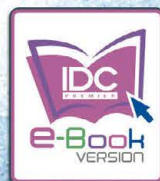




หนังสือคอมพิวเตอร์ที่มีผู้อ่านมากที่สุด



ศุภชัย สมพานิช



จากผู้แต่ง
หนังสือ
ASP.NET
ที่มียอดขายสูงที่สุด
ของไทย

BASIC ASP.NET 4.0



ดาวน์โหลด Free CD ได้ที่
www.infopress.co.th



Source Code

.NET Framework 4.0,
SQL Server 2008 R2
Management Studio Express
และ: SQL Server 2008 R2
Express Edition with Tools

Infopress
Group



10th ANNIVERSARY BEST COMPUTER BOOK

Everything IT, Everything we're expert.

BASIC ASP.NET 4.0

ผู้แต่ง

บรรณาธิการ

ออกแบบปก

ออกแบบและจัดรูปเล่ม

พิสูจน์อักษร

ประสานงานการผลิต

ศุภชัย สมพานิช

กิตินันท์ พลสวัสดิ์

อนัน วาโษะ

วุฒิพันธ์ สมพระเมฆ, วันวิสาข์ แซ่ลิ่ม

สุนทรี บรรลือศักดิ์, มนฤดี ศรีอุทโยภาส

สุพัตรา อาจปรุ, ฉัตรชนก แก้วจันทร์

thaivb@yahoo.com

kitinan@infopress.co.th

Visual Studio 2010 เป็นเครื่องหมายการค้าของบริษัท Microsoft Inc. และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้เทคโนโลยีสารสนเทศทั้งในระดับพื้นฐานและระดับสูง เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน โดยเฉพาะงานที่เกี่ยวข้องกับสารสนเทศ ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ editor@infopress.co.th หรือโทรศัพท์หมายเลข 0-2962-1081 (อัตรานาที 10 คู่สาย) โทรสาร 0-2962-1084

ข้อมูลทางบรรณานุกรม



ISBN

พิมพ์ครั้งที่ 1

จัดพิมพ์และจัดจำหน่ายโดย



ศุภชัย สมพานิช

BASIC ASP.NET 4.0

นนทบุรี : ไอดีซี, 2553

480 หน้า

1. วิชวลสตูดิโอ (โปรแกรมคอมพิวเตอร์)

2. การเขียนโปรแกรม I ชื่อเรื่อง

005.11

978-616-200-243-4

พฤษภาคม 2555

บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 5 ห้อง 516 อาคารจัสมินอินเตอร์เนชั่นแนลทาวเวอร์

ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120

โทรศัพท์ 0-2962-1081 (อัตรานาที 10 คู่สาย) โทรสาร 0-2962-1084

ราคา 220 บาท

สำหรับลูกค้า

ลูกค้าสัมพันธ์

โทรศัพท์ 0-2962-1081 ต่อ 121

โทรสาร 0-2962-1084

สำหรับร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2951-1300-2,

0-2591-9446

โทรสาร 0-2591-9447

บทบรรณาธิการ

หากจะกล่าวถึงการพัฒนาเว็บไซต์ในโลกออนไลน์ปัจจุบัน จะเห็นได้ว่ามีเครื่องมือที่ช่วยให้เราสามารถพัฒนาเว็บมีมากมาย และคงไม่มีใครปฏิเสธว่า Microsoft Visual Studio เป็นเครื่องมือที่มีความสะดวกในการใช้งาน และมีขีดความสามารถที่ไม่เป็นรองเครื่องมือใดๆ จึงทำให้ Microsoft Visual Studio ได้รับความนิยมในระดับต้นๆ ของเครื่องมือที่ใช้พัฒนาเว็บไซต์ที่มีอยู่ซึ่งปัจจุบันเป็นเวอร์ชัน Microsoft Visual Studio 2010 ซึ่งมาพร้อมกับ ASP.NET 4.0

เว็บไซต์ที่พัฒนาด้วย ASP.NET 4.0 สามารถเลือกเขียนโค้ดกำหนดการทำงานได้ 2 ภาษา คือ Visual Basic 2010 และ Visual C# 2010 ซึ่งให้ผลการทำงานเหมือนกัน ขึ้นอยู่กับว่าผู้อ่านจะมีความถนัดหรือมีความสนใจในภาษาใด อีกทั้งผู้อ่านท่านใดที่มีประสบการณ์ในการใช้งาน ASP ในเวอร์ชันก่อนหน้านี้อาจสามารถปรับใช้กับ ASP.NET 4.0 ได้เช่นกัน เนื่องจากโครงสร้างของภาษามีความแตกต่างกันไม่มากนัก โดยที่ ASP.NET 4.0 จะเพิ่มขีดความสามารถให้มากกว่าเวอร์ชันก่อนเพื่อให้ง่ายสำหรับผู้อ่านมากยิ่งขึ้น

หนังสือเล่มนี้ได้กล่าวถึงหลักการพัฒนาเว็บไซต์ที่มีความสำคัญสำหรับผู้ที่ต้องการใช้งาน ASP.NET 4.0 รวมถึงพื้นฐานต่างๆ ที่ผู้อ่านควรรู้ พร้อมกับ Workshops มากมายให้ผู้อ่านได้เรียนรู้และทดลองทำ ซึ่งเราได้นำเสนอไว้ทั้ง 2 ภาษาคือ Visual Basic 2010 และ Visual C# 2010

ทางทีมงานหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะมีส่วนช่วยให้ผู้อ่านมีความเข้าใจและสามารถพัฒนาเว็บไซต์ด้วย ASP.NET 4.0 ได้เป็นอย่างดี หากพบข้อผิดพลาดประการใดสามารถติชมกลับมาทางทีมงาน เพื่อให้เราได้ปรับปรุงผลงานให้มีประสิทธิภาพมากยิ่งขึ้น

ทีมงาน DEV BOOK ขอขอบคุณท่านผู้อ่านทุกท่านที่ให้ความสนับสนุนเราในการสร้างสรรค์ผลงานเขียนที่ดีมีคุณภาพ

ด้วยความเคารพอย่างสูง

กิตินันท์ พลสวัสดิ์

คำนำ

หนังสือ BASIC ASP.NET 4.0 เล่มนี้ เป็นเล่มแรกของหนังสือในซีรีส์พัฒนาเว็บแอปพลิเคชันด้วย ASP.NET 4.0 นำเสนอเนื้อหาสำหรับคุณผู้อ่านที่เป็นมือใหม่ในโลกของ .NET 4.0 (Visual Studio 2010) คลอบคลุมไปจนกระทั่งถึงการนำเสนอตัวอย่างการสร้างเว็บไซต์แบบง่ายๆ โดยใช้เวลาการเรียนรู้ไม่นานนัก

หนังสือเล่มนี้เป็นคู่มือแนวทางได้ในระดับหนึ่งว่า สิ่งที่ต้องรู้เป็นลำดับแรกๆ ของการสร้างเว็บไซต์ขึ้นมาด้วย ASP.NET 4.0 ว่ามีอะไรบ้าง ซึ่งผู้เขียนคิดว่าเป็นคำถามที่หลายๆ คนต้องการทราบ แน่นอนว่า หนังสือเพียง 1 เล่มไม่สามารถนำเสนอได้อย่างครบถ้วน แต่อย่างน้อยที่สุดถ้ายึดถือจุดที่ว่า นำเสนอเท่าที่จำเป็นเท่าที่ควรทราบ เท่าที่ต้องรู้ ก็น่าจะเป็นหนทางเริ่มต้นที่ดี และยั่งยืน

ท้ายที่สุดนี้ผู้เขียนขอขอบคุณคุณผู้อ่านทุกท่านที่ให้ความสนใจผลงานของผู้เขียน รวมไปถึงได้ให้ความกรุณาสอบถาม แนะนำ ตีติงจากผลงานที่ผ่านมาจากทุกช่องทาง ไม่ว่าจะเป็นอีเมล, MSN หรือหน้าเว็บบอร์ดของเว็บไซต์สำนักพิมพ์ ผู้เขียนต้องขอขอบคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วยครับ

และที่ลืมไม่ได้เลยก็คือ ผู้เขียนขอขอบคุณ คุณสัจจะ จรัสรุ่งรวีร ที่ได้ให้โอกาสผู้เขียนจัดทำหนังสือในซีรีส์ .NET 4.0 ทั้ง Windows และ Web Application อีกครั้งหนึ่งครับ

