหนึ่ง หรือวัตถุ (Object) ที่สามารถทำงานกับข้อมูลในเอ็กซ์เอ็มแอลได้ ซึ่งการทำงานรูปแบบนี้เป็นการทำงานกับข้อมูลข้ามรูปแบบของระบบ