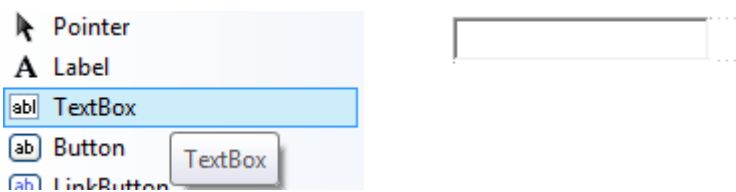


บทที่ 1 เทคนิคในการเขียนโปรแกรมกับ TextBox



บทนำ

TextBox เป็นเครื่องมือพื้นฐานในการป้อนข้อมูลตัวอักษรในการเขียนโปรแกรมแบบ GUI (Graphic User Interface) ไม่ว่าจะเป็นการเขียนโปรแกรมบนวินโดวส์ เว็บ มือถือ หรือ แอปพลิเคชันบนแพลตฟอร์มอื่นที่แสดงผลด้วยกราฟฟิก ในบทนี้จะนำเสนอเทคนิคการเขียนโปรแกรมกับ TextBox ให้มีคุณสมบัติในการตรวจสอบค่าตัวแปรประเภทต่างๆ อย่างง่ายโดยไม่ต้องไปซื้อหาคอนโทรลมาใช้ เช่น การป้อนค่าตัวเลขจำนวนจริง แบบมีทศนิยม เบอร์โทรศัพท์ อีเมล เว็บไซต์ วันที่ เป็นต้น โดยใช้จาวาสคริปต์ หรือ Ajax คอนโทรลช่วยในการเขียน รวมถึงการประยุกต์เขียนเป็นคอนโทรลนำไปใช้เองได้ทุกๆ โปรเจ็ค

Required :	Required!	<input type="text"/>
Integer :	Please input value!	<input type="text"/>
Double :		<input type="text"/>
Zip Code :		<input type="text"/>
Phone Number :		<input type="text"/>
Email :		<input type="text"/>
Web Url :		<input type="text"/>
dd/MM/yyyy :		<input type="text"/>
MM/dd/yyyy :		<input type="text"/>
yyyy/MM/dd :		<input type="text"/>
hh:mm:ss :		<input type="text"/>
dd/MM/yyyy hh:mm:ss :		<input type="text"/>
MM/dd/yyyy hh:mm:ss :		<input type="text"/>
yyyy/MM/dd hh:mm:ss :		<input type="text"/>

1 เทคนิคป้อนตัวเลขจำนวนเต็มแบบไม่มีจุดทศนิยม

1. เขียนจาวาสคริปต์ ดังนี้

```
<script language="javascript" type="text/javascript">
function onlyInteger()
{
    if ((window.event.keyCode < 48) || (window.event.keyCode > 57))
    { return false; }
}
</script>
```

2. ใน Code Behind อีเวนต์ Page_Load ให้กำหนด แอททริบิวต์ให้แก่ TextBox ดังนี้

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Attributes.Add("onkeypress", "return onlyInteger();");
}
```

เมื่อลองรันโปรแกรมจะเห็นได้ว่าจะสามารถพิมพ์ได้เฉพาะตัวเลข 0-9 เท่านั้น

2 เทคนิคป้อนตัวเลขจำนวนเต็มแบบมีจุดทศนิยม

1. เขียนจาวาสคริปต์ ดังนี้

```
function onlyDouble(input)
{
    if (window.event.keyCode == 46) // keyCode 46 is "."
    {
        if (input.value.split('.').length == 1)
        {
            return true;
        }
        else {
            return false;
        }
    }
    else
    if ((window.event.keyCode < 48) || (window.event.keyCode > 57))
    { return false; }
}
```

ลำดับ	คำสั่ง	คำอธิบาย
1	window.event.keyCode	ตรวจสอบรหัสจาก คีย์บอร์ด ซึ่งรหัส 46 คือตัวอักษร '.'
2	input.value.split('.')	เป็นการตัดข้อความไปเก็บไว้ในอาร์เรย์ด้วยตัวอักษรที่เราระบุ ในฟังก์ชันนี้ระบุตัวอักษร '.' หากข้อความใน TextBox ของเรา คือ '123.03' จะได้อาร์เรย์ สองตัว คือ '123' และ '03' ในตัวอย่างมีการตรวจสอบ ถ้าตัดได้ข้อความเดียว

แสดงว่าใน TextBox นั้นยังไม่มีตัวอักษร '.' จึงยอมให้คีย์ตัวอักษรนี้ได้

2. ใน Code Behind อีเวนต์ Page_Load ให้กำหนด แอททริบิวต์ให้แก่ TextBox ดังนี้

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Attributes.Add("onkeypress", "return onlyDouble(this);");
}
```

123.3445|

เมื่อลองรันโปรแกรมจะเห็นได้ว่าจะสามารถพิมพ์ได้เฉพาะตัวเลข 0-9 และจุดทศนิยมหนึ่งตัวเท่านั้น

3 เทคนิคป้อนตัวเลขรหัสไปรษณีย์

จริงๆ แล้วตัวเลขรหัสไปรษณีย์ก็คือเลขจำนวนเต็ม ซึ่งโดยทั่วๆ ไปที่พบเห็นทั่วโลกจะใช้ 5-6 หลัก ไม่เกินนี้ ที่ประเทศไทยก็จะใช้อยู่ 5 หลัก การเขียนก็จะเหมือนๆ กับการป้อนเลขทศนิยมเพียงแต่กำหนดจำนวนตัวอักษรเป็น 5 ตัว

1. เขียนจาวาสคริปต์ ดังนี้

```
function onlyZipCode(input)
{
    if (input.value.length >= 5)
    {
        return false;
    }
    if ((window.event.keyCode < 48) || (window.event.keyCode > 57))
    { return false; }
}
```

2. ใน Code Behind อีเวนต์ Page_Load ให้กำหนด แอททริบิวต์ให้แก่ TextBox ดังนี้

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Attributes.Add("onkeypress", "return onlyZipCode(this);");
}
```

33000|

เมื่อลองรันโปรแกรมจะเห็นได้ว่าจะสามารถพิมพ์ได้เฉพาะตัวเลข 0-9 ได้ 5 ตัวเท่านั้น

* การกำหนดความกว้างของตัวอักษรใน TextBox ไม่จำเป็นต้องเขียนโค้ดตรวจสอบ สามารถกำหนดคุณสมบัติ TextBox.MaxLength ที่ฝั่ง server side ได้

```
<asp:TextBox ID="TextBox1" runat="server" MaxLength="5"></asp:TextBox>
```

4

เทคนิคป้อนเบอร์โทรศัพท์

การป้อนเบอร์โทรศัพท์ จะยอมให้เพิ่มตัวอักษร (,)- ส่วนขนาดจำนวนตัวอักษร รูปแบบทั่วไป ของเบอร์โทรศัพท์คือ (xxx) xxx-xxxx จำนวนตัวเลขคือสิบตัว รวมกับตัวอักษรพิเศษคือ 13 ตัว

1. เขียนจาวาสคริปต์ ดังนี้

```
function PhoneNumber(input)
{
    if (input.value.length >= 14)
    {
        return false;
    }
    else
    if (window.event.keyCode == 40 || window.event.keyCode == 41 ||
        window.event.keyCode == 45) // "(,)-"
    {
        return true;
    }
    else
    if ((window.event.keyCode < 48) || (window.event.keyCode > 57))
    { return false; }
}
```

keyCode ที่เพิ่มขึ้นมา คือ 40=> (41 =>) 45 =>-

2. ใน Code Behind อีเวนต์ Page_Load ให้กำหนด แอททริบิวต์ให้แก่ TextBox ดังนี้

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Attributes.Add("onkeypress", "return PhoneNumber(this);");
}
```

(080)123-4567

เมื่อลองรันโปรแกรมจะเห็นได้ว่าสามารถพิมพ์ได้เฉพาะตัวเลข 0-9 ตัวอักษร '(,)' และ '-' ได้ 13 ตัวเท่านั้น

เพื่อให้ผู้ใช้ทราบรูปแบบในการป้อนข้อมูล สามารถนำเอา MaskEditorExtender ของ AjaxToolkit เข้ามาช่วย

(สามารถดาวน์โหลดได้ที่ <http://ajaxcontroltoolkit.codeplex.com>)

```
<form id="form1" runat="server">
<asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
</asp:ToolkitScriptManager>
<div>
<asp:TextBox ID="TextBox1" runat="server" ></asp:TextBox>
<asp:MaskedEditExtender ID="MaskedEditExtender1" runat="server"
TargetControlID="TextBox1" Mask="(999) 999-9999" MaskType="Number">
</asp:MaskedEditExtender>
</div>
```

A

B

A. สำหรับผู้ที่ไม่เคยใช้ AjaxControlToolkit ของไมโครซอฟท์ การจะใช้คอนโทรลได้ต้องวางคอนโทรล Script Manager ไว้ส่วนบนสุดในฟอร์มเสมอ สำหรับ ASP.NET 4.0 ให้ใช้ ToolkitScriptManager แทน

B. สำหรับคอนโทรล MaskEditorExtender จะวางไว้ที่ใดก็ได้ภายในฟอร์ม เมื่อรันเว็บ จะไม่แสดง

ตารางคุณสมบัติ MaskEditorExtender

ลำดับ	คุณสมบัติ	คำอธิบาย
1	TargetControlID	ระบุ ID ของคอนโทรลที่ต้องการกำหนด Mask ซึ่งจะใช้กับคอนโทรล TextBox เป็นหลัก
2	Mask	กำหนดรูปแบบของ Mask ตามตัวอย่าง (999)999-9999 คือ มีวงเล็บ ตัวเลขสามตัว ชิด และตัวเลขอีกสี่ตัว
3	MaskType	กำหนดรูปแบบ Mask (วันที่, ตัวเลข)

เมื่อรันเว็บจะได้ดังนี้



จะเห็นได้ว่าการบังคับรูปแบบการป้อนอัตโนมัติ

5 เทคนิคป้อนอีเมล

อีเมลมีรูปแบบตัวอักษรที่แน่นอน คือ `xxxxx@xxx.xxx` หรือ `xxxxx@xx.xx` ซึ่งจะเป็นรูปแบบใดนั้นณปัจจุบันก็จะมีชื่อเมลล์ตามด้วยตัวอักษร@ ซึ่งเราสามารถตัดข้อความและตรวจสอบได้ แต่ในตัวอย่างนี้จะใช้คุณสมบัติ expressions ในจาวาสคริปต์

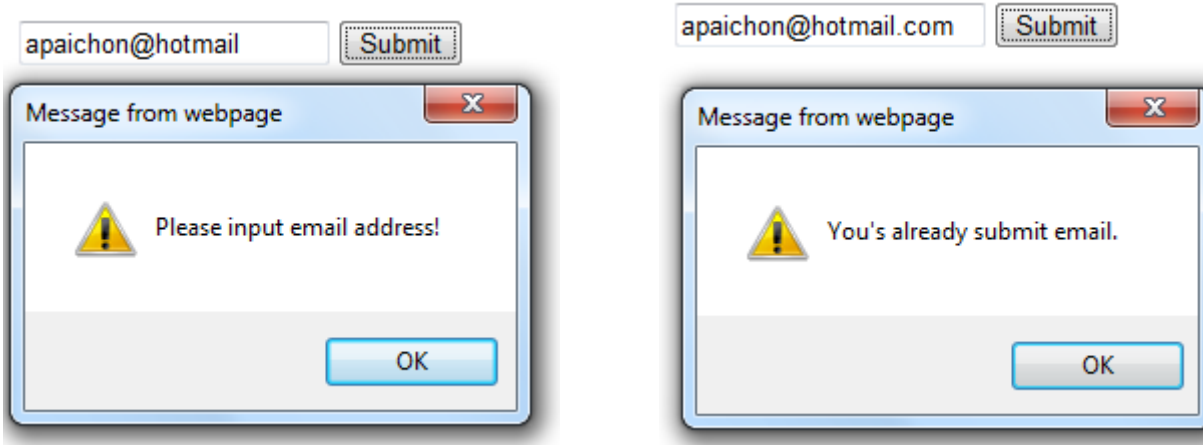
1. เขียนจาวาสคริปต์ ดังนี้

```
function isEmail(strmail)
{
    if (!strmail.match(/\b(^\S+@)((\S+\.com)|(\S+\.net)|(\S+\.edu)|(\S+\.mil)|(\S+\.gov)|(\S+\.org)|(\S+\{2,2\}))\b$/gi))
    {
        alert("Please input email address!");
        return false;
    }
    else
    {
        return true;
    }
}
```

2. เพิ่มปุ่ม Submit เพื่อทำการตรวจสอบหลังส่งยืนยันการตรวจสอบ

3. ป้อนแอตทริบิวต์ ให้กับปุ่ม Submit ใน Code Behind ดังนี้

```
protected void Page_Load(object sender, EventArgs e)
{
    btnSubmit.Attributes.Add("onclick", "return isEmail(document.getElementById('"+TextBox1.ClientID+"').value);");
}
```



หากป้อนผิดรูปแบบจะขึ้นข้อความฟ้อง แต่หากป้อนถูกรูปแบบ ฟังก์ชันจะคืนค่าเป็น true และทำคำสั่งในอีเวนต์ของปุ่ม Submit ที่ฝั่งเซิร์ฟเวอร์ต่อไป

ลำดับ	พารามิเตอร์	คำอธิบาย
1	/	ต้องป้อนเสมอก่อนกำหนดรูปแบบ
2	\b	ตัวอักษรที่ต้องตรงกัน ในตัวอย่างนี้ โค้ดคือ \b ซึ่งไม่ได้ระบุอักษรใด แสดงว่าไม่ได้เจาะจง แต่หากระบุ a\b ภายในข้อความจะต้องมีคำว่า a อยู่ด้วย
3	\s	จะต้องไม่มีช่องว่าง
4	{2,2}	{จำนวนอักษรที่ต้องตรงกัน,จำนวนอักษรต้องไม่เกิน}
5	/gi	ตัวอักษรตัวเล็กตัวใหญ่ถือว่าเป็นตรงกัน เช่น ab = AB

6 เทคนิคป้อน Web URL

การป้อนรูปแบบลิงค์แบบเว็บ url สามารถ นำตัวอย่างก่อนหน้ามาใช้ได้เพียงแต่เปลี่ยนรูปแบบของ expression เพียงเล็กน้อยเท่านั้น

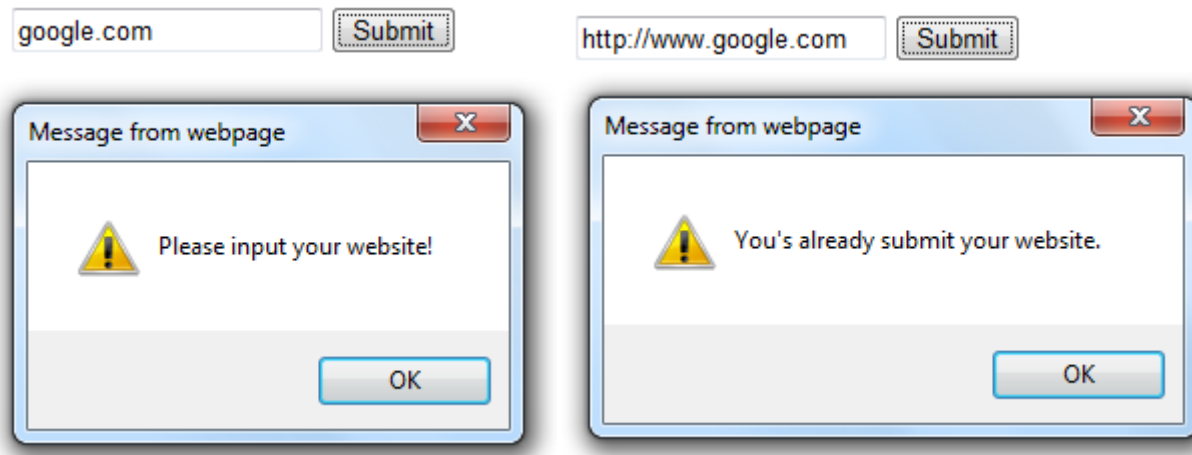
- เขียนจาวาสคริปต์ ดังนี้

```
function isWebUrl(strmail)
{
    if (!strmail.match(/http:\/\/\b[A-Za-z0-9\.-]{3,}\.[A-Za-z]{3}/) )
    {
        alert("Please input your website!");
        return false;
    }
    else {
        alert("You's already submit your website. ");
        return true;
    }
}
```