ศุภชัย สมพานิช

thaivb@yahoo.com (MSN),

thaivb.net@gmail.com

PART 1 พื้นฐาน ASP.NET 4.0

บทที่ 1 ก้าวแรกสู่ ASP.NET 4.0

แนวทางการนำเสนอของซีรีส์หนังสือ ASP.NET 4.0	3
วิธีการนำเสนอของหนังสือเล่มนี้	4
กว่าจะมาเป็น Web 2.0	5
ยุคที่ 1 Web 1.0 (Client Side Script)	5
ยุคที่ 2 Web 1.5 (Server Side Script)	6
ยุคที่ 3 Web 2.0 (AJAX)	7
ยุคที่ 4 Web 2.0 (Rich Internet Application-RIA)	7
ภาษา ASP.NET คืออะไร	8
การกำหนดให้ VS 2010 แสดงไดอะล็อกบ็อกซ์ Show New Project dialog box	9
การกำหนดให้ Editor ของ VS 2010 แสดงหมายเลขบรรทัด	11
ทำความรู้จักกับโปรเจกต์ด้าน Web ของ VS 2010	12
ทำความรู้จักกับโปรเจกต์ชนิด ASP.NET Empty Web Application	13
WebMatrix Platform	15
ทำความรู้จักกับ VS 2010 IDE แบบ Web Application	18
การกำหนดเว็บฟอร์มเริ่มต้น	19
ทำความรู้จักกับโครงสร้างเบื้องต้นของโปรเจกต์ชนิด ASP.NET 4.0	20
วิธีการทดสอบตัวอย่างสคริปต์ของ ASP.NET 4.0	23
ความสัมพันธ์ระหว่างเว็บคอนโทรลของ ASP.NET กับภาษา HTML	26

บทที่ 2 พื้นฐานการเขียนโปรแกรมด้วยภาษา VB 2010 และ VC# 2010

การกำหนดคำสั่งเพิ่มเติม (เฉพาะภาษา VB 2010 เท่านั้น)	27
การเขียนโปรแกรมแบบรองรับเหตุการณ์	28
การประกาศตัวแปร	30
ขอบเขตของตัวแปร	33
การใช้งานคำสั่ง If...Then...Else (if-else)	35
การใช้งานคำสั่ง Select Case (switch-case)	40
การใช้งานคำสั่ง For...Next (for)	42
พื้นฐานการใช้งาน ASP.NET ร่วมกับอิลิเมนต์ของภาษา HTML	44
การเขียนโค้ดควบคุมอิลิเมนต์ของ HTML	45

การใช้งานคำสั่ง Do While...Loop (while)	47
การใช้งานคำสั่ง Do...Loop While (do-while)	48
การใช้ตัวดำเนินการ AndAlso (&&) และตัวดำเนินการ OrElse ()	49
การสร้างฟังก์ชันขึ้นมาใช้งาน.....	50
เนมสเปซ (Namespaces) คืออะไร.....	52

ตอนที่ 3 การทำงานกับข้อมูลและสถานะของข้อมูล

ลำดับเหตุการณ์ของออบเจกต์ Page.....	55
ทำความเข้าใจกับคุณสมบัติ IsPostBack ของออบเจกต์ Page	57
การกำหนดให้เว็บคอนโทรลมีการส่งค่ากลับ	59
การ Trace หน้าเว็บเพจ.....	61
พื้นฐานการใช้งาน Cookies	63
การใช้งาน Cookies แบบมี Keys	66
การแสดง Cookies ที่มีอยู่ทั้งหมด	70
การลบ Cookies.....	71
การทำคำทักทายผู้เข้าเยี่ยมชมเว็บเพจ.....	72
พื้นฐานการใช้งานตัวแปรแบบ Session	77
พื้นฐานการใช้งานตัวแปรแบบ Application	84
การตรวจสอบจำนวนคนออนไลน์.....	86
ทำความเข้าใจกับ ViewState	92
การเก็บ Object ลงใน ViewState	94
การใช้งาน Session State แบบ Cookieless	99

ตอนที่ 4 การใช้งานเว็บคอนโทรลกลุ่ม Navigation

การใช้งานเว็บคอนโทรล Menu	105
การใช้งานเว็บคอนโทรล TreeView	108
การใช้งานเว็บคอนโทรล TreeView แบบมี CheckBox.....	112
การใช้งาน Site Map	114
การใช้งาน URL Mappings	116

บทที่ 5 การใช้งาน Themes และ Skin

พื้นฐานการใช้งาน CSS StyleSheet ร่วมกับเว็บคอนโทรล	119
พื้นฐานการใช้งานธีม	124
การกำหนดให้เว็บเพจเรียกใช้ธีมโดยตรง	124
เขียนโค้ดเพื่อกำหนดธีมให้กับเว็บเพจ	136
การใช้งานธีมร่วมกับคุณสมบัติ AutoPostBack	143

บทที่ 6 การใช้งานเว็บคอนโทรลกลุ่ม Validation

การใช้งานเว็บคอนโทรล RequiredFieldValidator	145
การใช้งานเว็บคอนโทรล CompareValidator	148
การใช้งานคุณสมบัติ ControlToCompare	150
การใช้งานคุณสมบัติ ValueToCompare	151
การใช้งานเว็บคอนโทรล RegularExpressionValidator	152
การใช้งานเว็บคอนโทรล RangeValidator	153
การใช้งานเว็บคอนโทรล CustomValidator	154
การใช้งานเว็บคอนโทรล ValidationSummary	156

บทที่ 7 พื้นฐานการใช้งานระบบฐานข้อมูล SQL Server 2008 SP1 Express Edition

การเตรียมสภาพแวดล้อมให้พร้อมใช้งาน	159
SQL Server 2008 R2 Express Edition With Tools	161
พื้นฐานการใช้งานฐานข้อมูล SQL Server 2008 SP1 Express Edition	162
การสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูล SQL Server 2008 SP1	164
การ Backup และ Restore ฐานข้อมูล SQL Server 2008 SP1	168
การ Backup ฐานข้อมูล SQL Server 2008 SP1	168
การ Restore ฐานข้อมูล SQL Server 2008 SP1	170
การยกเลิกป้องกันการแก้ไขโครงสร้างตาราง	174
การเพิ่มฐานข้อมูล Northwind เข้าไปใน SQL Server 2008 SP1 Express Edition	175
การใช้งานฐานข้อมูล SQL Server 2008 SP1 ในระบบไฟล์	176
การติดตั้ง SQL Server Database File เข้าสู่การใช้งานตามปกติ	179
การนำฐานข้อมูล SQL Server 2008 SP1 ไปใช้แบบระบบไฟล์	182

บทที่ 8 พื้นฐานใช้งาน ASP.NET 4.0 กับฐานข้อมูล

พื้นฐานการแสดงผลข้อมูลด้วยเว็บคอนโทรล GridView โดยการเขียนโค้ด	187
การเขียนโค้ดเรียกใช้งานกลุ่มออบเจกต์ ADO.NET	187
การใช้งานเว็บคอนโทรล SqlDataSource	192
การใช้งานฐานข้อมูล SQL Server ในระบบไฟล์	199
การใช้งานเว็บคอนโทรล SqlDataSource ในโหมด DataReader	202
การสร้างระบบ Login อย่างง่าย	202

บทที่ 9 การใช้งานเว็บคอนโทรล GridView และเว็บคอนโทรลอื่นๆ ร่วมกับระบบฐานข้อมูล

การแบ่งแสดงข้อมูลออกเป็นหน้าและการเรียงลำดับข้อมูล	211
การแบ่งแสดงข้อมูลออกเป็นหน้าและการเรียงลำดับข้อมูลของฝั่ง Client	216
การใช้งานคุณสมบัติ ShowFooter ของเว็บคอนโทรล GridView	218
การลงสีของแต่ละแถวของ GridView โดยการตรวจสอบค่าที่แสดงอยู่	222
การลงสีที่ Header และ Footer ของ GridView	225
การแก้ไขข้อมูลผ่านเว็บคอนโทรล GridView	228
การใช้งานเว็บคอนโทรล DetailsView	234
การแสดงผลข้อมูลแบบ Master-Details	237
การทำงานกับค่า Null	242
การใช้งานคอลัมน์ชนิด Select ของเว็บคอนโทรล GridView แบบมีเงื่อนไข	243
การค้นหาข้อมูลผ่านเว็บคอนโทรล SqlDataSource	248
การส่งออบเจกต์ DataSet ระหว่างหน้าเว็บเพจ	250
การแสดงผลข้อมูลแบบ Cascade	257
การสร้างคอลัมน์แบบ DropDownList ในเว็บคอนโทรล GridView	261

บทที่ 10 LINQ to SQL กับ ASP.NET 4.0

พื้นฐานการใช้งาน LINQ to SQL ใน Web Application	272
การแสดงผลข้อมูลด้วย LINQ to SQL โดยการเขียนโค้ด	281
การแสดงผลข้อมูลด้วย LINQ to SQL โดยการเขียนโค้ดแบบ 2 ตารางขึ้นไป	287
การใช้งานคำสั่ง Join	292
การเขียน LINQ to SQL โดยอาศัยไวยากรณ์แบบ Lambda Expression	294
การใช้งาน LINQ to SQL ไวยากรณ์แบบผสม	296

Anonymous Type กับ LINQ to SQL.....	298
พื้นฐานการเพิ่มและแก้ไขข้อมูลด้วย LINQ to SQL.....	304

บทที่ 11 SQL และ LINQ to SQL Language

การคิวรีข้อมูลระหว่างภาษา SQL กับภาษา LINQ to SQL.....	314
แสดงข้อมูลโดยไม่มีเงื่อนไข.....	314
แสดงข้อมูลเฉพาะฟิลด์ที่ต้องการ.....	315
การเรียงลำดับข้อมูล.....	316
การแสดงผลข้อมูลตั้งแต่ 2 ตารางขึ้นไป.....	317
แสดงข้อมูลโดยกำหนดเงื่อนไขเฉพาะอักขระตัวแรก.....	318
แสดงข้อมูลโดยกำหนดเงื่อนไขเฉพาะบางส่วนของข้อมูล.....	319
แสดงข้อมูลโดยกำหนดจำนวนข้อมูลที่สนใจ.....	320
แสดงข้อมูลโดยกำหนดเงื่อนไขมากกว่า 1 เงื่อนไขด้วยตัวดำเนินการ AND.....	321
แสดงข้อมูลโดยกำหนดเงื่อนไขแบบช่วงข้อมูล.....	321
แสดงข้อมูลโดยกำหนดเงื่อนไขด้วยฟิลด์ข้อมูล.....	323
แสดงข้อมูลโดยกำหนดเงื่อนไขมากกว่า 1 เงื่อนไขด้วยตัวดำเนินการ Or.....	323
แสดงข้อมูลจากชุดเงื่อนไขที่กำหนดไว้.....	324
แสดงข้อมูลผลรวมโดยใช้คำสั่ง SUM().....	325
แสดงข้อมูลค่าเฉลี่ยโดยใช้คำสั่ง AVG().....	326
แสดงข้อมูลการนับจำนวนโดยใช้คำสั่ง Count().....	326
แสดงข้อมูลผลรวมแต่ละรายการโดยใช้คำสั่ง SUM ().....	328
แสดงข้อมูลที่กำหนดเงื่อนไขให้กับข้อมูลที่ถูกรวบรวม.....	329

บทที่ 12 Master Page

ทำความเข้าใจกับ Master Page.....	331
ทำความเข้าใจกับโปรเจกต์ชนิด ASP.NET Web Application.....	333
การเพิ่มเว็บเพจอื่นๆ เข้ามาในโปรเจกต์ชนิด ASP.NET Web Application.....	335
ข้อแตกต่างระหว่างเว็บเพจปกติกับ Content Page.....	337
การควบคุมอิลิเมนต์ <title>...</title> ในไฟล์ Master Page.....	340
การเพิ่มส่วน Footer.....	342
การใช้งาน Sitemap ร่วมกับ Master Page.....	344
ระบบ Login กับ Master Page.....	347

PART 2 ASP.NET 4.0 Workshops

ตอนที่ 13 Workshop สมุดเยี่ยมชม (GuestBook)

การสร้างสมุดเยี่ยมชม.....	373
ขั้นตอนการสร้างสมุดเยี่ยมชม	375

ตอนที่ 14 Workshop กระดานข่าว (Webboard)

การใช้งาน Webboard.....	383
ขั้นตอนการสร้าง Webboard	387
เขียนโค้ดกำหนดการทำงานของ Webboard	406

ตอนที่ 15 Workshop การทำแบบสำรวจ (Poll)

การใช้งาน Poll (แบบสำรวจความคิดเห็น).....	425
ขั้นตอนการสร้าง Poll	430

PART 1

พื้นฐาน ASP.NET 4.0

- บทที่ 1 ก้าวแรกสู่ ASP.NET 4.0
- บทที่ 2 พื้นฐานการเขียนโปรแกรมด้วยภาษา VB 2010 และ VC#2010
- บทที่ 3 การทำงานกับข้อมูลและสถานะของข้อมูล
- บทที่ 4 การใช้งานเว็บคอนโทรลกลุ่ม Navigation
- บทที่ 5 การใช้งาน Themes และ Skin
- บทที่ 6 การใช้งานเว็บคอนโทรลกลุ่ม Validation
- บทที่ 7 พื้นฐานการใช้งานระบบฐานข้อมูล SQL Sever 2008 SP1 Express Edition
- บทที่ 8 พื้นฐานใช้งาน ASP.NET 4.0 ด้านฐานข้อมูล
- บทที่ 9 การใช้งานเว็บคอนโทรล GridView และเว็บคอนโทรลอื่นๆ
ร่วมกับระบบฐานข้อมูล
- บทที่ 10 LINQ to SQL กับ ASP.NET 4.0
- บทที่ 11 SQL และ LINQ to SQL Language
- บทที่ 12 Master Page

Visual Studio 2010



ก้าวแรกสู่ ASP.NET 4.0

หนังสือ BASIC ASP.NET 4.0 เล่มนี้เป็นหนังสือเล่มแรกในซีรีส์ของการพัฒนา Web Application ด้วย ASP.NET 4.0 สำหรับผู้อ่านที่เป็นมือใหม่ที่ยังไม่เคยใช้ .NET มาก่อน

ปัญหาที่สำคัญอย่างหนึ่งของผู้อ่านที่ยังไม่เคยทำเว็บไซต์ด้วย ASP.NET 4.0 มาก่อนเลยก็คือไม่รู้ว่า จะเริ่มต้นอย่างไรดี เพื่อให้ได้ผลงานออกมาเร็วที่สุดเท่าที่จะเป็นไปได้ อาจจะไม่ได้เด่นเท่าไรนัก แต่อย่างน้อยที่สุดก็อยากให้จับต้องได้เป็นชิ้นเป็นอันได้ในระดับหนึ่ง

เนื้อหาของหนังสือเล่มนี้เริ่มต้นด้วยเนื้อหาขั้นพื้นฐานเท่าที่ควรทราบ ไตรระดับไปเรื่อยๆ จนกระทั่งสร้างเว็บไซต์ที่ผู้เขียนคิดว่าเป็นแบบฝึกหัดที่ดี ก่อนที่จะเข้าสู่การสร้างเว็บไซต์ที่มีความสมบูรณ์ทั้งโครงสร้างและประสิทธิภาพในลำดับถัดไป

แนวทางการนำเสนอของซีรีส์หนังสือ ASP.NET 4.0

เนื่องจากรายละเอียดของ ASP.NET 4.0 ที่มากับ VS 2010 มีเนื้อหาค่อนข้างเยอะพอสมควร ผู้เขียนจึงแบ่งหนังสือในซีรีส์ของ ASP.NET 4.0 ออกเป็น 2 Step นั่นคือ

1. เนื้อหาและรายละเอียดต่างๆ ที่มากับ ASP.NET 4.0 แยกออกเป็น 4 เล่ม ได้แก่

- ◆ **ASP.NET 4.0 #1 Beginning** เป็นการนำเสนอพื้นฐานต่างๆ สำหรับผู้อ่านที่เป็นมือใหม่ในโลกของ ASP.NET 4.0 ที่ควรทราบในขั้นต้น ไปจนกระทั่งนำเนื้อหาของบทแรกๆ เข้ามาประกอบกันเป็นแบบฝึกหัดในการสร้างเว็บไซต์อย่างง่าย เพื่อให้ผู้อ่านเห็นผลงานโดยใช้ระยะเวลาเริ่มต้นไม่นานจนเกินไป
- ◆ **ASP.NET 4.0 #2 AJAX** การทำเว็บไซต์ให้สำเร็จเพียงอย่างเดียวไม่ใช่คำตอบของการทำเว็บในยุคปัจจุบัน หนทางหนึ่งที่สามารถปรับปรุงเว็บไซต์ให้ดียิ่งขึ้นก็คือ การใช้เทคนิค AJAX ประกอบด้วย
 - การใช้ภาษา JavaScript เพื่อแบ่งเบาภาระงานของฝั่ง Server ให้ฝั่ง Client เป็นผู้กระทำบ้าง
 - การนำเทคนิคการประมวลผลระหว่างฝั่ง Server กับฝั่ง Client โดยอาศัยภาษา JavaScript เข้ามาปรับปรุงการทำงานให้มีประสิทธิภาพมากยิ่งขึ้นที่เรียกว่า AJAX