2. เพิ่มปุ่ม Submit เพื่อทำการตรวจสอบหลังส่งยืนยันการตรวจสอบ

3. ป้อนแอตทริบิวต์ ให้กับปุ่ม Submit ใน Code Behind ดังนี้

```
protected void Page_Load(object sender, EventArgs e)
{
    btnSubmit.Attributes.Add("onclick", "return isWebUrl(document.getElementById(
        '"+TextBox1.ClientID +"' ).value);");
}
```



หากไม่ระบุ http:// จะถือว่ารูปแบบไม่ถูกต้องเนื่องจากในโค้ดมีการระบุ /http: W ซึ่งก็มีความหมายคือ http:// ตามด้วย \b[A-Za-z0-9\.-]{3,}\. ตัวอักษร A-Z หรือ 0-9 ตามด้วย . จำนวนสามตัวขึ้นไป \.[A-Za-z]{3}/) ตามด้วย . และตัวอักษร 3 ตัว

7 เทคนิคป้อน วันที่และเวลา

การป้อนวันที่และเวลา สามารถนำคอนโทรล Ajax และจาวาสคริปต์มาประยุกต์ใช้ร่วมกันได้ไม่ยาก โดยเริ่มจากใช้คอนโทรล MaskEditorExtender ใน AjaxControlToolkit ไลบรารีของไมโครซอฟท์ เพื่อสร้าง mask ให้สามารถป้อนได้เพียงตัวเลขใน TextBox วางคอนโทรลลงในฟอร์มแล้วกำหนดคุณสมบัติ ดังนี้

```
<asp:MaskedEditExtender ID="MaskedEditExtender1" runat="server" Mask="99/99/9999"
    MaskType="Date" TargetControlID="TextBox1">
</asp:MaskedEditExtender>
```



จะเห็นว่าเมื่อลองรันโปรแกรม จะสามารถป้อนได้เฉพาะตัวเลข แต่ก็ยังไม่สามารถป้องกันการป้อนข้อมูลผิดได้ 100% เพราะวันที่คือ 1-31 ซึ่งแล้วแต่เดือนด้วย ส่วนเดือนก็มีเพียง 1-12 เท่านั้นโอกาสที่ผู้ใช้งานจะป้อนผิดก็ยังมีอยู่ ดังนั้นในตัวอย่างนี้จะมีการนำจาวาสคริปต์เข้ามาช่วยในการตรวจสอบรูปการป้อนวันที่และเวลาให้ถูกต้องจึงจะยอมให้ส่งข้อมูล

โดยฟังก์ชันที่ใช้ในการตรวจสอบคือ CheckDateFormat(textboxID, DateFormat) ซึ่งค่าของตัวแปร textboxID คือสตริงที่อ้างถึง ID ของ textbox ที่ต้องการตรวจสอบ ส่วน DateFormat จะเป็นตัวแปรแบบ Number ที่รับค่าตัวเลข ซึ่งในโค้ดนี้จะมีการตรวจสอบรูปแบบวันเดือนปีและเวลาอยู่ 7 ชนิด

```
function CheckDateFormat (textboxID, DateFormat)
{
    var textBox = document.getElementById(textboxID) ;
    if (textBox.value.length == 0)
    {
        return false;
    }

    var text = textBox.value;
    var dates = text.split('/');

    if (DateFormat == 4) // Only Time
    {
        dates = text.split(':');
    }

    var iDate, iMonth, iYear, iHours, iMins, iSecs, times;

    if (dates.length < 3)
    {
        return false;
    }

    if (dates[2].length > 2)
    {
        var iIndex = dates[2].indexOf(' ');
        times = dates[2].substring(iIndex+1).split(':');
    }

    switch (DateFormat)
    {
        case 1: // dd/MM/yyyy
            iDate = parseInt(dates[0]);
            iMonth = parseInt(dates[1]);
            iYear = parseInt(dates[2]);
            break;

        case 2: // MM/dd/yyyy
            iDate = parseInt(dates[1]);
            iMonth = parseInt(dates[0]);
            iYear = parseInt(dates[2]);
            break;

        case 3: // yyyy/MM/dddd
            iDate = parseInt(dates[2]);
            iMonth = parseInt(dates[1]);
            iYear = parseInt(dates[0]);
            break;

        case 4: // hh:mm:ss
            iHours = parseInt(dates[0]);
            iMins = parseInt(dates[1]);
            iSecs = parseInt(dates[2]);
            break;

        case 5: // dd/MM/yyyy hh:mm:ss
            iDate = parseInt(dates[0]);
            iMonth = parseInt(dates[1]);
            iYear = parseInt(dates[2]);
    }
}
```

ค้นหาคอนโทรลจาก ID

ตรวจสอบว่าเป็นค่าว่างหรือไม่ ถ้าเป็นก็ออกจากฟังก์ชัน

ทำการตัดสตริงด้วยตัวอักษร '/'


```
        iHours = parseInt(times[0]);
        iMins = parseInt(times[1]);
        iSecs = parseInt(times[2]);
        break;

    case 6: // MM/dd/yyyy hh:mm:ss
        iDate = parseInt(dates[1]);
        iMonth = parseInt(dates[0]);
        iYear = parseInt(dates[2]);

        iHours = parseInt(times[0]);
        iMins = parseInt(times[1]);
        iSecs = parseInt(times[2]);
        break;

    case 7: // yyyy/MM/dddd hh:mm:ss
        iDate = parseInt(dates[2]);
        iMonth = parseInt(dates[1]);
        iYear = parseInt(dates[0]);

        iHours = parseInt(times[0]);
        iMins = parseInt(times[1]);
        iSecs = parseInt(times[2]);
        break;
}

if (DateFormat != 4) {
    if (iMonth < 0 || iMonth > 12) // Month Range
    {
        alert("Month is out of range 1-12.");
        return false;
    }

    var d = new Date(iYear, iMonth, 0);
    var lastDate = d.getDate();

    if (iDate > lastDate) // Date Range
    {
        alert("Date is out of range 1-" + lastDate + ".");
        return false;
    }
}

if (DateFormat > 4) {

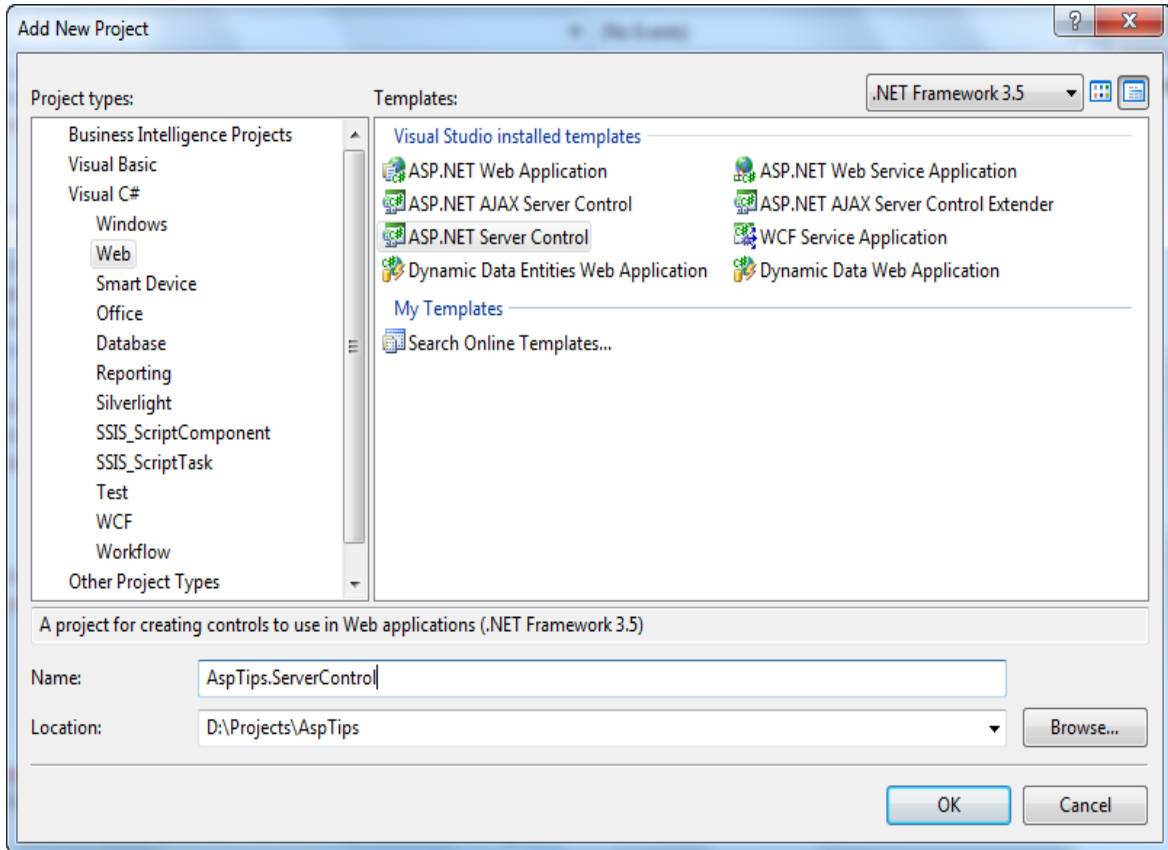
    if (iHours == NaN || (iHours < 1 || iHours > 12)) {
        alert("Hour is out of range 1-23.");
        return false;
    }

    if (iMins == NaN || (iMins < 0 || iMins > 59)) {
        alert("Minute is out of range 0-59.");
        return false;
    }

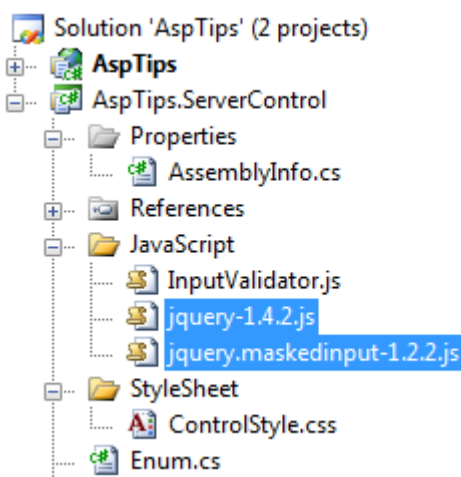
    if (iSecs == NaN || (iSecs < 0 || iSecs > 59)) {
        alert("Second is out of range 0-59.");
        return false;
    }
}
}
```

8 สร้างคอนโทรล TextBox ไว้ใช้งานเอง

จากเทคนิคการเขียนโปรแกรมกับ TextBox ที่ผ่านมา เราจะนำเทคนิคเหล่านั้นมารวบรวมอยู่ใน TextBox อันเดียว โดยเขียนเป็นคอนโทรลเพื่อนำไปใช้งานได้ทุกๆโปรเจคและจะช่วยให้ประหยัดเวลาในการพัฒนา



สร้างโปรเจคใหม่ เป็น ASP.NET Server Control



สร้างไฟล์เดอร์ชื่อ **JavaScript** สำหรับเก็บโค้ดภาษา JavaScript นำไฟล์ jquery-1.4.2.js และ jquery.maskedinput-1.2.2.js หรือหากมีเวอร์ชันที่สูงกว่าก็สามารถนำมาใช้ได้ ซึ่งโค้ดของ jquery สามารถโหลดได้จาก <http://jquery.org> เพิ่มไฟล์ InputValidator.js โดยนำโค้ดในตัวอย่างที่ผ่านมาเพิ่มลงในไฟล์นี้ จากนั้นทำการเพิ่มฟังก์ชัน ชื่อ ShowValidateMessage ใช้สำหรับสร้าง pop up เพื่อแสดงข้อความเมื่อเกิดข้อผิดพลาดบน TextBox

```
function ShowValidateMessage(obj, header, message)
{
```

```
    var oTbl = document.createElement("Table");
```

สร้างออบเจกต์ตาราง

```
    // Header
```

```
    var tdH = document.createElement("TD");
```

```
    var trH = document.createElement("TR");
```

```
    tdH.innerText = header;
```

```
    trH.appendChild(tdH);
```

สร้างเซลล์ "TD", สร้างบรรทัด "TR"

คุณสมบัติ innerText เป็นการอ้างถึงหรือ
กำหนดค่าตัวอักษรภายในเซลล์

```
    //Message
```

```
    var tdM = document.createElement("TD");
```

```
    var trM = document.createElement("TR");
```

```
    tdM.innerText = message;
```

```
    trM.appendChild(tdM);
```

ฟังก์ชัน appendChild เป็นการเพิ่มออบเจกต์
ลูกให้ที่คอมโพเนนต์แม่เข้าไปลำดับชั้น

```
    oTbl.appendChild(trH);
```

```
    oTbl.appendChild(trM);
```

```
    oTbl.style.top = obj.parentElement.offsetTop+5;
```

```
    oTbl.style.left = obj.parentElement.offsetLeft +10;
```

```
    oTbl.style.position = "absolute";
```

กำหนดค่าตำแหน่งของข้อความให้มีระยะ
เยื้องจาก TextBox

```
    oTbl.id ="tblMessage";
```

```
    obj.className = "RequiredBorder";
```

```
    oTbl.className = "PopupMessage";
```

กำหนดรูปแบบ stylesheet ให้กับ ตารางและ TextBox
Class RequireBoder และ PopupMessage จะ
กล่าวถึงในไฟล์ ControlStyle.css ต่อไป

```
    oTbl.onmouseover = function() { document.forms[0].removeChild(this); };
```

```
    obj.onclick = function() { this.className = ""; };
```

```
    document.forms[0].appendChild(oTbl);
```

```
}
```

- oTbl.onmouseover จะใส่ฟังก์ชันสำหรับย้ายตารางออกจาก TextBox เมื่อมีการนำตำแหน่งเมาส์วางบนตารางข้อความ
- obj.onclick จะทำการเคลียร์รูปแบบ stylesheet
- document.forms[0].appendChild(oTbl); เพิ่มตารางลงในฟอร์ม

เพิ่มฟังก์ชัน IsRequired สำหรับตรวจสอบค่าว่างใน TextBox

```
function IsRequired(txt)
```

```
{
```

```
    var svalue = trim(txt.value);
```

```
    if (svalue.length == 0)
```

```
{
```

```
        return false;
```

```
}
```

ตรวจสอบค่า ใน TextBox พร้อมตัดค่าว่างด้วยฟังก์ชัน
trim() หากมีการกดปุ่ม spacebar หากมีค่าว่างจะคืน
ค่าเป็น false

}
เพิ่มฟังก์ชัน ValidateEmail สำหรับตรวจสอบรูปแบบการป้อนอีเมล TextBox โดยนำฟังก์ชัน isEmail มาแก้ไขเปลี่ยนตัวแปรที่รับค่าจาก string เป็นออบเจกต์ textbox และไม่ต้องใช้ฟังก์ชัน alert

```
function ValidateEmail(txt)
{
    var strmail = txt.value;

    if
    (!strmail.match(/\b(^\S+@)\.+(\.com)|(\.net)|(\.edu)|(\.mil)|(\.gov)|(\.org)|(\.{2,2}))$
    )\b/gi)) {

        return false;
    }
}
```

เพิ่มฟังก์ชัน ValidateWebUrl สำหรับตรวจสอบรูปแบบการป้อนเว็บ TextBox โดยนำฟังก์ชัน isWebUrl มาแก้ไขเปลี่ยนตัวแปรที่รับค่าจาก string เป็นออบเจกต์ textbox และไม่ต้องใช้ฟังก์ชัน alert

```
function ValidateWebUrl(txt)
{
    var strurl = txt.value;
    var r1 = new RegExp("http://\/[A-Za-z0-9\.-]{3,}\.[A-Za-z]{3}");

    return (r1.test(strurl));
}
```

เพิ่มฟังก์ชัน ValidateDateFormat สำหรับตรวจสอบรูปแบบการป้อนอีเมล TextBox โดยนำฟังก์ชัน CheckDateFormat มาแก้ไขเปลี่ยนตัวแปรที่รับค่าจาก string ID ของ textbox เป็นออบเจกต์ textbox และไม่ต้องใช้ฟังก์ชัน alert เปลี่ยนเป็นการคืนค่าตัวเลขแทน

```
function ValidateDateFormat(txt, DateFormat) {
    var text = txt.value;

    if (text.length == 0) {
        return 0;
    }

    var dates = text.split('/');

    if (DateFormat == 4) // Only Time
    {
        dates = text.split(':');
    }

    var iDate, iMonth, iYear, iHours, iMins, iSecs, times;
```