- ◆ **ASP.NET 4.0 #3 jQuery, HTML 5, CSS3 และ Silverlight 4.0** ประกอบด้วย
 - JavaScript Framework ที่ชื่อว่า jQuery มีดีอย่างไร จึงทำให้ไม่ใคร่ขอพท์นำเข้ามาใช้งานร่วมกับ ASP.NET 4.0
 - การนำมาตรฐานเว็บใหม่ HTML 5 เข้ามาใช้งานร่วมกับ ASP.NET 4.0 ส่งผลให้ข้อมูลที่ปรากฏอยู่ในบราวเซอร์ไม่ใช่เพียงแค่การแสดงผลเหมือนภาษา HTML 4.0 เดิมอีกต่อไป แต่ยังสามารถอธิบายได้ด้วยว่า ข้อมูลที่กำลังนำเสนออยู่คืออะไร
 - การสร้างเว็บไซต์แบบ Rich Internet Application (RIA) ด้วย Silverlight 4.0 จะทำให้โลกของเว็บไซต์กับโลกของ Windows Application ขยับเข้าใกล้กันมากยิ่งขึ้น
 - ◆ **ASP.NET 4.0 #4 MVC 3.0 Framework & Microsoft WebMatrix** ประกอบด้วย
 - เนื้อหาที่ว่าด้วยเรื่องของการจัดโครงสร้างเว็บใหม่ โดยอาศัย MVC 3.0 Framework ซึ่งมากับ .NET 4.0 Framework
 - Microsoft WebMatrix เป็น Platform ที่เข้ามาปรับปรุงโครงสร้างของภาษา ASP.NET 4.0 ให้เข้าสู่มาตรฐานของภาษา HTML 5 อย่างแท้จริง
 - ปรับปรุงโครงสร้างของภาษา ASP.NET 4.0 โดยใช้ Engine ตัวใหม่เพื่อรวมส่วนแสดงผล (*.aspx) เข้ากับโค้ดการทำงาน (*.vb หรือ *.cs) เข้าเป็นหนึ่งเดียว (*.vbhtml หรือ *.cshtml)
2. การนำเนื้อหาที่นำเสนอใน 4 เล่มข้างต้นเข้ามาสู่การทำเว็บไซต์ที่เป็น Workshop อย่างแท้จริง

วิธีการนำเสนอของหนังสือเล่มนี้

ผู้เขียนนำเสนอเนื้อหาการใช้งาน ASP.NET 4.0 ทั้ง 2 ภาษาโดยอาศัย VS 2010 นั่นคือ ภาษา VB 2010 และภาษา VC# 2010 โดยที่บางตัวอย่างจะลงโค้ดทั้ง 2 ภาษา โดยที่ลงภาษา VB 2010 เป็นลำดับแรกตามด้วยภาษา VC# 2010 อยู่ภายในเครื่องหมายวงเล็บ

บางตัวอย่างผู้เขียนเลือกลงโค้ดภาษาใดภาษาหนึ่งเท่านั้นเพื่อให้กระชับต่อการนำเสนอ โดยมีจุดตัดสินใจที่ว่า โค้ดชุดนั้นๆ ภาษาใดมีประเด็นที่น่าสนใจมากกว่ากัน

ในกรณีที่ไม่มีความแตกต่างในส่วนของการทำงานของโค้ด ผู้เขียนก็จะลงโค้ดของภาษา VB 2010 เพียงภาษาเดียวเป็นหลัก ผู้เขียนคิดว่าไวยากรณ์ของภาษา VB 2010 ไม่เป็นปัญหาต่อผู้อ่านที่สนใจภาษา VC# 2010 โดยที่ตัวอย่างโค้ดที่อยู่ใน CD-ROM มีครบทั้ง 2 ภาษาทุกตัวอย่างอยู่ในโฟลเดอร์ VB และ CS

โดยที่ผู้เขียนใช้โปรแกรม Visual Studio 2010 Ultimate Edition เรียกสั้นๆ ว่า VS 2010 IDE เป็น Edition ที่ใหญ่ที่สุดเป็นหลัก ในกรณีนี้ที่ผู้อ่านใช้ Edition อื่นที่ไม่ตรงกับผู้เขียนอาจจะมีการเมนูไม่ครบตามรูปภาพที่ลงในหนังสือเล่มนี้

สคริปต์ที่ผู้เขียนนำลงในหนังสือไม่มีโค้ดที่เกิดจากการใช้คำสั่ง Auto Format... ซึ่งทำหน้าที่ตกแต่งให้เว็บคอนโทรลต่างๆ เช่น GridView, DataList ฯลฯ ให้สวยงามขึ้น เพราะว่าเมื่อนำลงในหนังสือแล้วจะทำให้ผู้อ่านดูสคริปต์ของหน้าเว็บเพจ *.aspx ค่อนข้างลำบาก ผู้เขียนถือว่าเป็นสคริปต์ที่ถูกสร้างขึ้นมาโดยอัตโนมัติไม่ใช่ส่วนที่ต้องการนำเสนอแต่อย่างใด

ผู้เขียนต้องการนำเสนอเฉพาะเนื้อหาหลักของหัวข้อดังกล่าวเป็นสำคัญ ส่วนโค้ดที่เกิดจากคำสั่งสำเร็จรูปเหล่านี้ผู้อ่านสามารถศึกษาเพิ่มเติมเองได้ในภายหลังอยู่แล้ว

ในหลายๆ หัวข้อมีการทำงานสัมพันธ์กันระหว่างส่วนแสดงผล ซึ่งเป็นหน้าที่ของสคริปต์ ASP.NET 4.0 (*.aspx) กับโค้ดการทำงาน (*.vb หรือ *.cs) เมื่อผู้เขียนใช้คำว่า คุณสมบัติ หมายถึง การมองเว็บคอนโทรลหรือออบเจกต์ที่กำลังเรียกใช้อยู่ในมุมมองของโค้ดทำงาน และใช้คำว่า แอตทริบิวต์ เมื่อมองในฐานอะลิเมนต์ตัวหนึ่งของสคริปต์ ASP.NET 4.0 แต่ทั้ง 2 คำที่กล่าวมาหมายถึงสิ่งเดียวกัน

กว่าจะเป็น Web 2.0

ในปัจจุบันเว็บไซต์จำนวนมากมายั่วยุเข้าสู่สิ่งที่เราเรียกว่า Web 2.0 เติบโตมาพร้อมกับอินเทอร์เน็ตความเร็วสูง เราจะมาดูว่าประวัติความเป็นมาเป็นอย่างไร ข้อจำกัดของผู้เขียนก็คือ ไม่สามารถนำเสนอรายละเอียดความเป็นมาได้อย่างครบถ้วน หรือสมบูรณ์แบบ เพียงแต่นำเสนอจุดเปลี่ยนที่สำคัญที่น่าสนใจ ผู้อ่านสามารถศึกษาค้นคว้าเพิ่มเติมในแต่ละช่วงเวลาเพื่อให้ได้สาระที่ครบถ้วน และถูกต้องยิ่งขึ้น

ผู้เขียนขอแบ่งช่วงช่วงการพัฒนาเว็บไซต์ออกเป็น 4 ช่วงใหญ่ๆ โดยใช้หลักการการทำงานเป็นเกณฑ์ของการแบ่งในแต่ละยุค ดังนี้

ยุคที่ 1 Web 1.0 (Client Side Script)

ในยุคเริ่มแรกของเครือข่ายอินเทอร์เน็ต เว็บไซต์ถูกสร้างขึ้นมาจากด้วยภาษา HyperText Markup Language เรียกสั้นๆ ว่า ภาษา HTML (นามสกุล *.htm หรือ *.html) เป็นภาษาที่ทำงานด้าน Client (Browser) ทั้งหมด 100%

ผู้ที่สร้างเว็บไซต์ขึ้นมาเป็นผู้กำหนดว่าต้องการนำเสนอข้อมูล เนื้อหา สาระ (Contents) อะไรบ้าง ผู้ใช้ไม่มีส่วนร่วมใดๆ ทั้งสิ้นกับเนื้อหาดังกล่าว ผู้เข้าเยี่ยมชมเว็บไซต์อยู่ในฐานะเป็นผู้เสพข่าวสารที่สนใจเท่านั้น

ภาษา HTML 1.0 ถูกสร้างขึ้นมากเพื่อกำหนดมาตรฐานของเอกสารที่แสดงในเว็บไซท์ โดยองค์กรที่ชื่อว่า W3C (www.w3.org) ถูกพัฒนาเรื่อยมาจนกระทั่งถึงยุคของภาษา HTML 3.2 ถูกสร้างขึ้นมากเพื่อกำหนดให้ภาษา HTML เป็นมาตรฐานกลางของการแสดงผลในบราวเซอร์

ในยุคของภาษา HTML 3.2 มีบราวเซอร์ยักษ์ใหญ่ 2 รายคือ Internet Explorer กับ Netscape Navigator ครอบครองอยู่ ถือเป็นยุครุ่งเรืองที่สุดของบราวเซอร์ Netscape Navigator อย่างแท้จริง

ช่วงเวลาดังกล่าวแม้ว่าองค์กร W3C จะเป็นผู้กำหนดมาตรฐานของเอกสารที่แสดงบนเว็บผ่านทางภาษา HTML 3.2 ก็จริงอยู่ แต่ผู้ผลิตบราวเซอร์ยักษ์ใหญ่ทั้ง 2 ต่างก็สร้างมาตรฐานของตัวเองขึ้นมา เพื่อให้การแสดงผลในเว็บเพจให้แตกต่างไปจากมาตรฐานกลางของภาษา HTML 3.2 จึงเป็นที่มาของปัญหาที่ว่า

ถ้าใช้มาตรฐานของบริษัทหนึ่งก็จะแสดงผลในบราวเซอร์ของอีกบริษัทหนึ่งไม่ได้ หรือแสดงผลผิดพลาด ในยุคนี้เกิดภาษาสคริปต์ที่ช่วยให้การแสดงผลในเว็บเพจดีขึ้น 3 ภาษาคือ ภาษา JavaScript, ภาษา VBScript ผู้เขียนขอกล่าวรวมถึงภาษา CSS (Cascading Style Sheets) ด้วย

ในยุคนี้เองมีคำศัพท์เกิดขึ้นมา 1 คำคือ คำว่า Cross Browser ซึ่งหมายถึง การสร้างเว็บเพจขึ้นมาสามารถแสดงผลได้อย่างถูกต้องทั้งใน Internet Explorer กับ Netscape Navigator

ยุคที่ 2 Web 1.5 (Server Side Script)

ต่อมาองค์กร W3C กำหนดมาตรฐานของภาษา HTML เข้าสู่เวอร์ชัน 4.0 ท่ามกลางความขัดแย้งของมาตรฐานการแสดงผลในหน้าเว็บเพจของบราวเซอร์ทั้ง 2 ค่าย

ในยุคนี้เกิดเทคโนโลยีที่เรียกว่า Flash ขึ้นมา ความมหัศจรรย์ของเทคโนโลยี Flash ก็คือ สามารถแสดงข้อมูลที่นอกเหนือไปจากข้อความและรูปภาพธรรมดาของภาษา HTML ไฟล์ขนาดเล็กก็สปีดสามารถโต้ตอบกับผู้ใช้ได้อย่างน่าสนใจ และยิ่งใหญ่ต่อเนื่องมาจนยุคปัจจุบัน

ข้อจำกัดของภาษา HTML 4.0 ก็คือ เป็นภาษาที่ทำงานด้าน Client ทั้งหมด ไม่สามารถรองรับการสร้างเว็บไซต์ที่เติบโตอย่างก้าวกระโดดได้อีกต่อไป จึงเกิดภาษาที่ทำหน้าที่ทางฝั่ง Server (Web Server) ขึ้นมา ภาษาที่จัดอยู่ในยุคบุกเบิกที่สำคัญ ได้แก่ ภาษา CGI, PHP, ASP, JSP ฯลฯ ภาษาเหล่านี้ทำงานทางฝั่ง Server ทั้งหมด 100% เช่นกัน

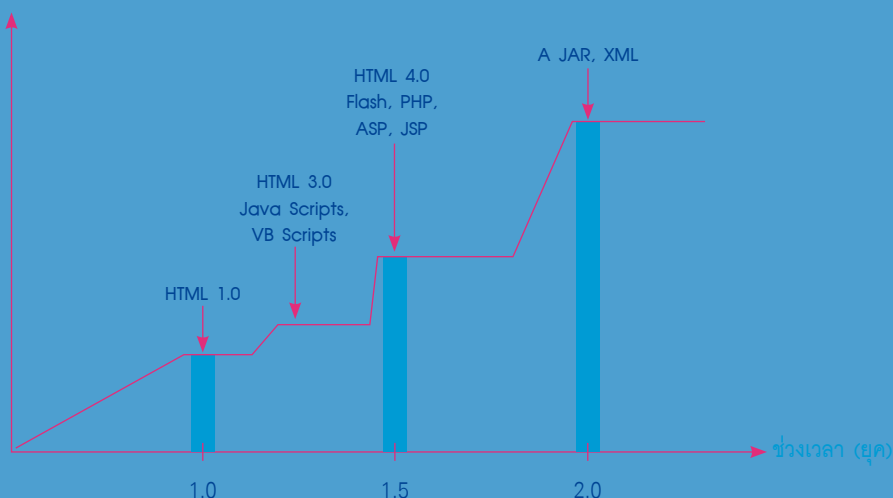
หลักการทำงานก็คือ Web Server จะแปลภาษาเหล่านี้ให้เป็นภาษา HTML ก่อนแล้วส่งกลับมายังฝั่งผู้ใช้ (Client) เพื่อแสดงผลในบราวเซอร์ในลำดับต่อมา

ภาษาที่ทำงานทางฝั่ง Server เหล่านี้ช่วยให้การสร้างเว็บไซต์แตกต่างไปจากเดิมอย่างสิ้นเชิง เจ้าของเว็บไซต์ไม่จำเป็นต้องนำเสนอเนื้อหาแต่เพียงฝ่ายเดียวอีกต่อไป เปิดโอกาสให้ผู้ใช้เป็นผู้นำเสนอเนื้อหาได้บ้าง เว็บไซต์จำพวกนี้ในยุคบุกเบิก ได้แก่ การเซ็นสมุดเยี่ยมเว็บไซต์ (GuestBook), เว็บบอร์ด (Webboard), การเขียนสมุดบันทึก (Diary Online) ฯลฯ

ในยุคนี้เองที่ผู้เยี่ยมชมเว็บไซต์มีส่วนร่วมของเนื้อหาในเว็บไซต์มากยิ่งขึ้น อาจกล่าวได้ว่าเป็นเจ้าของเนื้อหาสาระที่อยู่ในเว็บไซต์ในระดับหนึ่ง

ไมโครซอฟท์พัฒนา ASP ให้เข้ามาสู่ยุคของ .NET Framework ส่งผลให้ ASP เดิมซึ่งเรียกว่า ASP Classic ถูกภาษา ASP.NET เข้ามาทำหน้าที่รับผิดชอบการพัฒนา Web Application ด้วย .NET Framework แทน

ระดับความสามารถ



ยุคที่ 3 Web 2.0 (AJAX)

จากที่ผู้เขียนกล่าวไว้ว่า ใช้หลักการทำงานเป็นเกณฑ์ในการแบ่งแต่ละยุค อาจกล่าวโดยสรุปได้ว่า

1. ยุค Web 1.0 ภาษาของยุคนี้ทำงานทางฝั่ง Client (Browser) ทั้งหมด

2. ยุค Web 1.5 ภาษาของยุคนี้ทำงานทางฝั่ง Server (Web Server) ทั้งหมด

เห็นได้ว่าหลักการทำงานเปลี่ยนจากประมวลผลฝั่ง Client (ยุค Web 1.0) เข้ามาสู่ประมวลผลฝั่ง Server ทั้งหมด (Web 1.5) เราจะมาดูว่าการย้ายทุกสิ่งทุกอย่างไปประมวลผลฝั่ง Server ทั้งหมดก็มีข้อเสียเช่นกัน

ภาษาที่ทำงานฝั่ง Server ช่วยให้การสร้างเว็บไซต์แตกต่างไปจากเดิมมากมาย ทำให้ผู้ใช้มีส่วนร่วมมากขึ้นก็จริงอยู่ แต่การย้ายทุกสิ่งทุกอย่างไปฝั่ง Server ทั้งหมดจึงเกิดคำถามตามมาว่าทำไมเราจึงไม่แบ่งงานบางอย่างมาให้ฝั่ง Client ทำงานบ้าง

ด้วยเหตุข้างต้นจึงเกิดเทคนิคที่เรียกว่า Asynchronous JavaScript And XML เรียกสั้นๆ ว่า AJAX (เอ-แจ็กซ์ หรืออา-แจ็กซ์) หลักการของเทคนิคนี้ก็คือ ใช้ภาษา JavaScript เข้ามาช่วยแบ่งเบาภาระการทำงานของภาษาที่ทำงานฝั่ง Server

หลักการทำงานของเทคนิค AJAX มีอยู่ 2 ประการคือ

1. แบ่งงานให้ฝั่ง Client ประมวลผล เช่น แบ่งงานบางอย่างให้ฝั่ง Client ทำ

2. ถ้าเป็นงานของฝั่ง Server ให้ประมวลผลเท่าที่จำเป็น หมายความว่า ให้ฝั่ง Server ทำเท่าที่ควรจะทำไม่ใช่ทำทั้งหมด