อ่านบทความไอทีฟรีได้ที่ <http://borntodev.blogspot.com> สอน ASP.NET ฟรี สนใจติดต่อ apaichon@hotmail.com

```
if (dates.length < 3) {
    return 6;
}

if (dates[2].length > 2) {

    var iIndex = dates[2].indexOf(' ');
    times = dates[2].substring(iIndex + 1).split(':');
}

switch (DateFormat) {
    case 1: // dd/MM/yyyy
        iDate = parseInt(dates[0]);
        iMonth = parseInt(dates[1]);
        iYear = parseInt(dates[2]);
        break;

    case 2: // MM/dd/yyyy
        iDate = parseInt(dates[1]);
        iMonth = parseInt(dates[0]);
        iYear = parseInt(dates[2]);
        break;

    case 3: // yyyy/MM/dddd
        iDate = parseInt(dates[2]);
        iMonth = parseInt(dates[1]);
        iYear = parseInt(dates[0]);
        break;

    case 4: // hh:mm:ss
        iHours = parseInt(dates[0]);
        iMins = parseInt(dates[1]);
        iSecs = parseInt(dates[2]);
        break;

    case 5: // dd/MM/yyyy hh:mm:ss
        iDate = parseInt(dates[0]);
        iMonth = parseInt(dates[1]);
        iYear = parseInt(dates[2]);

        iHours = parseInt(times[0]);
        iMins = parseInt(times[1]);
        iSecs = parseInt(times[2]);
        break;

    case 6: // MM/dd/yyyy hh:mm:ss
        iDate = parseInt(dates[1]);
        iMonth = parseInt(dates[0]);
        iYear = parseInt(dates[2]);

        iHours = parseInt(times[0]);
        iMins = parseInt(times[1]);
        iSecs = parseInt(times[2]);
        break;

    case 7: // yyyy/MM/dddd hh:mm:ss
        iDate = parseInt(dates[2]);
        iMonth = parseInt(dates[1]);
        iYear = parseInt(dates[0]);

        iHours = parseInt(times[0]);
```

```
        iMins = parseInt(times[1]);
        iSecs = parseInt(times[2]);
        break;
    }

    if (DateFormat != 4) {
        if (iMonth < 0 || iMonth > 12) // Month Range
        {
            return 1;
        }

        var d = new Date(iYear, iMonth, 0);
        var lastDate = d.getDate();

        if (iDate > lastDate) // Date Range
        {
            return 2;
        }
    }

    if (DateFormat > 4) {
        if (iHours == NaN || (iHours < 0 || iHours > 23))
        {
            return 3;
        }

        if (iMins == NaN || (iMins < 0 || iMins > 59))
        {
            return 4;
        }

        if (iSecs == NaN || (iSecs < 0 || iSecs > 59))
        {
            return 5;
        }
    }

    return 0;
}
```

คืนค่า Error เป็นตัวเลขเพื่อนำไปใช้ในการแสดงข้อความ

เพิ่มฟังก์ชัน ShowDataErrorMessage สำหรับแสดงข้อความ เมื่อมีการป้อนรูปแบบวันที่ผิดพลาดตามเงื่อนไขต่างๆ ดังนี้

```
function ShowDateErrorMessage(txt, ErrorCode)
{
    switch (ErrorCode)
    {
        case 1:
            ShowValidateMessage(txt, "Invalid Month!", "Month is out of range 1-12.");
            break;
    }
}
```

กรณีที่ 1 แสดงข้อความเพื่อให้รู้ว่า ค่าเดือนที่ป้อนต้องอยู่ในระหว่าง 1-12 เท่านั้น

กรณีที่ 2 แสดงข้อความเพื่อให้รู้ว่า ค่าวันที่ที่ป้อนต้องอยู่ในระหว่าง 1-วันสุดท้ายของเดือน เท่านั้น

```
case 2:
    ShowValidateMessage(txt, "Invalid Date!", "Date is out of range 1-" + lastDate +
    ".");
    break;
```

กรณีที่ 3 แสดงข้อความเพื่อให้รู้ว่า ค่าชั่วโมงที่ป้อนต้องอยู่ในระหว่าง 0-23 เท่านั้น

```
case 3:
    ShowValidateMessage(txt, "Invalid Hour!", "Hour is out of range 0-23.");
    break;
```

กรณีที่ 4 แสดงข้อความเพื่อให้รู้ว่า ค่านาทีที่ป้อนต้องอยู่ในระหว่าง 0-59 เท่านั้น

```
case 4:
    ShowValidateMessage(txt, "Invalid Minute!", "Minute is out of range 0-59.");
    break;
```

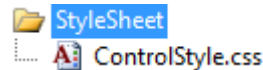
กรณีที่ 5 แสดงข้อความเพื่อให้รู้ว่า ค่าวินาทีที่ป้อนต้องอยู่ในระหว่าง 0-59 เท่านั้น

```
case 5:
    ShowValidateMessage(txt, "Invalid Second!", "Second is out of range 0-59.");
    break;
```

กรณีที่ 6 แสดงข้อความเพื่อให้รู้ว่า การป้อนรูปแบบวันที่ยังไม่สมบูรณ์

```
case 6:
    ShowValidateMessage(txt, "Date incompleted!", "Please input completed date
    format.");
    break;
}
}
```

สร้างไฟล์เดอร์ชื่อ StyleSheet เพื่อเก็บไฟล์ การจัดรูปแบบ StyleSheet
ภายในไฟล์ ControlStyle.css เพิ่มฟังก์ชันดังนี้



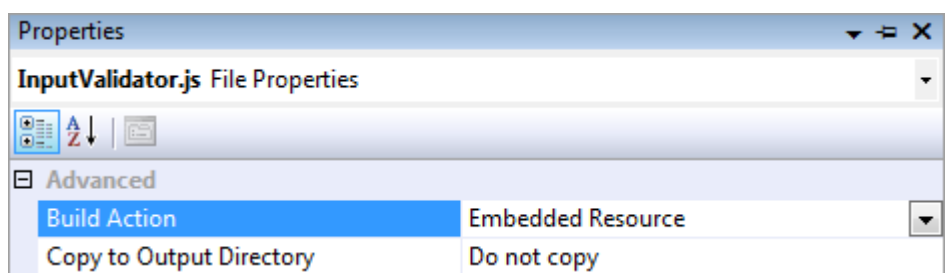
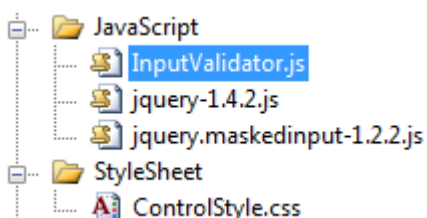
```
.RequiredBorder
{
    border-width:2px;
    border-style:solid;
    border-color:red;
}
```

ในฟังก์ชันนี้จะกำหนดรูปแบบให้กับคอนโทรล มีขอบสีแดงหนา 2 px

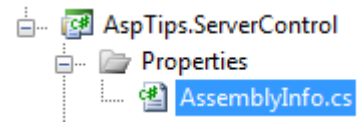
```
.PopupMessage
{
    border-width:2px;
    border-style:solid;
    border-color:red;
    background-color:#FFFFCC;
}
```

ในฟังก์ชันนี้จะกำหนดรูปแบบให้กับคอนโทรล มีขอบสีแดงหนา 2 px พื้นหลัง เป็นสีรหัส #FFFFCC

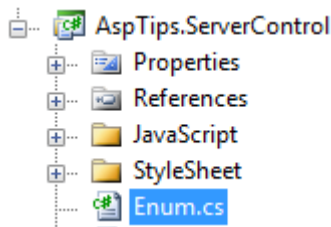
ในไฟล์ JavaScript และ StyleSheet นั้น ให้กำหนดคุณสมบัติของ Build Action เป็น Embedded Resource



ในไฟล์ AssemblyInfo.cs ทำการลงทะเบียน Resource ต่างๆที่ใช้ในคอนโทรล



```
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.InputValidator.js",
"text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery-1.4.2.js",
"text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.maskedinput-
1.2.2.js", "text/javascript")]
[assembly:
System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.TextBoxMultiType.js",
"text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.StyleSheet.ControlStyle.css",
"text/css", PerformSubstitution = true)]
```

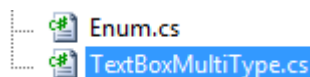


สร้างไฟล์ชื่อ Enum.cs เพื่อกำหนดรูปแบบประเภทของข้อมูลที่ป้อนใน TextBox ดังนี้

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace AspTips.ServerControl
{
    public enum ValidateType
    {
        Text
        , Integer
        , Double
        , ZipCode
        , PhoneNumber
        , Email
        , WebUrl
        , DateTime_ddMMyyyy
        , DateTime_MMddyyyy
        , DateTime_yyyyMMdd
        , DateTime_hhmmss
        , DateTime_ddMMyyyyhhmmss
        , DateTime_MMddyyyyhhmmss
        , DateTime_yyyyMMddhhmmss
    }
}
```

สร้างไฟล์ชื่อ TextBoxMultiType.cs




```
[DefaultProperty("Text")]
[ToolboxData("<{0}:TextBoxMultiType runat=server></{0}:TextBoxMultiType>")]
public class TextBoxMultiType : TextBox
{
    [Bindable(true)]
    [Category("Appearance")]
    [DefaultValue("")]
    [Localizable(true)]
    public string Text
    {
        get
        {
            String s = (String)ViewState["Text"];
            return ((s == null) ? "[" + this.ID + "]" : s);
        }
        set
        {
            ViewState["Text"] = value;
        }
    }
}
```

สืบทอดคุณสมบัติจาก TextBox

เพิ่มตัวแปรและคุณสมบัติให้ TextBox ดังนี้

```
private ValidateType _ValidateType = ValidateType.Text ;
private bool _IsRequired = false;
private string _TargetControlID = "";
```

```
[DefaultValue(ValidateType.Text)]
public ValidateType InputValidateType
{
    get { return _ValidateType; }
    set { _ValidateType = value; }
}

[DefaultValue(false)]
public bool IsRequired
{
    get { return _IsRequired; }
    set { _IsRequired = value; }
}

public string TargetControlID
{
    get { return _TargetControlID; }
    set { _TargetControlID = value; }
}
```

คุณสมบัติ InputValidateType สำหรับ
กำหนดค่า ชนิดของข้อมูลใน TextBox ว่า
เป็นประเภทไหน

คุณสมบัติ IsRequired สำหรับกำหนดค่า ใน
TextBox ว่าเป็นค่าว่างได้หรือไม่

คุณสมบัติ TargetControlID สำหรับ
กำหนดค่า ID ของคอนโทรลที่จะทำการ
ตรวจสอบข้อมูลใน TextBox เมื่อมีการ click

เพิ่มฟังก์ชันชื่อ LoadAttributes ซึ่งฟังก์ชันนี้จะทำหน้าที่ตรวจสอบประเภทข้อมูลของ TextBox ว่าถูกกำหนดเป็นประเภทใดและจะกำหนดฟังก์ชันในจาวาสคริปต์ให้เข้ากับเงื่อนไขของชนิดข้อมูลแต่ละประเภท

```
private void LoadAttributes()
{
```

```
    ClientScriptManager cs = this.Page.ClientScript;

    string script = @"jQuery(function($){ $('#"+
this.ClientID + "').mask('@MaskType');});";
```

เรียกใช้ฟังก์ชัน jQuery ซึ่งอยู่ในไฟล์ jquery-x.xx.js และ jquery.maskinput.xx.js พร้อมส่ง ID ของ TextBox และกำหนด Mask Type

ประกาศตัวแปรรับ ClientScript ออบเจกต์จากหน้าเว็บที่เรียกใช้งานคอนโทรลเพื่อทำการลงทะเบียน จาวาสคริปต์

```
string ValidateScript = "";
```

```
this.Attributes.Add("onclick", "removeMessageBox();");
```

เรียกใช้ฟังก์ชัน removeMessageBox โปรแกรมจะทำการลบ MessageBox ออก เมื่อมีการคลิกที่ TextBox

```
switch (InputValidateType)
{
```

```
    case ValidateType.Double:
```

```
        this.Attributes.Add("onkeypress", "return onlyDouble(this);");
        break;
```

กรณีที่กำหนด InputValidateType เป็นประเภท Double เมื่อมีการพิมพ์ที่คีย์บอร์ด จะเรียกใช้ฟังก์ชัน onlyDouble เพื่อให้สามารถพิมพ์ได้เพียงตัวเลขและจุดทศนิยมได้เท่านั้น

```
    case ValidateType.Email:
```

```
        ValidateScript = "var txt= document.getElementById('" + this.ClientID +
        "') ;var b =ValidateEmail(txt);" +
        "if(b==false){ShowValidateMessage(txt, 'Invalid Email!', 'Please input
correctly email format such as example@exmample.com!'); return b;};";
        break;
```

กรณีที่กำหนด InputValidateType เป็นประเภท Email จะทำการตรวจสอบว่าเป็นรูปแบบอีเมลหรือไม่ เมื่อมีการ click ที่ TargetControl หากไม่ใช่จะแสดงข้อความเตือน

```
    case ValidateType.Integer:
```

```
        this.Attributes.Add("onkeypress", "return onlyInteger();");
        break;
```

กรณีที่กำหนด InputValidateType เป็นประเภท Integer เมื่อมีการพิมพ์ที่คีย์บอร์ด จะเรียกใช้ฟังก์ชัน onlyInteger เพื่อให้สามารถพิมพ์ได้เพียงตัวเลขได้เท่านั้น

อ่านบทความไอทีฟรีได้ที่ <http://bor>

```

    case ValidateType.PhoneNumber:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "(999) 999-9999"), true);
        break;

```

กรณีที่กำหนด InputValidateType เป็นประเภท PhoneNumber จะทำการตรวจสอบว่าเป็นรูปแบบเบอร์โทรศัพท์หรือไม่ เมื่อมีการ click ที่ TargetControl หากไม่ใช่จะแสดงข้อความเตือน

```

    case ValidateType.Text:
        break;

```

```

    case ValidateType.WebUrl:

```

```

        ValidateScript = "var txt= document.getElementById('" + this.ClientID +
        "') ;var b= ValidateWebUrl(txt); " +
        "if(b==false) {ShowValidateMessage(txt, 'Invalid WebUrl!', 'Please
input correctly Web Url format such as http://www.exempl.com!'); return b;};";
        break;

```

กรณีที่กำหนด InputValidateType เป็นประเภท WebUrl จะทำการตรวจสอบว่าเป็นรูปแบบเบอร์เว็บไซต์หรือไม่ เมื่อมีการ click ที่ TargetControl หากไม่ใช่จะแสดงข้อความเตือน

```

    case ValidateType.ZipCode:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "99999"), true);
        break;

```

กรณีที่กำหนด InputValidateType เป็นประเภท ZipCode จะทำการตรวจสอบว่าเป็นรูปแบบเบอร์รหัสไปรษณีย์หรือไม่ เมื่อมีการ click ที่ TargetControl หากไม่ใช่จะแสดงข้อความเตือน

```

    case ValidateType.DateTime_ddMMyyyy:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "99/99/9999"), true);
        ValidateScript = "var txt = document.getElementById('" + this.ClientID +
        "') ;var d= ValidateDateFormat(txt,1); if (d>0){ShowDateErrorMessage(txt,d);return
false;};";
        break;

```

กรณีที่กำหนด InputValidateType เป็นประเภท DateTime_ddMMyyyy จะทำการตรวจสอบว่าเป็นรูปแบบวันที่แบบ วัน/เดือน/ปี หรือไม่ เมื่อมีการ click ที่ TargetControl หากไม่ใช่หรือค่าวันที่หรือเดือนไม่มีอยู่จริงจะแสดงข้อความเตือน

```

    case ValidateType.DateTime_MMdyyyy:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "99/99/9999"), true);

```