ในยุคนี้เองที่ประชากรของอินเทอร์เน็ต สามารถเข้าถึงอินเทอร์เน็ตความเร็วสูงอย่างแพร่หลาย จึงทำให้เว็บไซต์ต่างๆ สามารถนำเสนอข้อมูล ข่าวสารในรูปแบบต่างๆ มากมาย เช่น การนำเสนอข่าวสารแบบวิดีโอของเว็บ Youtube, เว็บไซต์ของสถานีโทรทัศน์ออกอากาศแบบความละเอียดสูง Full HD หรือแม้กระทั่งไฟล์มัลติมีเดียความละเอียด 4K (Super HD) เป็นต้น

เราเห็นเว็บไซต์มากมายที่ผู้ใช้เป็นเจ้าของเนื้อหาทั้งหมดอย่างแท้จริง เราสร้างเว็บไซต์ข้อมูลส่วนตัวของเราใน hi5, เราเป็นผู้สื่อข่าวจำเป็นใน Twitter, เราสร้างสังคมออนไลน์ของเราใน facebook, ช่วงเวลาพักผ่อนของเราก็เขียน Blog บันทึกเรื่องราวที่เกิดขึ้น ฯลฯ

ที่สำคัญอีกประการหนึ่งก็คือ องค์กร W3C สร้างมาตรฐานของภาษา HTML ใหม่เข้าสู่เวอร์ชัน 5.0 เป้าหมายของภาษา HTML 5.0 คือ การนำเสนอข้อมูลที่นอกเหนือไปจากข้อความและรูปภาพนั่นคือ ข้อมูลมัลติมีเดียต่างๆ นั่นเอง

ตั้งแต่ยุคบุกเบิกจนกระทั่งปัจจุบัน การนำเสนอข้อมูลมัลติมีเดียต่างๆ ในหน้าเว็บเพจเป็นหน้าที่ของ Flash มาโดยตลอด ภาษา HTML 5.0 กำลังจะเข้ามาทำหน้าที่นี้ โดยมีบราวเซอร์ยักษ์ใหญ่ในปัจจุบันทั้งหมดสนับสนุนมาตรฐานของ HTML 5.0

ยุคที่ 4 Web 2.0 (Rich Internet Application-RIA)

เรากำลังเข้าสู่ยุคปลายของ Web 2.0 ไม่มีใครบอกได้ว่าเป็นเมื่อใด ไม่มีใครเป็นผู้กำหนดได้ว่า Web 3.0 ต้องประกอบไปด้วยอะไรบ้าง บอกได้แต่เพียงว่าควรจะมีอะไรบ้าง ผู้เขียนเองก็เช่นกัน เป็นเพียงคนๆ หนึ่งที่ขยี้เสฟข่าวสารอยู่ รู้รู้แต่เพียงว่าเว็บไซต์ที่พบเห็น “อลังการงานสร้างอีกแล้ว”

ภาษา ASP.NET คืออะไร

สำหรับผู้อ่านที่เคยสร้างเว็บไซต์ด้วยภาษา HTML ไม่เคยเขียนโปรแกรม ไม่ว่าจะเป็นใน Windows หรือในเว็บก็ตาม ย่อมเกิดคำถามขึ้นมว่าทำไมการสร้างเว็บไซต์จึงต้องมีการเขียนโปรแกรม

ผู้อ่านต้องไม่ลืมว่าเว็บไซต์ที่สร้างมาจากภาษา HTML ไม่สามารถตอบสนองความต้องการของผู้อ่านได้ทุกอย่าง ผู้อ่านสามารถทำลิงค์เชื่อมโยง, แทรกรูปภาพ, กำหนดสีข้อความต่างๆ, กำหนดสีพื้นหลัง ฯลฯ ได้ แต่ในบางครั้งผู้อ่านอาจจะเคยตั้งคำถามขึ้นมว่า เว็บไซต์สมาชิกทำอย่างไร, เว็บขายของทำอย่างไร เป็นต้น

เห็นได้ว่าการต้องการของผู้อ่านเพิ่มมากขึ้นเรื่อยๆ จึงเป็นที่มาของภาษาต่างๆ ในยุคที่ 2 นั้นเอง ที่ต้องเพิ่มเติมความสามารถของตัวภาษาขึ้นมาเพื่อรองรับกับความต้องการต่างๆ เหล่านี้ เว็บไซต์ที่เกิดจากภาษา HTML เพียงอย่างเดียวเดินทางมาถึงขอบเขตของมันแล้ว ส่งผลให้ผู้อ่านต้องศึกษาการเขียนโปรแกรมสำหรับทำเว็บไซต์ไปโดยปริยาย

อุปสรรคอย่างแรกของผู้อ่านที่ยังไม่เคยเขียนโปรแกรมมาก่อนเลยนั่นคือ ไม่ทราบกฎ กติกาของตัวภาษา รวมถึงไม่ทราบว่าจะเริ่มต้นอย่างไร นั่นจึงเป็นที่มาของหนังสือเล่มนี้

ผู้อ่านที่เป็นมือใหม่เริ่มต้นสร้างเว็บไซต์ที่เต็มไปด้วยเทคโนโลยีต่างๆ มากมายที่เกิดขึ้นมาในปัจจุบัน จึงเกิดปัญหาที่ไม่รู้จักรว่า คำศัพท์ที่ได้ยินคืออะไร ? ทำหน้าที่อะไร ? เราควรที่จะศึกษาหรือไม่ ? หรือเกี่ยวข้องกับเราหรือไม่ ? ล้วนแล้วแต่เป็นคำถามที่มีมือใหม่ต้องการทราบ

หน้าที่ของหนังสือเล่มนี้ไม่สามารถบอกทุกสิ่งทุกอย่างของการทำเว็บไซต์ แต่สามารถบอกมือใหม่ได้ว่า ผู้อ่านกำลังอยู่ ณ ตำแหน่งใดของโลกอินเทอร์เน็ต ผู้อ่านกำลังทำอะไรอยู่ อาจจะไม่ได้ลงรายละเอียดครบถ้วน แต่อย่างน้อยที่สุดผู้เขียนอยากให้เป็นจุดเริ่มต้นที่ดี รายละเอียดหรือเนื้อหาที่ไม่ได้กล่าวถึงก็ขอให้ผู้อ่านค้นคว้าจากแหล่งอื่นๆ เพิ่มเติมเพื่อให้ครบถ้วนยิ่งขึ้น

ภาษา ASP.NET เป็นภาษาที่ทำงานฝั่งเซิร์ฟเวอร์เรียกว่า Server Side Script เป็นของไมโครซอฟท์ ในปัจจุบันคือ ASP.NET 4.0 หมายความว่า เป็นภาษาที่รันบน .NET Framework 4.0 ซึ่งมาพร้อมกับ Visual Studio 2010 ผู้อ่านสามารถเขียน ASP.NET 4.0 ได้ 2 ภาษาคือ

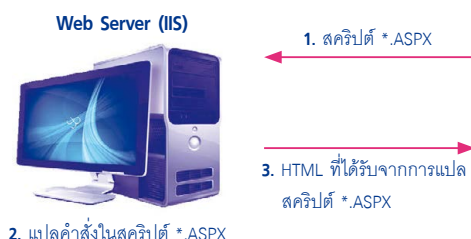
◆ ภาษา VB 2010

◆ ภาษา VC# 2010

เมื่อผู้อ่านสร้างเว็บไซต์ด้วย ASP.NET แล้ว สคริปต์ของ ASP.NET ถูกส่งไปประมวลผลที่ฝั่ง Server ก่อนจากนั้นจะส่งกลับมายังเบราว์เซอร์ ในรูปแบบของภาษา HTML เพื่อแสดงให้ผู้อ่านเห็นเป็นหน้าเว็บเพจ ดังรูปที่ 1-1

รูปที่ 1-1

ขั้นตอนการทำงานของ
สคริปต์ ASP.NET



เบราว์เซอร์ (IE)



จากรูปที่ 1-1 เห็นได้ว่าองค์ประกอบของสคริปต์ ASP.NET ประกอบด้วยการทำงาน 3 ขั้นตอนคือ

1) Web Server ที่ทำหน้าที่แปลภาษาสคริปต์ ASP.NET คือ IIS (Internet Information Services)

2) สคริปต์ภาษา ASP.NET (*.aspx)