```

ValidateScript = "var txt = document.getElementById('" + this.ClientID +
");var d= ValidateDateFormat(txt,2); if (d>0){ShowDateErrorMessage(txt,d);return
false;};"; break;

    case ValidateType.DateTime_yyyyMMdd:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "9999/99/99"), true);
        ValidateScript = "var txt = document.getElementById('" + this.ClientID +
");var d= ValidateDateFormat(txt,3); if (d>0){ShowDateErrorMessage(txt,d);return
false;};";

        break;

    case ValidateType.DateTime_hhmmss:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "99:99:99"), true);
        ValidateScript = "var txt = document.getElementById('" + this.ClientID +
");var d= ValidateDateFormat(txt,4); if (d>0){ShowDateErrorMessage(txt,d);return
false;};";

        break;

    case ValidateType.DateTime_ddMMyyyyhhmmss:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "99/99/9999 99:99:99"), true);
        ValidateScript = "var txt = document.getElementById('" + this.ClientID +
");var d= ValidateDateFormat(txt,5); if (d>0){ShowDateErrorMessage(txt,d);return
false;};";

        break;

    case ValidateType.DateTime_MMddyyyyhhmmss :
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "99/99/9999 99:99:99"), true);
        ValidateScript = "var txt = document.getElementById('" + this.ClientID +
");var d= ValidateDateFormat(txt,6); if (d>0){ShowDateErrorMessage(txt,d);return
false;};";

        break;

    case ValidateType.DateTime_yyyyMMddhhmmss:
        cs.RegisterStartupScript(this.GetType(), "mask" + this.ClientID,
script.Replace("@MaskType", "9999/99/99 99:99:99"), true);
        ValidateScript = "var txt = document.getElementById('" + this.ClientID +
");var d= ValidateDateFormat(txt,7); if (d>0){ShowDateErrorMessage(txt,d);return
false;};";

        break;
}

if (!string.IsNullOrEmpty ( TargetControlID))
{
    object oTarget = this.Page.FindControl(TargetControlID);
    Type tTarget = oTarget.GetType();

```

หากมีการระบุ TargetControlID จะทำการหาคอนโทรลเป้าหมาย เมื่อมีการคลิกที่
คอนโทรลนั้นจะทำการตรวจสอบ TextBox ตาม InputValidateType ที่กำหนด

หากมีการระบุ IsRequired เป็น true จะใส่ฟังก์ชันเพื่อนตรวจสอบและไม่ยอมให้ TextBox มีค่าเป็นค่าว่าง

```
if (IsRequired)
{
    ValidateScript = "var txt = document.getElementById('" + this.ClientID +
    "'); var b= IsRequired(txt); if (b==false){ShowValidateMessage(txt, 'Required!', 'Please
    input value!'); return b;};" + ValidateScript;
}
```

ตรวจสอบว่า TargetControl เป็นคอนโทรลประเภทใดแล้ว ทำการกำหนดฟังก์ชันในการตรวจสอบความถูกต้องของรูปแบบข้อมูลใน TextBox เมื่อมีการคลิกที่คอนโทรล

```
switch (tTarget.FullName)
{
    case "System.Web.UI.WebControls.Button":
        Button btn = (Button)oTarget;
        if (btn.Attributes["onclick"] != null)
        {
            ValidateScript = btn.Attributes["onclick"] + ValidateScript;
        }
        btn.Attributes.Add("onclick", ValidateScript);
        break;

    case "System.Web.UI.WebControls.LinkButton":
        LinkButton lnkBtn = (LinkButton)oTarget;
        if (lnkBtn.Attributes["onclick"] != null)
        {
            ValidateScript = lnkBtn.Attributes["onclick"] + ValidateScript;
        }
        lnkBtn.Attributes.Add("onclick", ValidateScript);
        break;

    case "System.Web.UI.WebControls.ImageButton":
        ImageButton imbBtn = (ImageButton)oTarget;
        if (imbBtn.Attributes["onclick"] != null)
        {
            ValidateScript = imbBtn.Attributes["onclick"] + ValidateScript;
        }
        imbBtn.Attributes.Add("onclick", ValidateScript);
        break;
}
}
```

เพิ่มเหตุการณ์ onPreRender ซึ่งจะถูกเกิดขึ้นเมื่อมีการนำคอนโทรลไปวางในหน้าจ่ออกแบบ

```
protected override void OnPreRender (EventArgs e)
{
    base.OnPreRender (e);

    string JQuery = "AspTips.ServerControl.JavaScript.jquery-1.4.2.js";
    string JMask = "AspTips.ServerControl.JavaScript.jquery.maskedinput-1.2.2.js";
}
```

```
string JavaScriptName = "AspTips.ServerControl.JavaScript.InputValidator.js";
string StyleSheetName = "AspTips.ServerControl.StyleSheet.ControlStyle.css";
ClientScriptManager cs = this.Page.ClientScript;
cs.RegisterClientScriptResource(this.GetType(), JavaScriptName);
cs.RegisterClientScriptResource(this.GetType(), JQuery);
cs.RegisterClientScriptResource(this.GetType(), JMask);
```

ทำการลงทะเบียนฟังก์ชันจากจาวาสคริปต์และ StyleSheet ที่มีการเรียกใช้งานในคอนโทรล

```
string includeLocation = cs.GetWebResourceUrl(this.GetType(), StyleSheetName);
HtmlLink cssLink = new HtmlLink();
```

```
cssLink.Href = includeLocation;
```

```
cssLink.Attributes.Add("rel", "stylesheet");
```

```
cssLink.Attributes.Add("type", "text/css");
```

```
this.Page.Header.Controls.Add(cssLink);
```

```
LoadAttributes();
```

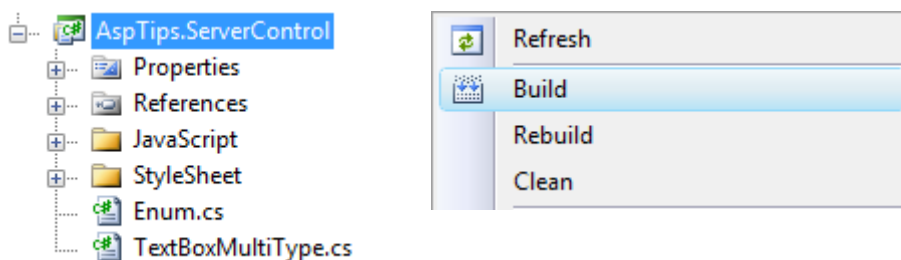
ทำการลงทะเบียน StyleSheet ให้อยู่ในส่วน Header ของเว็บเพจ

```
}
```

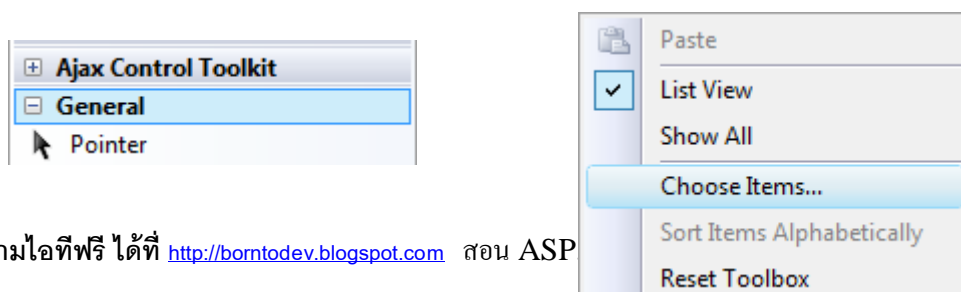
เรียกใช้ฟังก์ชัน LoadAttributes เพื่อกำหนดเงื่อนไขให้กับ TextBox

การนำคอนโทรลไปใช้งาน

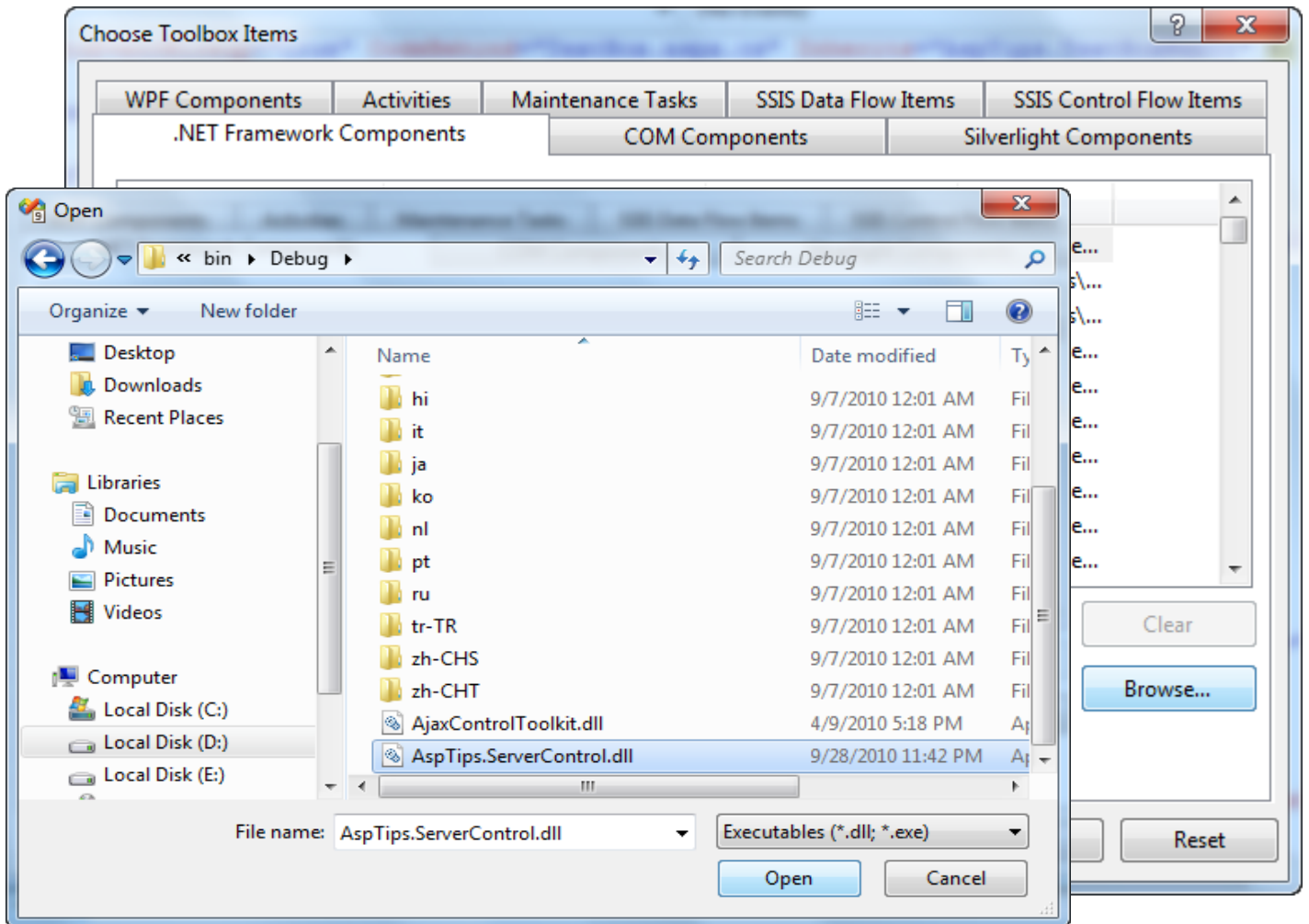
1. คลิกขวาที่โปรเจค เลือก Build



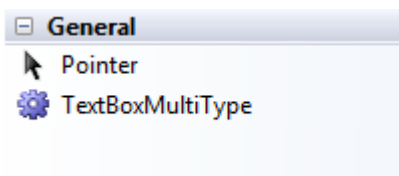
2. เมื่อโปรเจค Build สำเร็จไม่มีบั๊กเกิดขึ้น ให้กลับไปเว็บโปรเจค แล้วทำการดับเบิลคลิกที่หน้าเว็บเพจ .aspx หน้าใดหน้าหนึ่ง จากนั้นเลือก Toolbox เลือกแท็บที่ต้องการเพิ่มคอนโทรลเข้าไป แล้วคลิกขวาเลือก Choose Items ...



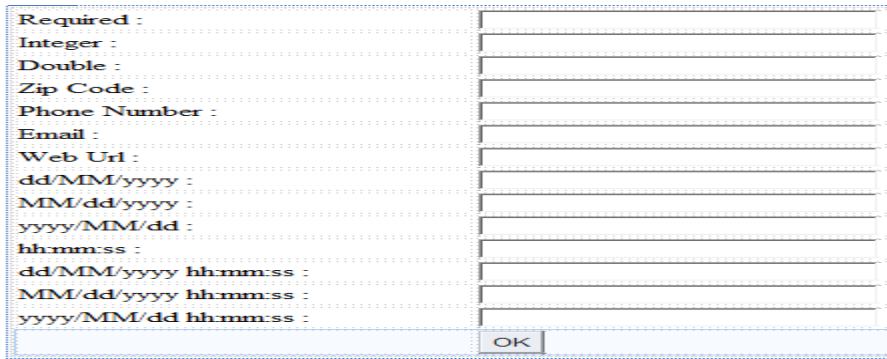
3. เลือกไฟล์นามสกุล dll ที่ถูก Build แล้ว



คอนโทรลจะถูกเพิ่มอยู่ในแท็บที่เราเลือกไว้



4. ทดสอบโดยการลากคอนโทรลไปวางในเว็บเพจแล้วกำหนดค่า เป็น InputValidateType ประเภทต่างๆ ดังรูป

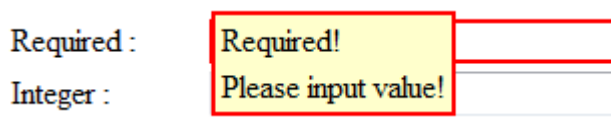


กำหนดคุณสมบัติต่างๆของ TextBox ตามลำดับ ดังนี้

ลำดับ	คุณสมบัติ	ค่า
1	IsRequired	True
	TargetControlID	btnOK * ในตัวอย่างนี้ให้กำหนด TargetControlID เป็นปุ่ม btnOK เหมือนกันหมด
2	InputValidateType	Integer
3	InputValidateType	Double
4	InputValidateType	ZipCode
5	InputValidateType	PhoneNumber
6	InputValidateType	Email
7	InputValidateType	WebUrl
8	InputValidateType	DateTime_ddMMyyyy
9	InputValidateType	DateTime_MMddyyyy
10	InputValidateType	DateTime_yyyyMMdd
11	InputValidateType	DateTime_hhmmss
12	InputValidateType	DateTime_ddMMyyyyhhmmss
13	InputValidateType	DateTime_MMddyyyyhhmmss
14	InputValidateType	DateTime_yyyyMMddhhmmss

5. ทดสอบ TextBox แต่ละตัว

5.1 ทดสอบ IsRequired ให้กดปุ่ม OK โดนที่ไม่ต้องป้อนข้อมูลอะไร หรือ ทำการกด spacebar



5.2 ทดสอบพิมพ์ข้อความที่ TextBox Integer จะสามารถพิมพ์ได้เฉพาะตัวเลขจำนวนจริง



5.3 ทดสอบพิมพ์ข้อความที่ TextBox Double จะสามารถพิมพ์ได้เฉพาะตัวเลขและจุดทศนิยม

Double :

5.4 ทดสอบพิมพ์ข้อความที่ TextBox ZipCode จะสามารถพิมพ์ได้เฉพาะตัวเลข 5 หลัก

Zip Code :

5.5 ทดสอบพิมพ์ข้อความที่ TextBox PhoneNumber จะสามารถพิมพ์ได้เฉพาะตัวเลขในรูปแบบโทรศัพท์ที่กำหนดไว้

Phone Number :

5.6 ทดสอบพิมพ์ข้อความเช่น abcd ที่ TextBox Email แล้วกดปุ่ม OK

Email :
 Web Url :

จะมี ข้อความเตือนว่ารูปแบบการป้อนต้องเป็นรูปแบบของ เมล์เท่านั้น

5.7 ทดสอบพิมพ์ข้อความเช่น hello ที่ TextBox Web Url แล้วกดปุ่ม OK

Web Url :
 dd/MM/yyyy :

5.8 ทดสอบป้อนวันที่ให้กับ Datetime แบบ dd/MM/yyyy โดยป้อนแค่วันที่ แล้วกดปุ่ม OK

dd/MM/yyyy :

เมื่อกดปุ่ม OK จะมีข้อความเตือนว่ารูปแบบวันที่ยังไม่สมบูรณ์

ให้ทดลองป้อนวันที่เกินช่วงของเดือนนั้น จะมีข้อความเตือนว่าช่วงวันที่เกิน

dd/MM/yyyy :

ให้ทดลองป้อนเดือนเกิน 12 จะมีข้อความเตือนเมื่อป้อนช่วงเดือนเกิน

dd/MM/yyyy :

ทดสอบกับรูปแบบวันที่แบบต่างๆเหมือนกับขั้นตอนที่ 2.8 ว่าคอนโทรลสามารถทำงานได้ถูกต้องหรือไม่

บทที่ 2 เทคนิคในการสร้างคอนโทรล TextBoxFilter สำหรับกรองข้อมูล

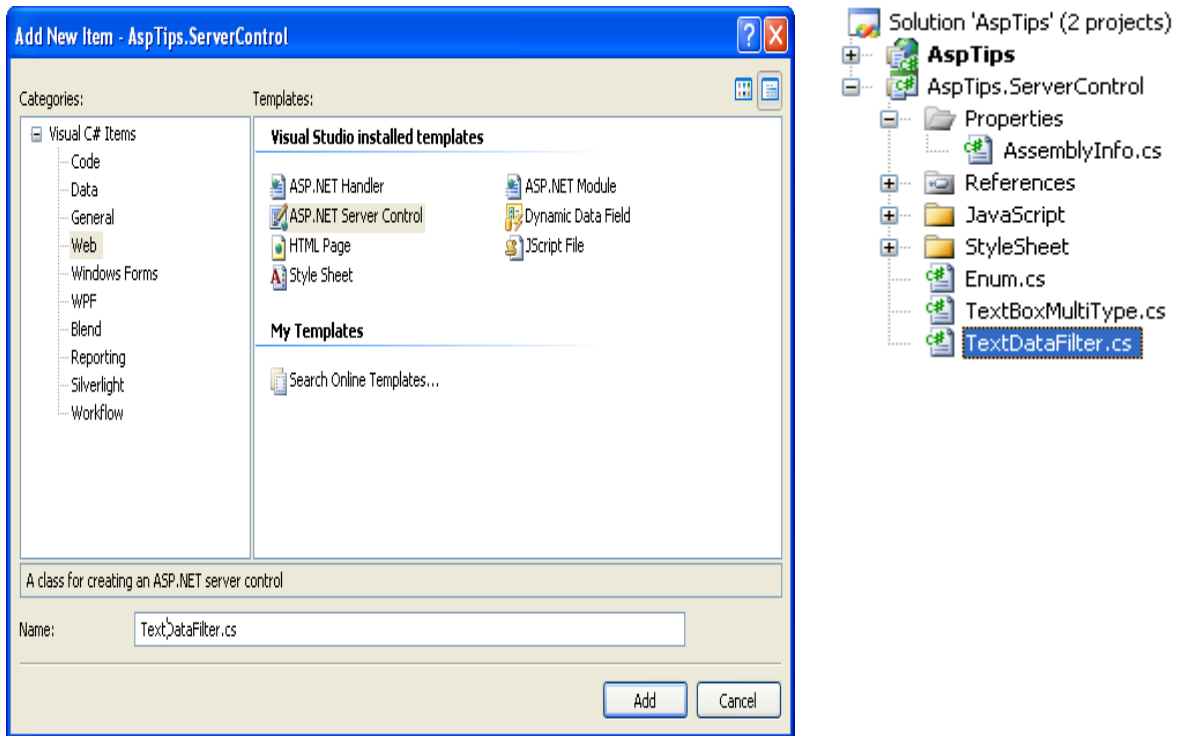


บทนำ

ในการป้อนข้อมูลใน TextBox ผู้ใช้งานอาจต้องการความสะดวกในการเลือกข้อมูลที่มีอยู่แล้วในระบบเพื่อความสะดวกรวดเร็ว ซึ่งการนำ TextBox มาใช้ในการค้นหาและกรองข้อมูล ปัจจุบันได้มีระบบหลายๆตัวนำมาประยุกต์ใช้เพื่อความง่ายและรวดเร็วของผู้ใช้งาน เช่น ระบบของตัว สามารถค้นหาต้นทางและปลายทางได้จากการพิมพ์ข้อความใน TextBox ระบบก็จะทำการกรองข้อมูลให้เหลือ

ขั้นตอนสร้างคอนโทรล

1. สร้าง ASP.NET Server Control ชื่อ TextDataFilter.cs



2. แก้ไขโค้ด โดยทำการสืบทอดคุณสมบัติคอนโทรลมาจาก TextBox

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:TextDataFilter runat=server></{0}:TextDataFilter>")]
public class TextDataFilter : TextBox
{
```

สืบทอดคุณสมบัติจาก TextBox

3. สร้าง Property สำหรับกำหนดค่าคอนโทรล ดังนี้

ลำดับ	คุณสมบัติ	ชนิดตัวแปร	คำอธิบาย
1	ConnectionString	string	เก็บค่าการเชื่อมต่อฐานข้อมูล Ms-Sql
2	DataValues	string	เก็บค่าคือข้อมูลที่ทำการกรองจากฐานข้อมูล
3	MaxFilter	int	กำหนดจำนวนสูงสุดของข้อมูลในการกรอง
4	SelectCommand	string	กำหนดคำสั่ง sql ในการค้นหาข้อมูล
5	TextDataField	string	กำหนดชื่อคอลัมภ์ในฐานข้อมูลที่ต้องการนำมาแสดงข้อมูล
6	ValueDataField	string	กำหนดชื่อคอลัมภ์ในฐานข้อมูลที่เป็นคีย์
7	Text	string	คุณสมบัติหลักของ TextBox ในการแสดงข้อความและเก็บค่า
8	ValidChars	int	กำหนดจำนวนตัวอักษรในการเริ่มต้นค้นหาเมื่อมีการพิมพ์ข้อความ

```
public string ConnectionString
{
    get
    {
        String s = (String)ViewState["ConnectionString"];
        return ((s == null) ? String.Empty : s);
    }

    set
    {
        ViewState["ConnectionString"] = value;
    }
}

public string DataValues
{
    get
    {
        String s = (String)ViewState["DataValues"];
        return ((s == null) ? String.Empty : s);
    }

    set
    {
        ViewState["DataValues"] = value;
    }
}

public int MaxFilter
{
    get
    {
        int iMaxFilter = (ViewState["MaxFilter"] == null ? 50 :
            (int)ViewState["MaxFilter"]);
        return iMaxFilter;
    }
    set { ViewState["MaxFilter"] = value; }
}

public string SelectCommand
{
    get
    {
        String s = (String)ViewState["SelectCommand"];
        return ((s == null) ? String.Empty : s);
    }

    set
    {
        ViewState["SelectCommand"] = value;
    }
}

public string TextDataField
{
    get
    {
        String s = (String)ViewState["TextDataField"];
        return ((s == null) ? String.Empty : s);
    }
}
```

```
        set
        {
            ViewState["TextDataField"] = value;
        }
    }

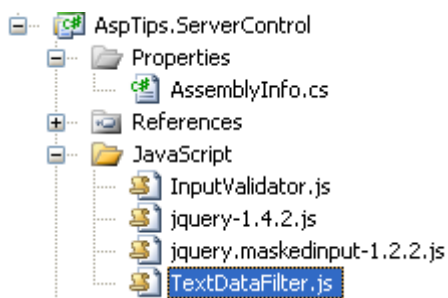
    public string ValueDataField
    {
        get
        {
            String s = (String)ViewState["ValueDataField"];
            return ((s == null) ? String.Empty : s);
        }

        set
        {
            ViewState["ValueDataField"] = value;
        }
    }

    [Bindable(true)]
    [Category("Appearance")]
    [DefaultValue("")]
    [Localizable(true)]
    public string Text
    {
        get
        {
            String s = (String)ViewState["Text"];
            return ((s == null) ? String.Empty : s);
        }
        set
        {
            ViewState["Text"] = value;
        }
    }

    public int ValidChars
    {
        get
        {
            int iValidChars = (ViewState["ValidChars"] == null ? 3 :
                (int)ViewState["ValidChars"]);
            return iValidChars;
        }
        set { ViewState["ValidChars"] = value; }
    }
}
```

4. สร้าง JavaScript ชื่อ TextDataFilter.js และป้อนโค้ด ดังนี้



```
var tbResult;
```

ประกาศตัวแปรชื่อ tbResult เป็นแบบโกลบอลเพื่อให้มองเห็นได้จากทุกฟังก์ชัน ตัวแปรนี้จะเป็นตัวแปรที่ใช้แสดงผลในการค้นหาข้อมูล

```
function onChangePostBack(txt, validChars)
```

```
{
  var Text = txt.value;
  if (Text.length >= validChars)
  {
    document.forms[0].submit();
  }
  else {
    if (tbResult != null)
    {
      try
      {
        document.forms[0].removeChild(tbResult);
        document.forms[0].submit();
      }
      catch (Error)
      {
      }
    }
  }
}
```

สร้างฟังก์ชัน onChangePostBack เพื่อทำการตอบสนองการทำงานฝั่ง server เมื่อ textbox มีการพิมพ์จำนวนตัวอักษรได้ตามที่เรากำหนดไว้

หากตัวอักษรน้อยกว่าที่กำหนดไว้ แล้วมีการแสดงผลการค้นหา ให้ทำการลบตารางออกจากฟอร์ม

```
function ShowSearchResult(txt)
```

```
{
  tbResult.style.display = "";
  tbResult.style.top = txt.parentElement.offsetTop + 30;
  tbResult.style.left = txt.parentElement.offsetLeft + 10;
  tbResult.style.position = "absolute";
}
```

แสดงผลการค้นหา โดยให้มีตำแหน่งเยื้องจากตำแหน่งของ TextBox ทางมุมบนและซ้ายเล็กน้อย

นำข้อมูลที่ค้นหาได้จากฐานข้อมูลมาแสดงบนตาราง

```
function AddSearchResult(dataList,txt,MaxRows,values)
```

```
{
  if (tbResult == null)
  {
    tbResult = document.createElement("Table");
    document.forms[0].appendChild(tbResult);
  }
}
```

ตรวจสอบว่ามีตารางอยู่ในฟอร์มหรือไม่ถ้าไม่มีให้เพิ่มในฟอร์ม

```
var datas = dataList.split(';');
```

ทำการตัดข้อความให้อยู่ในรูปอาร์เรย์

```
var dataLength = datas.length -2;
if (dataLength > MaxRows)
{
  dataLength = MaxRows;
}
```

หากข้อมูลมีมากกว่า ค่าสูงสุดที่กำหนดให้แสดงผลให้แสดงผลเท่ากับค่าสูงสุดเท่านั้น

```
for (var i = 0; i < dataLength; i++)
{
```

```
  var tdH = document.createElement("TD");
  var trH = document.createElement("TR");
```

ป้อนข้อมูลที่ค้นหาได้ลงในตาราง

```

tdH.innerText = datas[i];
tdH.className = "NormalStyle";
tdH.onmouseover = function() { this.className = "OnOverStyle"; };
tdH.onmouseout = function() { this.className = "NormalStyle"; };
tdH.onclick = function() {

    txt.value = this.innerText;
    tbResult.style.display = "none";

};
trH.appendChild(tdH);
tbResult.appendChild(trH);
}
ShowSearchResult(txt);
}

function FocusLastIndex(txt)
{

    var cursorPosition = txt.value.length;

    if (txt.createTextRange) {
        var textRange = txt.createTextRange();
        textRange.move('character', cursorPosition);
        textRange.select();
    }
    else {
        if (txt.selectionStart)
        {
            txt.focus();
            txt.setSelectionRange(cursorPosition, cursorPosition);
        }
        else
        {
            txt.focus();
        }
    }
}
}

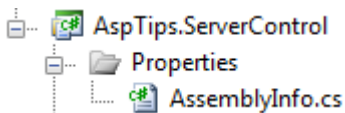
```

เมื่อมีการคลิกที่ข้อมูลบนเซลล์ให้แสดง
ข้อมูลใน TextBox และทำการซ่อนตาราง

เรียกฟังก์ชัน ShowSearchResult เพื่อ
แสดงตารางข้อมูล

ฟังก์ชัน FocusLastIndex ใช้สำหรับกำหนด
ตำแหน่ง Focus ให้กับ textbox ที่ตำแหน่ง
ตัวอักษรตัวสุดท้ายเสมอเมื่อมีการพิมพ์ข้อความ

5. ทำการลงทะเบียน JavaScript ใน AssemblyInfo.cs



```
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.TextDataFilter.js", "text/javascript")]
```

6. ที่ไฟล์ TextDataFilter.cs เพิ่มฟังก์ชัน ดังนี้

```
protected override void RenderContents(HtmlTextWriter output)
{
    output.Write(Text);
}

```

```
protected override void OnPreRender(EventArgs e)
```

อ่านบทความไอทีฟรีได้ที่ <http://borntodev.blogspot.com> สอน ASP.NET ฟรี สนใจติดต่อ apaichon@hotmail.com


```

{
    base.OnPreRender(e);

    string JavaScriptName = "AspTips.ServerControl.JavaScript.TextDataFilter.js";
    ClientScriptManager cs = this.Page.ClientScript;
    cs.RegisterClientScriptResource(this.GetType(), JavaScriptName);

    string StyleSheetName = "AspTips.ServerControl.StyleSheet.ControlStyle.css";

    string includeLocation = cs.GetWebResourceUrl(this.GetType(), StyleSheetName);
    HtmlLink cssLink = new HtmlLink();

    cssLink.Href = includeLocation;

    cssLink.Attributes.Add("rel", "stylesheet");

    cssLink.Attributes.Add("type", "text/css");

    this.Page.Header.Controls.Add(cssLink);

    this.Attributes.Add("onkeyup", "onChangePostBack(this, "+this.ValidChars.ToString() +");");
}

```

ก่อนที่คอนโทรลจะถูกเขียนใน แอปพลิเคชัน ทำการลงทะเบียน JavaScript และ StyleSheet ที่จะใช้งานในคอนโทรลเสียก่อน

```

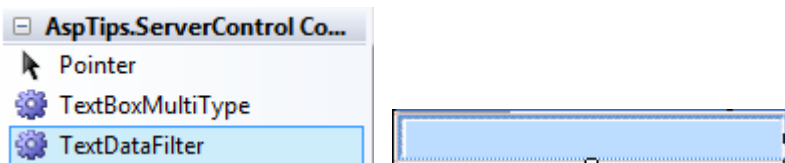
protected override void OnLoad(EventArgs e)
{
    base.OnLoad(e);

    this.TextChanged += new EventHandler(TextDataFilter_TextChanged);
}

```

การนำคอนโทรลไปใช้งาน

1. วางคอนโทรลลงในฟอร์ม



2. กำหนดคุณสมบัติ ดังนี้

ลำดับ	คุณสมบัติ	ค่า
1	ConnectionString	Data Source=APAICHON-PC;Initial Catalog=TestDB;Integrated Security=True (กำหนดค่าในการเชื่อมต่อฐานข้อมูล MS-SQL)
2	SelectCommand	Select * from Product Where Product_Name Like '%@TextSearch%' คำสั่ง sql ที่ใช้ในการดึงข้อมูล โดยค่าที่พิมพ์ใน

		TextBox จะไปแทนที่ค่าพารามิเตอร์ @TextSearch
3	TextDataField	Product_Name ชื่อคอมลัมน์ที่ใช้แสดงผล
4	ValueDataField	Product_ID ชื่อคอมลัมน์ในฐานข้อมูลที่เป็นคีย์
5	ValidChars	3 เมื่อพิมพ์ข้อความตั้งแต่ 3 คำจะเริ่มค้นหา

3. ทำการรัน เว็บแอปพลิเคชัน แล้วลองพิมพ์คำว่า Sport ลงใน TextBox

คอนโทรลจะทำการค้นหา Product_Name ที่มีคำว่า Sport อยู่แล้วแสดงรายการ

ทั้งหมดที่ค้นหาเจอ เมื่อคลิกที่รายการที่ต้องการก็จะมาแสดงค่ายัง TextBox

บทที่ 3 เทคนิคในการสร้างคอนโทรล DateTimePicker สำหรับป้อนวันที่เวลา

บทนำ

คอนโทรล TextBox มีความยืดหยุ่นในการป้อนข้อมูลในรูปแบบต่างๆ แต่ก็จะเป็นค่าในรูปแบบสตริงค์เท่านั้น ในบทนี้จะนำคอนโทรล TextBox มาประยุกต์ใช้กับการป้อนข้อมูลวันที่และเวลาและสามารถนำค่าไปใช้ในรูปแบบตัวแปร DateTime ได้ทันที ซึ่งจะช่วยอำนวยความสะดวกในการนำไปใช้ในการป้อนข้อมูลในฐานะข้อมูลรูปแบบต่างๆ และประหยัดเวลาในการพัฒนาการเขียนโค้ดในการตรวจสอบรูปแบบของวันที่ได้ ในบทนี้จะมีการนำโค้ดของ JQuery เข้ามาผนวกกับการสร้างคอนโทรลของ ASP.NET