3) บราวเซอร์ทำหน้าที่แสดงผลจากภาษา HTML ที่ได้จากการแปลภาษาสคริปต์ ASP.NET

ถ้าจะกล่าวในเชิงเทคนิคสักเล็กน้อย สามารถสรุปได้ว่า

- ◆ **ผู้ใช้ (ฝั่ง Client)** ร้องขอ (Request) เว็บเพจ (*.aspx) ไปยัง Web Server (ฝั่ง Server)
- ◆ **Web Server (ฝั่ง Server)** แปลสคริปต์ ASP.NET ได้ผลลัพธ์เป็น HTML ตอบสนอง (Response) มายังผู้ใช้ (ฝั่ง Client) แสดงผลในบราวเซอร์

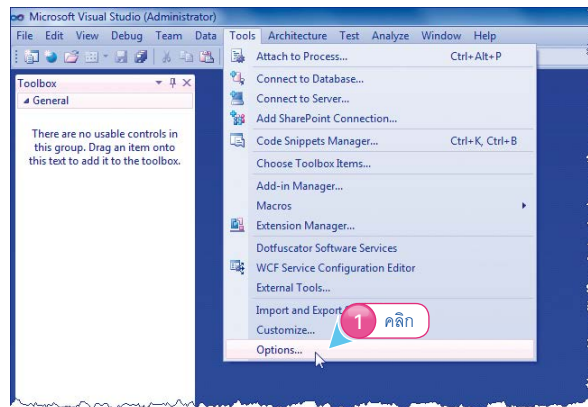
การกำหนดให้ VS 2010 แสดงไดอะล็อกบ็อกซ์ Show New Project dialog box

ไม่ว่าผู้อ่านจะใช้ VS 2010 Edition ใดก็ตาม ต้องมีการกำหนดค่าให้กับ VS 2010 เล็กน้อยเพื่อให้เหมาะสมกับการใช้งานกล่าวคือ ทุกๆ ครั้งที่ผู้อ่านเปิดโปรแกรม VS 2010 ขึ้นมา ควรจะมีไดอะล็อกบ็อกซ์ Show New Project dialog box ปรากฏขึ้นมา เพื่อแสดงโปรเจกต์ชนิดต่างๆ ให้ผู้อ่านเลือกใช้งาน ซึ่งมีขั้นตอนดังนี้

1. เปิด VS 2010 ขึ้นมา คลิกเมนู Tools > Options...

รูปที่ 1-2

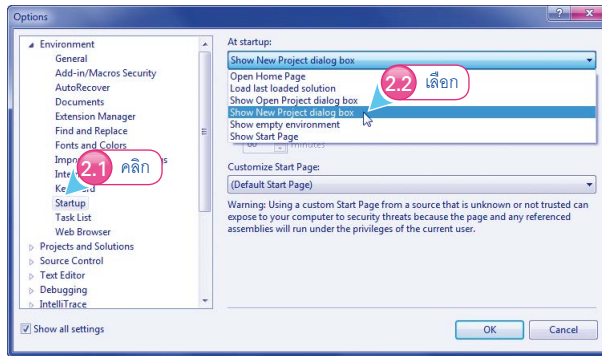
การคลิกเมนู Tools > Options...



2. ที่หัวข้อ Environment คลิกเลือก Startup ที่ช่อง At startup: คลิกเลือก Show New Project dialog box

รูปที่ 1-3

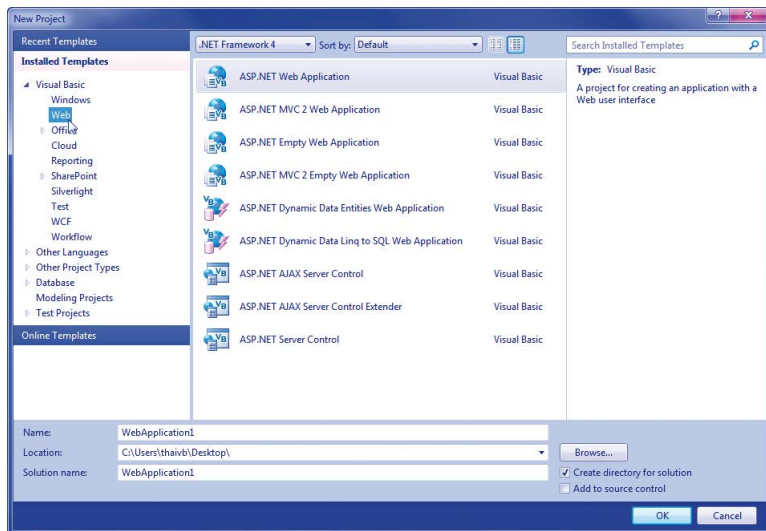
การเลือกรายการ
Show New Project
dialog box



ทุกๆ ครั้งที่ผู้อ่านเปิดโปรแกรม VS 2010 ขึ้นมาก็จะมีไดอะล็อกบ็อกซ์ Show New Project dialog box ปรากฏขึ้นมาเป็นลำดับแรก ให้ผู้อ่านเลือกชนิดโปรเจกต์ที่ต้องการใช้งานดังรูป

รูปที่ 1-4

ไดอะล็อกบ็อกซ์ Show
New Project dialog
box

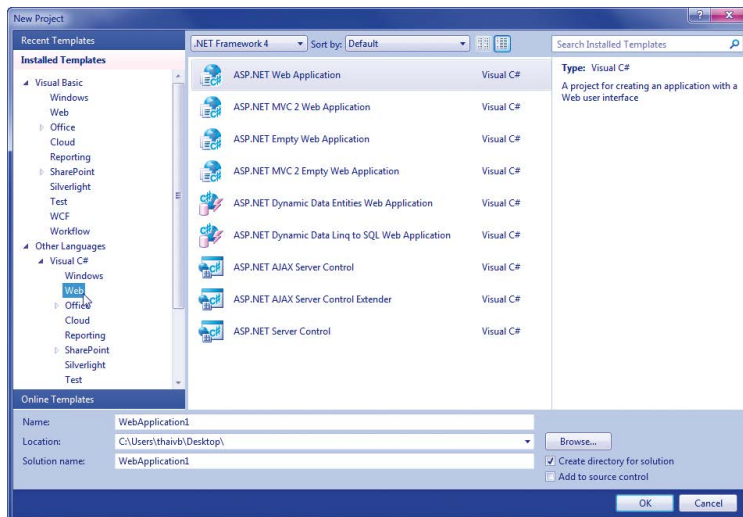


จากรูปที่ 1-4 ขอบเขตของหนังสือเล่มนี้ต้องการพัฒนาเว็บแอปพลิเคชันด้วย ASP.NET 4.0 (รันบน .NET Framework 4.0) จึงเลือกหัวข้อ Web โดยที่

- ◆ **Visual Basic** หมายถึง ใช้ภาษา VB 2010
- ◆ **Other Languages** หมายถึง ใช้ภาษาอื่นๆ ในกรณีนี้คือ ภาษา VC# 2010

รูปที่ 1-5

กรณีเลือกใช้ภาษา
VC# 2010

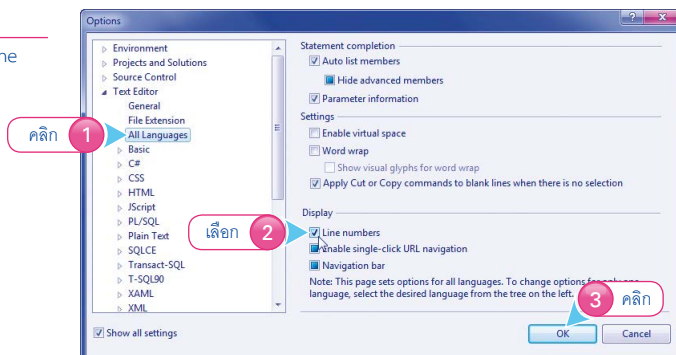


การกำหนดให้ Editor ของ VS 2010 แสดงหมายเลขบรรทัด

การกำหนดให้ Editor ของ VS 2010 แสดงหมายเลขบรรทัดเพื่อให้ง่ายในการปรับปรุงแก้ไขการทำงานของโค้ด โดยผู้อ่านสามารถกำหนดได้โดยคลิกเมนู Tools > Options... คลิกเลือก Text Editor > All Languages คลิกเลือก Line numbers