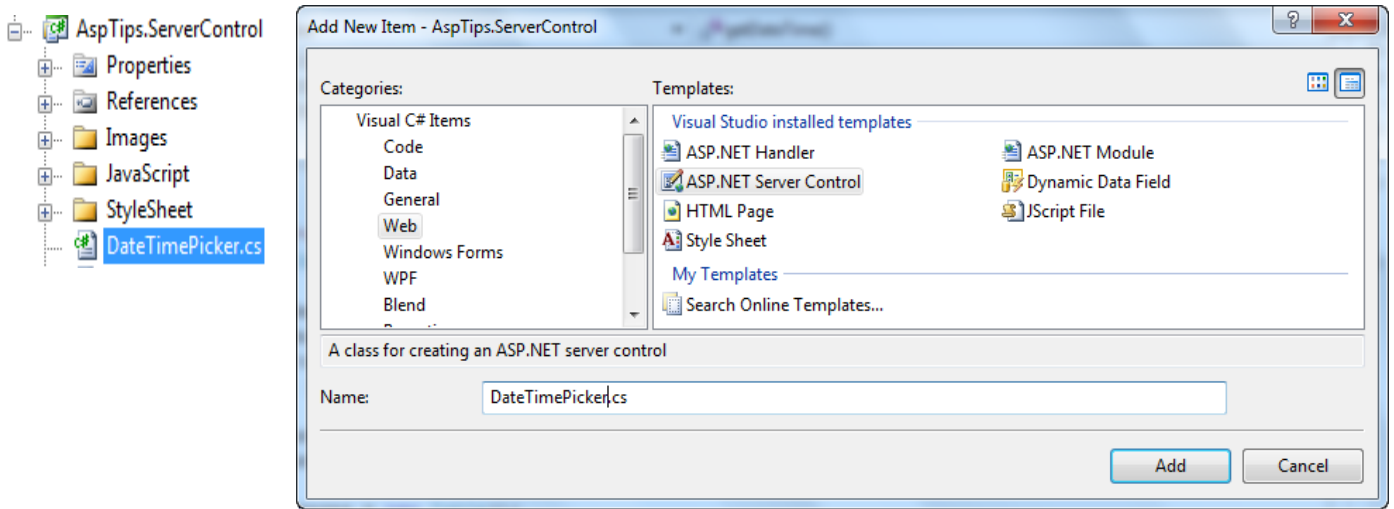


The image shows a date picker control. At the top, there is a text input field with a date format mask (e.g., "1-1-11"). Below it is a calendar grid for January 2011. The days of the week are labeled in Korean: 일 (Sun), 월 (Mon), 화 (Tue), 수 (Wed), 목 (Thu), 금 (Fri), 토 (Sat). The date 19 is highlighted in yellow.

1월 (JAN) 2011년						
일	월	화	수	목	금	토
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

ขั้นตอนสร้างคอนโทรล

1. สร้าง ASP.NET Server Control ชื่อ DateTimePicker.cs



2. สร้างสมาชิก ดังนี้

```
#region Members
```

```
Table tbMain = new Table();
TableRow tr = new TableRow();
TableCell []tc = new TableCell [2];
```

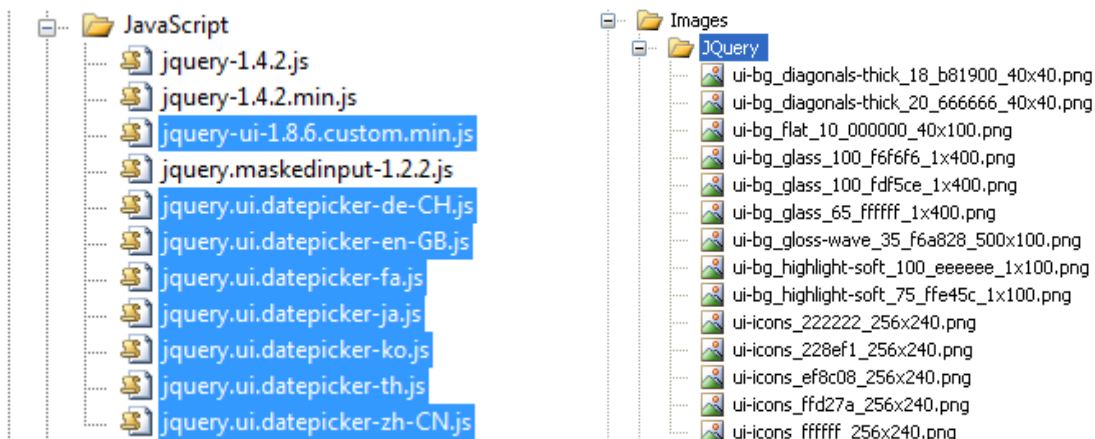


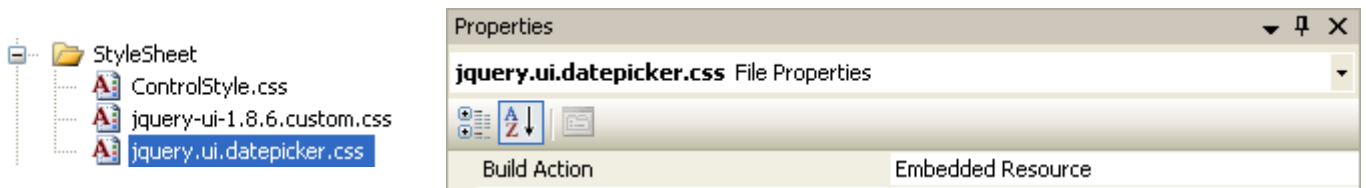
```
TextBox txtDate = new TextBox();
TextBox txtTime = new TextBox();
```

```
#endregion
```

คอนโทรลย่อยในคอนโทรล DateTimePicker นี้จะประกอบด้วย ตาราง 1 บรรทัด 2 เซลล์ และ TextBox สองอัน

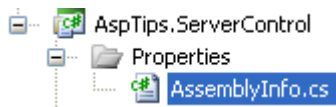
3. ดาวน์โหลด JavaScript , CSS และรูปภาพ สำหรับคอนโทรล DateTimePicker ที่เว็บไซต์ www.jquery.org





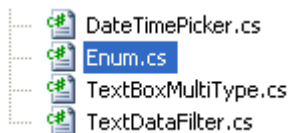
จากนั้นทำการกำหนดคุณสมบัติ Build Action ให้ไอเทมต่างๆ ให้เป็น Embedded Resource

4. ทำการลงทะเบียน Resource ต่างๆที่ใช้ในคอนโทรลในไฟล์ AssemblyInfo.cs



```
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.ui.datepicker-de-CH.js", "text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.ui.datepicker-en-GB.js", "text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.ui.datepicker-fa.js", "text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.ui.datepicker-ja.js", "text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.ui.datepicker-ko.js", "text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.ui.datepicker-th.js", "text/javascript")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.jquery.ui.datepicker-zh-CN.js", "text/javascript")]
// Style Sheet
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.StyleSheet.jquery.ui.datepicker.css", "text/css", PerformSubstitution = true)]
// JQuery Images
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_diagonals-thick_18_b81900_40x40.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_diagonals-thick_20_666666_40x40.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_flat_10_000000_40x100.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_glass_100_f6f6f6_1x400.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_glass_100_fdf5ce_1x400.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_glass_65_ffffff_1x400.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_gloss-wave_35_f6a828_500x100.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_highlight-soft_100_eeeeee_1x100.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-bg_highlight-soft_75_ffe45c_1x100.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-icons_222222_256x240.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-icons_228ef1_256x240.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-icons_ef8c08_256x240.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-icons_ffd27a_256x240.png", "image/png")]
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.Images.JQuery.ui-icons_ffffff_256x240.png", "image/png")]
```

5. เพิ่ม enum สำหรับ DateTimePicker



```
public enum DateTimePickerType
{
    Date
    , Time
    , Both
}
```

สำหรับกำหนดประเภทการแสดงผลวันที่
และเวลาของคอนโทรล

```
public enum DateTimeCulture
{
    Arabic,
    Default,
    German,
    British,
    Japanese,
    Korean,
    Thai,
    Chinese
}
```

สำหรับกำหนดรูปแบบการแสดงผลวันที่ของ
ประเทศต่างๆ

6. เพิ่มโค้ดกำหนดคุณสมบัติในไฟล์ DateTimePicker.cs

```
public DateTimePickerType DisplayType
{
    get
    {
        if (ViewState["DisplayType"] == null)
        {
            ViewState["DisplayType"] = DateTimePickerType.Date;
        }
        return (DateTimePickerType)ViewState["DisplayType"];
    }
    set { ViewState["DisplayType"] = value; }
}
```

สำหรับอ่านและกำหนดรูปแบบการแสดงผล
วันที่ของคอนโทรลที่ต้องการแสดง

```
public DateTimeCulture Culture
{
    get
    {
        if (ViewState["Culture"] == null)
        {
            ViewState["Culture"] = DateTimeCulture.Default;
        }

        return (DateTimeCulture)ViewState["Culture"];
    }
    set { ViewState["Culture"] = value; }
}
```

สำหรับอ่านและกำหนดรูปแบบการแสดงผลวันที่
ของคอนโทรลที่ต้องการแสดงตามรูปแบบ
ประเทศต่างๆที่กำหนด

```
public Unit DateWidth
{
    get {
        Unit u = new Unit(50, UnitType.Point);

        if (txtDate.Width != null)
        {
            u = txtDate.Width;
        }

        return u;
    }
    set
    {
        txtDate.Width = value;
    }
}
```

สำหรับอ่านและกำหนดขนาดความกว้าง
ของคอนโทรล TextBox วันที่

```

    }

    public Unit TimeWidth
    {
        get
        {
            Unit u = new Unit(20, UnitType.Point);

            if (txtTime.Width != null)
            {
                u = txtTime.Width;
            }

            return u;
        }
    }

```

สำหรับอ่านและกำหนดขนาดความกว้าง
ของคอนโทรล TextBox เวลา

```

        set
        {
            txtTime.Width = value;
        }
    }

```

สำหรับอ่านและกำหนดขนาดความสูง
ของคอนโทรล TextBox วันที่

```

    public Unit DateHeight
    {
        get
        {
            Unit u = new Unit(50, UnitType.Point);

            if (txtDate.Height != null)
            {
                u = txtDate.Height;
            }

            return u;
        }

        set
        {
            txtDate.Height = value;
        }
    }

```

สำหรับอ่านและกำหนดขนาดความสูง
ของคอนโทรล TextBox เวลา

```

    public Unit TimeHeight
    {
        get
        {
            Unit u = new Unit(50, UnitType.Point);

            if (txtTime.Height != null)
            {
                u = txtTime.Height;
            }

            return u;
        }

        set
        {
            txtTime.Height = value;
        }
    }

```

```

    }

    public override Unit Height
    {
        get
        {
            return txtDate.Height ;
        }
        set
        {
            txtDate.Height = value;
            txtTime.Height = value;
        }
    }

    public override Unit Width
    {
        get
        {
            return txtDate.Width;
        }
        set
        {
            txtDate.Width = value;
            txtTime.Width = value;
        }
    }

    public string DateText
    {
        get { return txtDate.Text; }
        set { txtDate.Text =value ; }
    }

    public string TimeText
    {
        get { return txtTime.Text; }
        set { txtTime.Text = value; }
    }

    public DateTime DateTimeValue
    {
        get { return getDateTime(); }
    }
    
```

สำหรับอ่านและกำหนดขนาดความสูง
ของคอนโทรล TextBox วันที่และเวลา

สำหรับอ่านและกำหนดขนาดความกว้าง
ของคอนโทรล TextBox วันที่และเวลา

สำหรับอ่านและกำหนดค่าในคอนโทรล
TextBox วันที่ในรูปแบบตัวอักษร

สำหรับอ่านและกำหนดค่าในคอนโทรล
TextBox เวลาในรูปแบบตัวอักษร

สำหรับอ่านวันที่และเวลาในรูปแบบ วันที่
และเวลา (DateTime)

7. เพิ่มฟังก์ชันในไฟล์ DateTimePicker.cs

```

protected override void OnInit(EventArgs e)
{
    this.EnsureChildControls();
}

protected override void Render(HtmlTextWriter writer)
{
    tbMain.RenderControl(writer);
}
    
```

คอนโทรลทุกอย่างจะอยู่ในตาราง tbMain
ทำการแสดงในรูปแบบ HTML เพื่อให้
browser อ่านและแสดงในรูปแบบการฟิสิกได้


```
protected override void OnPreRender (EventArgs e)
{
    base.OnPreRender (e);

    string sJavaScript =
@" AspTips.ServerControl.JavaScript.jquery-1.4
AspTips.ServerControl.JavaScript.jquery-1.4.2.min.js;
AspTips.ServerControl.JavaScript.jquery-ui-1.8.6.custom.min.js;
AspTips.ServerControl.JavaScript.jquery.ui.datepicker-de-CH.js;
AspTips.ServerControl.JavaScript.jquery.ui.datepicker-en-GB.js;
AspTips.ServerControl.JavaScript.jquery.ui.datepicker-fa.js;
AspTips.ServerControl.JavaScript.jquery.ui.datepicker-ja.js;
AspTips.ServerControl.JavaScript.jquery.ui.datepicker-ko.js;
AspTips.ServerControl.JavaScript.jquery.ui.datepicker-th.js;
AspTips.ServerControl.JavaScript.jquery.ui.datepicker-zh-CN.js;
AspTips.ServerControl.JavaScript.jquery.maskedinput-1.2.2.js";
    string[] JavaScripts = sJavaScript.Split(';');

    string sStyleSheet =
@" AspTips.ServerControl.StyleSheet.ControlStyle.css;
    AspTips.ServerControl.StyleSheet.jquery.ui.datepicker.css;
    AspTips.ServerControl.StyleSheet.jquery-ui-1.8.6.custom.css";

    string[] sStyleSheets = sStyleSheet.Split(';');

    ClientScriptManager cs = this.Page.ClientScript;

    foreach (string s in JavaScripts)
    {
        cs.RegisterClientScriptResource (this.GetType (), s.Trim () );
    }

    foreach (string s in sStyleSheets)
    {
        string includeLocation = cs.GetWebResourceUrl (this.GetType (), s.Trim () );
        HtmlLink cssLink = new HtmlLink ();
        cssLink.Href = includeLocation;
        cssLink.Attributes.Add ("rel", "stylesheet");
        cssLink.Attributes.Add ("type", "text/css");

        this.Page.Header.Controls.Add (cssLink);
    }
}

protected override void CreateChildControls ()
{
    // Create Table cells for Table Main.
    for (int i = 0; i < tc.Length; i++)
    {
        tc[i] = new TableCell ();
        tc[i].ID = UniqueID + "_tcDateTimePicker"+ i.ToString ();
    }

    //Add control to table.

    tc[0].Controls.Add (txtDate);
    tc[1].Controls.Add (txtTime);
}
```

ก่อนที่จะแสดงคอนโทรลทำการลงทะเบียน JavaScript และ CSS ในหน้าเว็บเพื่อให้คอนโทรลสามารถเรียกใช้ได้

ลงทะเบียน JavaScript

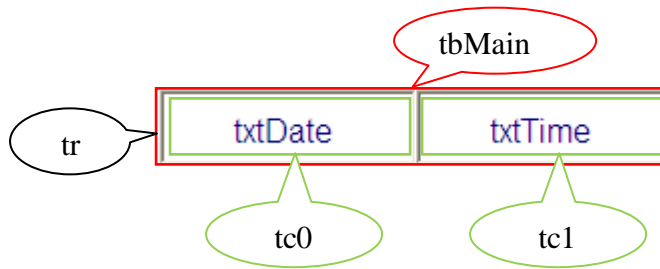
ลงทะเบียน CSS

ฟังก์ชันนี้จะเกิดเมื่อมีการสร้างคอนโทรลลูกขึ้น

สร้างเซลล์ขึ้นเพื่อทำการแทรกคอนโทรล TextBox วันที่และเวลาเข้าไป

```
tr.Cells.AddRange(tc);
```

```
tbMain.Rows.Add(tr);
```



```
// Set Style of Table.
```

```
tbMain.CellPadding = 0;
tbMain.CellSpacing = 0;
```

กำหนดค่าระยะห่างทางแนวตั้งและแนวนอนของเซลล์ในตารางเป็น 0 เพื่อไม่ให้เห็นขอบของตาราง

```
// Set Control's ID.
```

```
txtDate.ID = UniqueID + "_txtDate";
txtTime.ID = UniqueID + "_txtTime";
```

เขียนฟังก์ชันของ JQuery เพื่อทำการ mark up รูปแบบการป้อนวันที่

```
this.Controls.Add(tbMain);
```

```
ClientScriptManager cs = this.Page.ClientScript;
string MaskDateScript = @"jQuery(function($) { $('#" + txtDate.ClientID + "' ).mask('@MaskType'); });";
```

เขียนฟังก์ชันของ JQuery เพื่อทำการ mark up รูปแบบการป้อนเวลา

```
string MaskTimeScript = @"jQuery(function($) { $('#" + txtTime.ClientID + "' ).mask('99:99:99'); });";
```

```
string DatePickerScript = @" $(document).ready(function() { $('#" + txtDate.ClientID + @"').datepicker(); $.datepicker.setDefaults($.datepicker.regional['@Culture']); }); ";
```

เขียนฟังก์ชันของ JQuery เพื่อทำการกำหนดรูปแบบวันที่ของประเทศต่างๆ

```
switch (DisplayType)
{
    case DateTimePickerType.Date:
        txtTime.Visible = false;
        txtDate.Visible = true;

        break;
    case DateTimePickerType.Time:
        txtTime.Visible = true;
        txtDate.Visible = false ;

        cs.RegisterStartupScript(this.GetType(), "RegisTime" + txtTime.ClientID, MaskTimeScript, true);

        break;

    case DateTimePickerType.Both:
        txtTime.Visible = true;
```

คุณสมบัติ DisplayType จะกำหนดรูปแบบการแสดงคอนโทรลซึ่งมีสามแบบ คือ Date, Time และ Both

- Date แสดงเฉพาะวันที่
- Time แสดงเฉพาะเวลา
- Both แสดงทั้งวันที่และเวลา

```
cs.RegisterStartupScript(this.GetType(), "RegisTime" + txtTime.ClientID, MaskTimeScript, true);
```

```
break;
```

```
case DateTimePickerType.Both:
    txtTime.Visible = true;
```

```

txtDate.Visible = true ;

cs.RegisterStartupScript(this.GetType(), "RegisTime" + txtTime.ClientID,
MaskDateScript, true);

    break;
}
switch (Culture)
{
    case DateTimeCulture.Arabic:
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "9999/99/99"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "fa");
        break;

    case DateTimeCulture.Default :
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "99/99/9999"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "");
        break;
    case DateTimeCulture.British:
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "99/99/9999"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "en-GB");
        break;
    case DateTimeCulture.Chinese :
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "9999-99-99"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "zh-CN");
        break;

    case DateTimeCulture.German:
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "99.99.9999"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "de-CH");
        break;

    case DateTimeCulture.Japanese:
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "9999/99/99"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "ja");
        break;
    case DateTimeCulture.Korean:
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "9999-99-99"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "ko");
        break;
    case DateTimeCulture.Thai:
        cs.RegisterStartupScript(this.GetType(), "mask" + txtDate.ClientID,
MaskDateScript.Replace("@MaskType", "99/99/9999"), true);
        DatePickerScript = DatePickerScript.Replace("@Culture", "th");
        break;
}
cs.RegisterStartupScript(this.GetType(), "RegisCulture" + txtDate.ClientID,
DatePickerScript, true);
}

```

คุณสมบัติ Culture จะกำหนดรูปแบบการแสดงผลวันที่ในรูปแบบของท้องถิ่นต่างๆ

ฟังก์ชันสำหรับดึงค่าวันที่โดยแปลงค่าจาก TextBox ให้เป็นตัวแปรประเภทวันที่

```
private DateTime getDateFromTextBox()
{
    DateTime dt = DateTime.Now ;

    string[] Dates = txtDate.Text.Split('/', '.', '-', ':');
    string[] Times = txtTime.Text.Split(':');

    int iYear, iMonth, iDay;

    if ( Culture == DateTimeCulture.Arabic ||
        Culture == DateTimeCulture.Chinese ||
        Culture == DateTimeCulture.Japanese ||
        Culture == DateTimeCulture.Korean)
    {
        iYear = 0; iMonth = 1; iDay = 0;
    }
    else
    {
        iYear = 2; iMonth = 1; iDay = 0;
    }

    if (DisplayType == DateTimePickerType.Time)
    {
        dt = new DateTime(DateTime.Now.Year, DateTime.Now.Month,
            DateTime.Now.Day, int.Parse(Times[0]), int.Parse(Times[1]),
            int.Parse(Times[2]));
    }
    else if (DisplayType == DateTimePickerType.Both)
    {
        dt = new DateTime(int.Parse(Dates[iYear ]), int.Parse(Dates[iMonth ]),
            int.Parse(Dates[iDay ]), int.Parse(Times[0]), int.Parse(Times[1]),
            int.Parse(Times[2]));
    }
    else
    {
        dt = new DateTime(int.Parse(Dates[iYear ]), int.Parse(Dates[iMonth ]),
            int.Parse(Dates[iDay ]));
    }

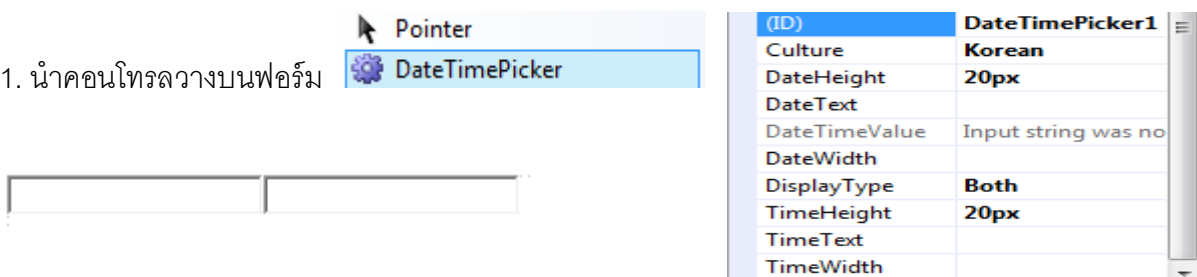
    return dt;
}
```

รูปแบบวันที่ของท้องถิ่นที่ระบุด้านบนจะขึ้นต้นด้วย ปี เดือน วัน เมื่อทำการตัดข้อความ



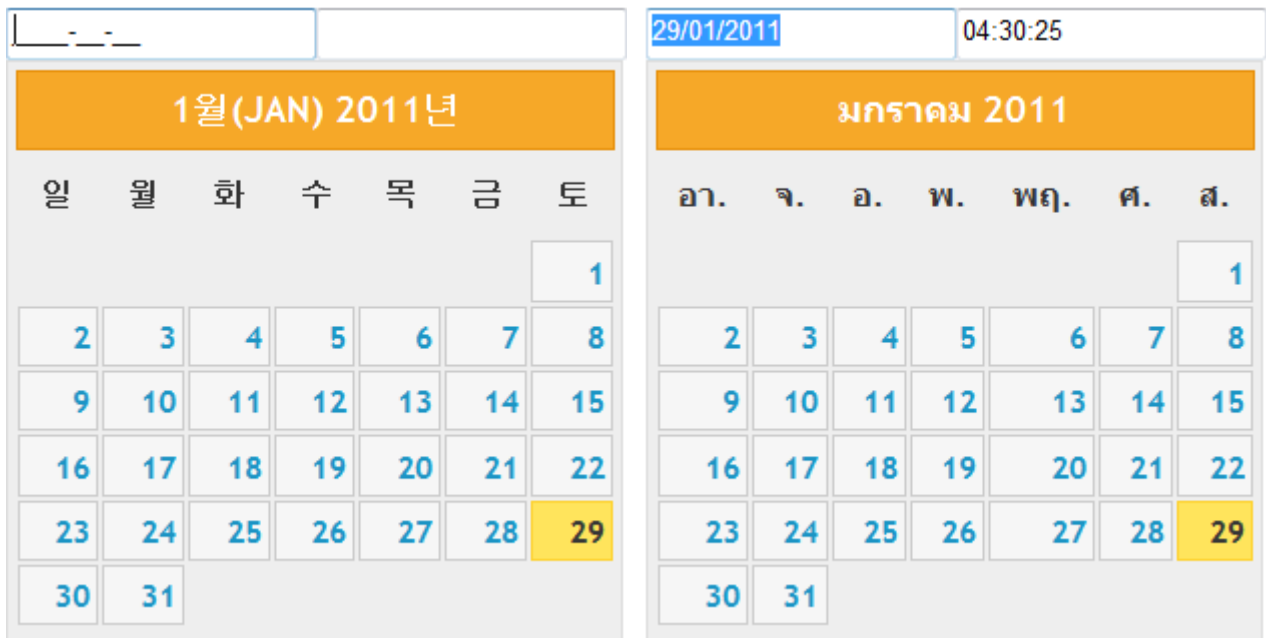
การนำคอนโทรลไปใช้งาน

1. นำคอนโทรลวางบนฟอร์ม

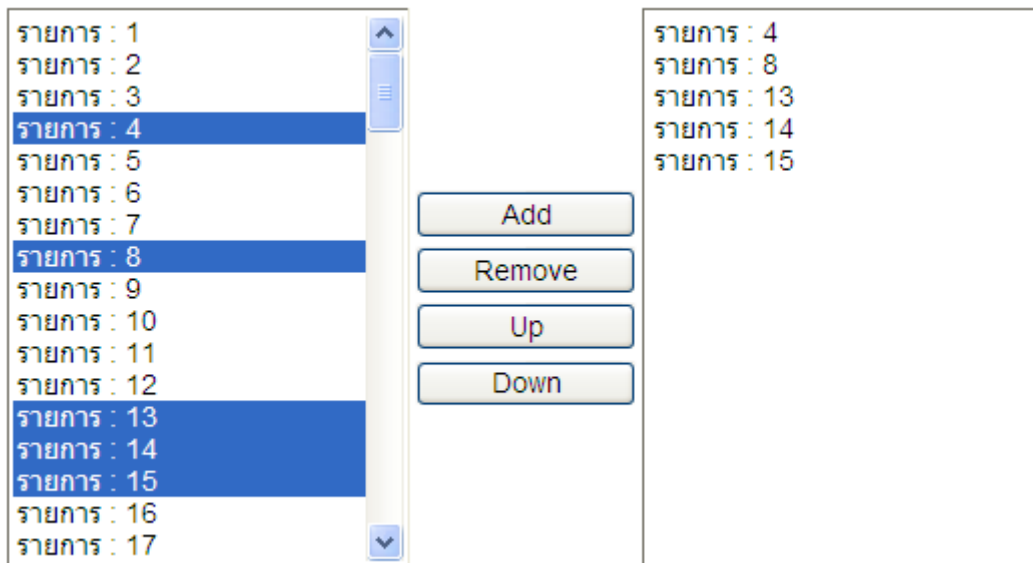


ลำดับ	คุณสมบัติ	ค่า
1	Culture	กำหนดรูปแบบการแสดงวันที่ตามท้องถิ่นที่ต้องการ
2	DateHeight	กำหนดความสูงของ TextBox วันที่
3	DateText	ตั้งหรือกำหนดค่า TextBox วันที่
4	DateWidth	กำหนดความกว้างของ TextBox วันที่
5	DisplayType	กำหนดรูปแบบการแสดงวันที่ เวลา หรือทั้งสองอย่าง
6	TimeHeight	กำหนดความสูงของ TextBox เวลา
7	TimeText	ตั้งหรือกำหนดค่า TextBox เวลา
8	TimeWidth	กำหนดความกว้างของ TextBox เวลา

2. กำหนดค่าคุณสมบัติต่างๆตามต้องการแล้วทดสอบโดยการรันเว็บ



บทที่ 5 เทคนิคในการสร้างคอนโทรล Swap List Box สำหรับเลือกรายการจากอันหนึ่งไปยังอีกอัน

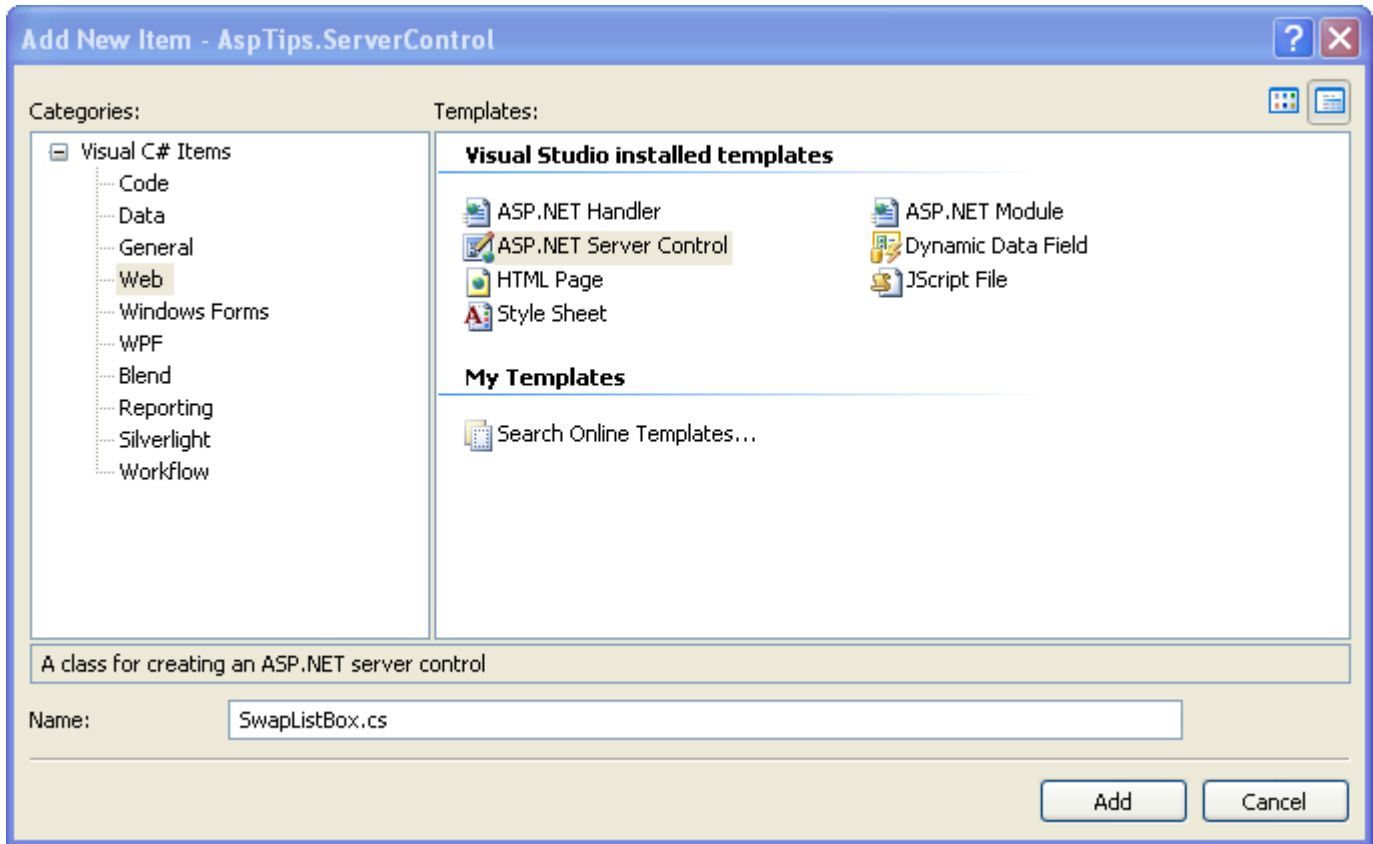


บทนำ

ในการเพิ่มข้อมูลที่ไม่มีควมจำเป็นต้องแสดงข้อมูลหลายๆคอลัมน์ เช่น การเพิ่มลบผู้ใช้งานในกลุ่มการใช้งานต่างๆ , การเลือกสินค้า, การเลือกรายการต่างๆ ฯลฯ โดยทั่วไปก็สามารถนำคอนโทรล ListBox มาเขียนเพื่อรองรับการใช้งานที่ต้องการได้ แต่หากการป้อนข้อมูลใน ListBox ในโปรแกรมของเรามีการใช้งานบ่อยๆ การสร้างคอนโทรลสำเร็จรูปก็จะช่วยอำนวยความสะดวกและลดเวลาในการพัฒนาได้

ขั้นตอนการสร้างคอนโทรล

1. สร้าง ASP.NET Server Control ชื่อ SwapListBox.cs



2. สร้างสมาชิก ดังนี้

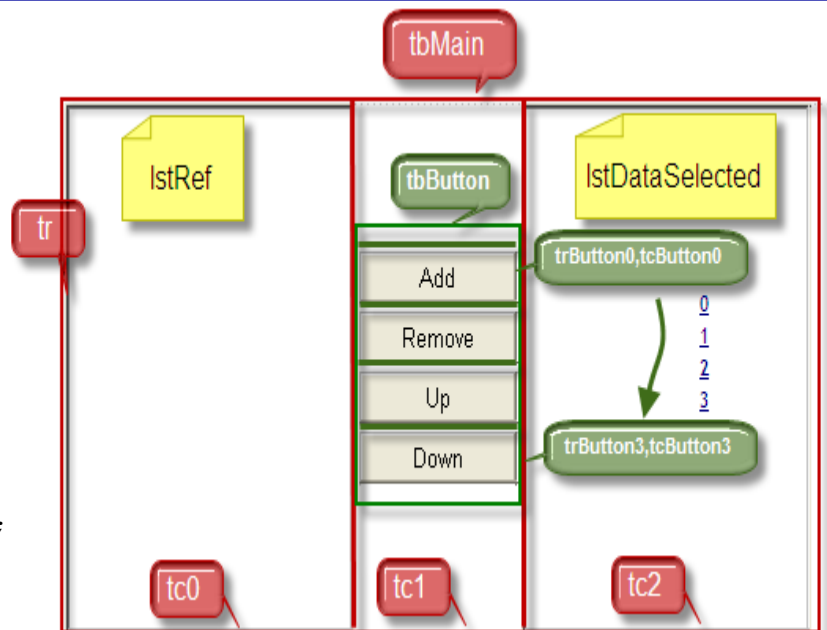
```
Table tbMain = new Table();
TableRow tr = new TableRow();
TableCell[] tc;

int iCellButton = 4, iCell = 3;

Table tbButton = new Table();
TableRow[] trButton;
TableCell[] tcButton;

Button btnAdd = new Button();
Button btnRemove = new Button();
Button btnUp = new Button();
Button btnDown = new Button();

ListBox lstRef = new ListBox();
ListBox lstDataSelected = new ListBox();
HiddenField hdd = new HiddenField();
```



3. เพิ่มคุณสมบัติ ดังนี้

```
public ListBox ListReference
{
    get { return lstRef; }
}
```

คุณสมบัติ ListReference จะคืนขอบเขต lstRef เพื่อให้สามารถกำหนดค่าคุณสมบัติที่มีอยู่ทั้งหมดใน lstRef ได้ แต่ค่าทั้งหมดสามารถกำหนดได้ในขณะรันเท่านั้น หากกำหนดไว้ในหน้าจ้ออกแบบจะไม่มีผลอะไร

```

public object DataSource
{
    get { return lstRef.DataSource; }
    set { lstRef.DataSource = value; }
}

public override Unit Height
{
    get
    {
        return base.Height;
    }
    set
    {
        base.Height = value;
        lstRef.Height = value;
        lstDataSelected.Height = value;
    }
}

public override Unit Width
{
    get
    {
        return base.Width;
    }
    set
    {
        base.Width = value;
        lstRef.Width = value;
        lstDataSelected.Width = value;
    }
}

public Unit ButtonWidth
{
    get { return (ViewState["ButtonWidth"]==null?new Unit(100,UnitType.Pixel) :
        (Unit)ViewState["ButtonWidth"]); }

    set
    {
        ViewState["ButtonWidth"] = value;
        btnAdd.Width = value;
        btnDown.Width = value;
        btnRemove.Width = value;
        btnUp.Width = value;
    }
}

public object SelectedDataSource
{
    get { return lstDataSelected.DataSource; }
}

```

คุณสมบัติ DataSource คืนและกำหนดค่า DataSource ให้กับคอนโทรล lstRef

คุณสมบัติ Height กำหนดค่าความสูงให้กับ คอนโทรล ListBox และตั้งค่าความสูงที่ถูกกำหนด

คุณสมบัติ Width กำหนดค่าความกว้างให้กับ คอนโทรล ListBox และตั้งค่าความกว้างที่ถูกกำหนด

คุณสมบัติ ButtonWidth กำหนดค่าความกว้างให้กับ คอนโทรล ปุ่ม Add,Down,Remove และ Up และตั้ง ค่าความกว้างที่ถูกกำหนด

คุณสมบัติ SelectedDataSource สำหรับคืนค่า DataSource ของ lstDataSelected

4. เพิ่มฟังก์ชันในเหตุการณ์ต่างๆ ดังนี้

```
protected override void Render(HtmlTextWriter writer)
```

```
{
    tbMain.RenderControl(writer);
}
```

เพิ่มฟังก์ชัน tbMain.RenderControl ใน
เหตุการณ์ Render สำหรับแสดงคอนโทรล
ในหน้าจอขณะออกแบบ

```
protected override void OnInit(EventArgs e)
```

```
{
    this.EnsureChildControls();
}
```

ฟังก์ชัน EnsureChildControls ในขณะเริ่มสร้าง
คอนโทรลตรวจสอบว่ามีการสร้างคอนโทรลลูก
แล้วหรือไม่

```
protected override void OnPreRender(EventArgs e)
```

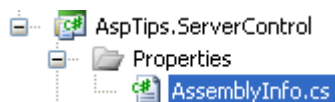
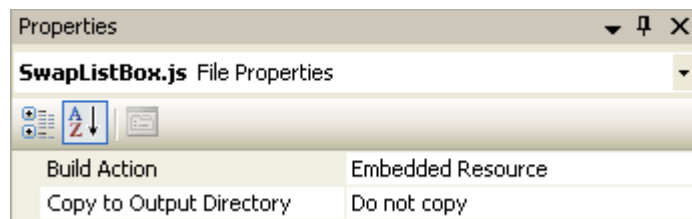
```
{
    base.OnPreRender(e);

    string sJavaScript = "AspTips.ServerControl.JavaScript.SwapListBox.js";
    ClientScriptManager cs = this.Page.ClientScript;
    cs.RegisterClientScriptResource(this.GetType(), sJavaScript);
}
```

ทำการลงทะเบียน JavaScript ก่อนที่จะแสดง
คอนโทรลในหน้าจอขณะออกแบบ

* อย่าลืมกำหนดคุณสมบัติของไฟล์ JavaScript ให้เป็นแบบ Embedded Resource และลงทะเบียนในไฟล์

AssemblyInfo.cs



```
[assembly: System.Web.UI.WebResource("AspTips.ServerControl.JavaScript.SwapListBox.js", "text/javascript")]
```

5. เพิ่มฟังก์ชันในจาวาสคริปต์ ดังนี้

```
function SetListBoxScrollPosition(lstId, hddId)
{
    var lst = document.getElementById(lstId);
    var hdd = document.getElementById(hddId);

    lst.focus();
    lst.scrollTop = hdd.value;
}
```

ฟังก์ชันนี้จะใช้สำหรับกำหนดตำแหน่งให้กับ Scrollbar เมื่อมีการสลับตำแหน่งของไอเท็มขึ้นหรือลง ตำแหน่ง Scrollbar จะเลื่อนไปยังตำแหน่ง 0 เสมอหากไม่มีการเก็บค่าตำแหน่งเดิมไว้

```
function GetListBoxScrollPosition(lstId, hddId)
{
    var lst = document.getElementById(lstId);
    var hdd = document.getElementById(hddId);

    hdd.value = lst.scrollTop;
}
```

ฟังก์ชันนี้จะใช้สำหรับเก็บค่าตำแหน่งของ Scrollbar ไว้ที่ Hidden Field

6. เพิ่มฟังก์ชันใน C# ดังนี้

```
protected override void CreateChildControls()
{
    tc = new TableCell[iCell];
    trButton = new TableRow[iCellButton];
    tcButton = new TableCell[iCellButton];

    for (int i = 0; i < trButton.Length; i++)
    {
        trButton[i] = new TableRow();
        tcButton[i] = new TableCell();
        trButton[i].Cells.Add(tcButton[i]);
        tbButton.Rows.Add(trButton[i]);

        if (i < tc.Length)
        {
            tc[i] = new TableCell();
            tr.Cells.Add(tc[i]);
        }
    }

    tcButton[0].Controls.Add(btnAdd);
    tcButton[1].Controls.Add(btnRemove);
    tcButton[2].Controls.Add(btnUp);
    tcButton[3].Controls.Add(btnDown);
    tcButton[3].Controls.Add(hdd);

    btnAdd.Width = ButtonWidth;
    btnDown.Width = btnAdd.Width;
    btnRemove.Width = btnAdd.Width;
    btnUp.Width = btnAdd.Width;

    tbMain.Rows.Add(tr);
}
```

ฟังก์ชัน CreateChildControls จะเริ่มสร้างคอนโทรลลูกก่อนที่จะทำการแสดงคอนโทรลในหน้าจ่ออกแบบ เราจะเริ่มทำการเพิ่มคอนโทรลต่างๆลงในตารางตามที่เรากำหนดไว้

เพิ่มเซลล์จำนวน 4 เซลล์ในแต่ละแถวของตาราง tbButton เพื่อทำการวางปุ่มลงในแต่ละเซลล์

เพิ่มปุ่มในเซลล์แต่ละเซลล์

```
tr.Cells[0].Controls.Add(lstRef);
tr.Cells[1].Controls.Add(tbButton );
tr.Cells[2].Controls.Add(lstDataSelected);
```

เพิ่ม ListBox และตาราง
ปุ่ม ในเซลล์ของ tbMain

```
lstDataSelected.SelectionMode = ListSelectionMode.Multiple;
lstRef.SelectionMode = ListSelectionMode.Multiple;
```

กำหนดคุณสมบัติ SelectionMode เป็นแบบ
Multiple เพื่อให้สามารถเลือกได้หลายไอเท็ม

```
lstRef.ID = UniqueID + "_lstRef" ;
lstDataSelected.ID = UniqueID + "_lstDataSelected";
tbMain.ID = UniqueID + "_tbMain";
hdd.ID = UniqueID + "_hdd";
```

กำหนดชื่อ ID ให้แก่คอนโทรล

```
tbMain.CellPadding = 0;
tbMain.CellSpacing = 0;
```

กำหนดความกว้างทางแนวตั้งและแนวนอน
ให้เป็น 0 เพื่อไม่ให้เห็นขอบของตาราง

```
btnAdd.Click +=new EventHandler(btnAdd_Click);
btnDown.Click += new EventHandler(btnDown_Click);
btnRemove.Click +=new EventHandler(btnRemove_Click);
btnUp.Click +=new EventHandler(btnUp_Click);
```

กำหนดฟังก์ชันให้กับเหตุการณ์เมื่อทำการ
คลิกที่ปุ่มต่างๆ

```
btnUp.Attributes.Add("onclick", "GetListBoxScrollPosition('" +
lstDataSelected.ClientID + "', '" + hdd.ClientID + "')");
btnDown.Attributes["onclick"] = btnUp.Attributes["onclick"];
```

```
this.Controls.Add(tbMain);
```

```
}
```

```
protected void btnAdd_Click(object sender, EventArgs e)
{
```

```
var lst = (from data in lstRef.Items.Cast<ListItem >()
where data.Selected select data);
```

```
foreach (ListItem a in lst)
```

เมื่อคลิกที่ปุ่ม Add เลือกเฉพาะไอเท็มที่ถูกเลือก

```
{
if (lstDataSelected.Items.FindByValue(a.Value) == null)
{
ListItem aItem = new ListItem(a.Text, a.Value);
lstDataSelected.Items.Add(aItem);
}
}
```

```

    }
}

```

ทำการเพิ่มเฉพาะไอเท็มที่ถูกเลือกลงใน ListBox

```

protected void btnRemove_Click(object sender, EventArgs e)
{
    for (int i = lstDataSelected.Items.Count - 1; i > -1; i--)
    {
        if (lstDataSelected.Items[i].Selected)
        {
            lstDataSelected.Items.RemoveAt(i);
        }
    }
}

```

ทำการลบรายการจากอันสุดท้ายก่อน เพราะหากทำการลบตามลำดับจากอันดับแรกไปอันดับสุดท้ายจะทำให้จำนวนไอเท็มลดลงแล้วไอเท็มที่ต้องการลบอยู่ตำแหน่งท้ายๆจะไม่มีทำให้เกิดข้อผิดพลาด

ตัวอย่าง มีไอเท็มอยู่ 3 อัน ต้องการลบ ไอเท็ม 0 และ 2 เมื่อทำการลบ ไอเท็มที่ 0 ออกไปก็จะทำให้เหลือ ไอเท็มทั้งหมด 2 อัน เมื่อถึงรอบการลบไอเท็มที่ 2 ก็เกิดข้อผิดพลาดขึ้นเนื่องจากไอเท็มที่ 2 ไม่มีแล้ว

```

protected void btnUp_Click(object sender, EventArgs e)
{
    int iLastPosition = 0;

    for (int i = 0; i < lstDataSelected.Items.Count; i++)
    {
        if (lstDataSelected.Items[i].Selected)
        {
            if (i > 0 && !lstDataSelected.Items[i - 1].Selected)
            {
                ListItem bottom = lstDataSelected.Items[i];
                lstDataSelected.Items.Remove(bottom);
                lstDataSelected.Items.Insert(i - 1, bottom);
                lstDataSelected.Items[i - 1].Selected = true;
                iLastPosition = i - 1;
            }
        }
    }
}

```

เมื่อกดปุ่ม Up ทำการย้ายรายการที่ถูกเลือกขึ้นไปหนึ่งตำแหน่ง โดยมีเงื่อนไขว่าตำแหน่งของไอเท็มจะต้องไม่ใช่ตำแหน่งที่ 0 และตำแหน่งที่อยู่สูงกว่าหนึ่งตำแหน่งจะต้องไม่ถูกเลือกอยู่

```

ScriptManager.RegisterStartupScript(this.Page, this.Page.GetType(),
    "SetScrollPosition", "SetListBoxScrollPosition('" + lstDataSelected.ClientID
    + "', '" + hdd.ClientID + "')";", true);
}

```

ทำการกำหนดตำแหน่ง Scrollbar ให้อยู่ตำแหน่งเดิมเมื่อมีการปรับตำแหน่งไอเท็ม

```
protected void btnDown_Click(object sender, EventArgs e)
{
    int iLastPosition = 0;
    int startindex = lstDataSelected.Items.Count - 1;

    for (int i = startindex; i > -1; i--)
    {
        if (lstDataSelected.Items[i].Selected

            if (i < startindex && !lstDataSelected.Items[i + 1].Selected)
            {
                ListItem bottom = lstDataSelected.Items[i];
                lstDataSelected.Items.Remove(bottom); lstDataSelected.Items.Insert
                (i + 1, bottom);
                lstDataSelected.Items[i + 1].Selected = true;
                iLastPosition = i;
            }
        }
    }
}
```

เมื่อกดปุ่ม Down ทำการย้ายรายการที่ถูกเลือกลงมาหนึ่งตำแหน่ง โดยมีเงื่อนไขว่าตำแหน่งของไอเท็มจะต้องน้อยกว่าตำแหน่งที่ startindex และตำแหน่งที่อยู่ต่ำกว่าหนึ่งตำแหน่งจะต้องไม่ถูกเลือกอยู่

```
ScriptManager.RegisterStartupScript(this.Page, this.Page.GetType(),
    "SetScrollPosition", "SetListBoxScrollPosition('" + lstDataSelected.ClientID +
    "','" + hdd.ClientID + "')";", true);
}

public void BindData()
{
    lstRef.DataBind();
}
```

ทำการเชื่อมข้อมูลให้กับ lstRef

การนำคอนโทรลไปใช้งาน

1. วางคอนโทรลลงบนฟอร์ม

เนื่องจากคอนโทรลที่สร้างมีการเรียกใช้ ScriptManager จึงต้องวางคอนโทรลไว้ใน UpdatePanel

2. กำหนดคุณสมบัติของคอนโทรลเบื้องต้นตามความเหมาะสม ดังนี้

ลำดับ	คุณสมบัติ	ค่า
1	Width	กำหนดค่าความกว้างของ ListBox เป็น 200 px
2	ButtonWidth	กำหนดความสูงของ TextBox วันที่
3	DateText	ตั้งหรือกำหนดค่า TextBox วันที่
4	DateWidth	กำหนดความกว้างของ TextBox วันที่
5	DisplayType	กำหนดรูปแบบการแสดงผลวันที่ เวลา หรือทั้งสองอย่าง
6	TimeHeight	กำหนดความสูงของ TextBox เวลา
7	TimeText	ตั้งหรือกำหนดค่า TextBox เวลา
8	TimeWidth	กำหนดความกว้างของ TextBox เวลา