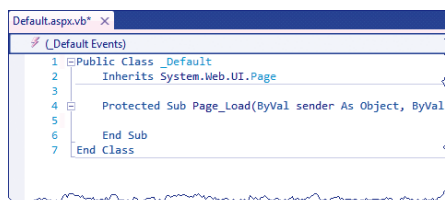
รูปที่ 1-6

การคลิกเลือก Line
numbers



รูปที่ 1-7

แสดงหมายเลขบรรทัด
ใน Editor ของ
VS 2010

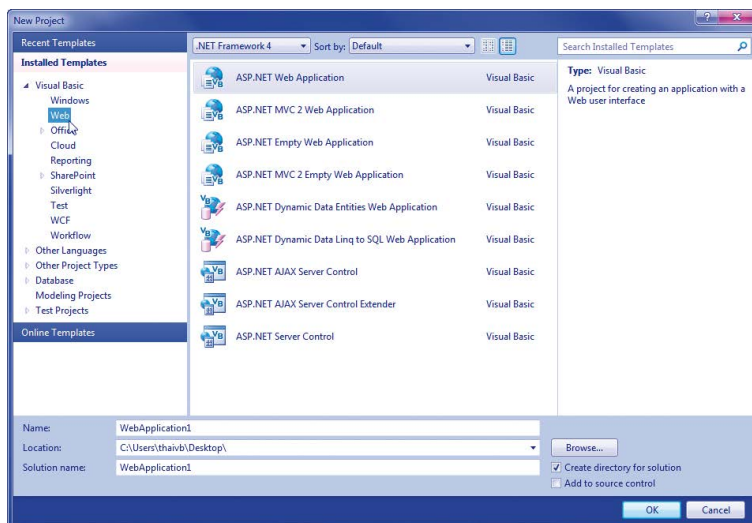


ทำความรู้จักกับโปรเจกต์ด้าน Web ของ VS 2010

VS 2010 มีโปรเจกต์ที่เกี่ยวข้องกับการทำเว็บด้วย ASP.NET หลายประเภท เรียกว่า Project Template เราจะทำความรู้จักขั้นต้นว่า โปรเจกต์แต่ละประเภทของ VS 2010 ด้านเว็บมีอะไรบ้าง

รูปที่ 1-8

ไดอะล็อกบ็อกซ์ New Project



Project Template ด้านเว็บแต่ละประเภทของ ASP.NET 4.0 กำหนดโครงสร้างโปรเจกต์เริ่มต้นให้ผู้อ่านไม่เหมือนกัน อย่างน้อยที่สุดการเลือกใช้ Project Template ที่ใกล้เคียงกับความต้องการของเรามากที่สุดย่อมจะเป็นจุดเริ่มต้นที่ดี ดังต่อไปนี้

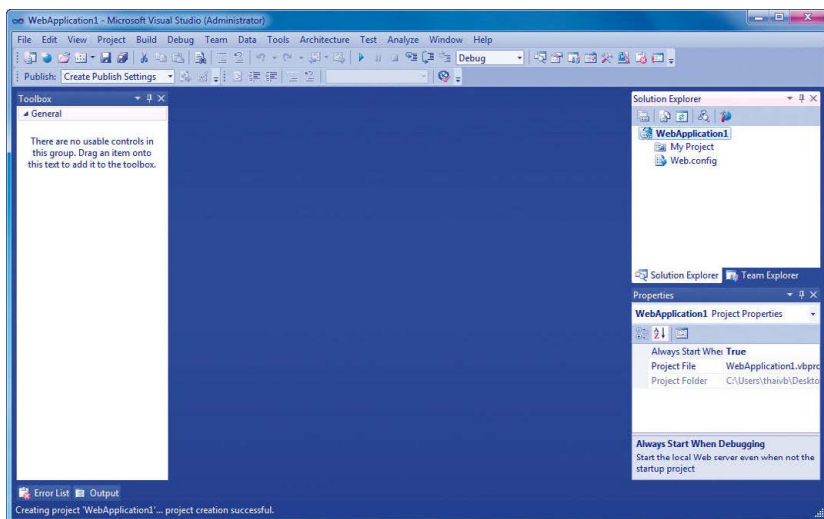
- 1. ASP.NET Web Application** เหมาะสมกับผู้ที่เคยใช้งาน ASP.NET เวอร์ชันก่อนๆ มานั่ง ประกอบไปด้วย
 - ◆ การสร้างเว็บไซต์โดยใช้โครงสร้างเว็บไซต์แบบ Master Page
 - ◆ สามารถเรียกใช้ jQuery เวอร์ชัน 1.4.1
- 2. ASP.NET MVC 2 Web Application** เป็นสภาพแวดล้อมของการพัฒนาเว็บเต็มรูปแบบประกอบไปด้วย
 - ◆ การสร้างเว็บไซต์โดยใช้โครงสร้างแบบ MVC (Model-View-Controller) ที่มากับ VS 2010 คือ MVC Framework 2.0
 - ◆ สามารถเรียกใช้ jQuery เวอร์ชัน 1.4.1
 - ◆ พร้อมใช้เทคนิคการประมวลผลแบบ Ajax
- 3. ASP.NET Empty Web Application** เป็นโปรเจกต์ว่างเปล่าเหมาะสมกับผู้ใช้มือใหม่ที่ไม่เคยใช้งาน ASP.NET มาก่อนเลย
- 4. ASP.NET MVC 2 Empty Web Application** เป็นโปรเจกต์ว่างเปล่าเช่นกันแต่โครงสร้างของโปรเจกต์เป็นแบบ MVC 2.0 Framework

ทำความรู้จักกับโปรเจกต์ชนิด ASP.NET Empty Web Application

Project Template ชนิด ASP.NET Empty Web Application เป็นสภาพแวดล้อมที่ VS 2010 ไม่มีการกำหนดโครงสร้างใดๆ ทั้งสิ้นให้กับผู้อ่าน เมื่อต้องการใช้งานอะไรผู้อ่านต้องเป็นผู้เพิ่มเข้ามาในโปรเจกต์เอง

รูปที่ 1-9

สภาพแวดล้อมของ ASP.NET Empty Web Application

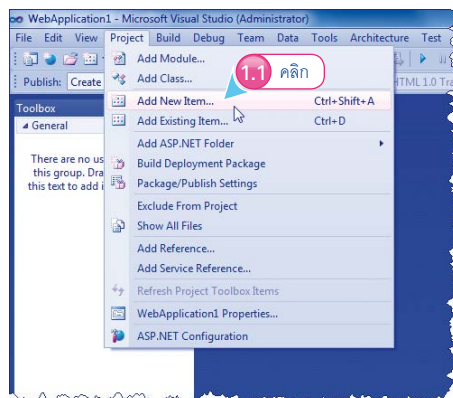


จากรูปที่ 1-9 พบว่าที่หน้าต่าง Solution Explorer มีเพียงไฟล์ Web.Config เพียงไฟล์เดียวอยู่ในโปรเจกต์ สิ่งที่ต้องทำในขั้นตอนนี้

1. การเพิ่มเว็บฟอร์มเข้ามาในโปรเจกต์ สามารถทำได้โดยคลิกเมนู Project > Add New Item... และเลือกไอคอน Web Form

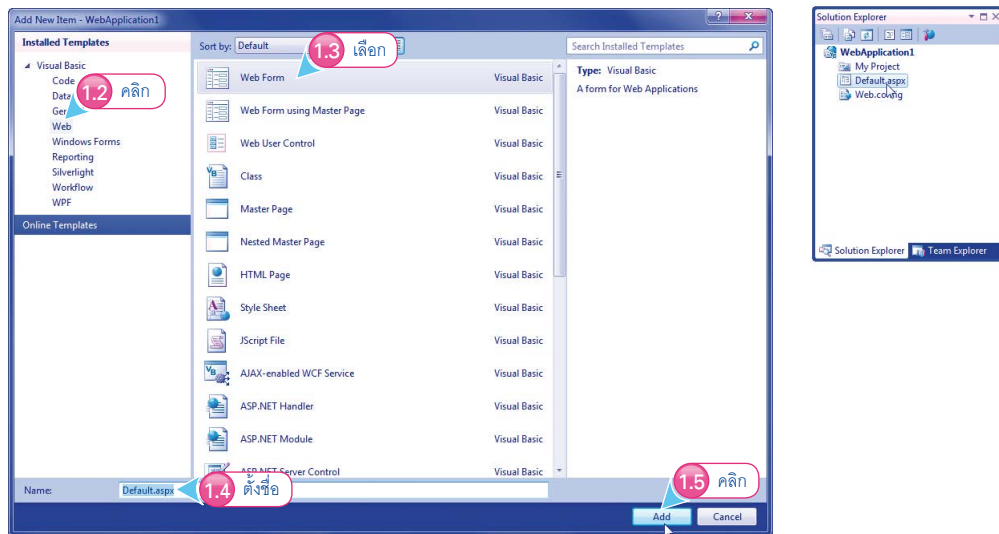
รูปที่ 1-10

การเพิ่มเว็บฟอร์มเข้ามาในโปรเจกต์ปัจจุบัน



รูปที่ 1-10 (ต่อ)

การเพิ่มเว็บฟอร์มเข้า
มาในโปรเจกต์ปัจจุบัน



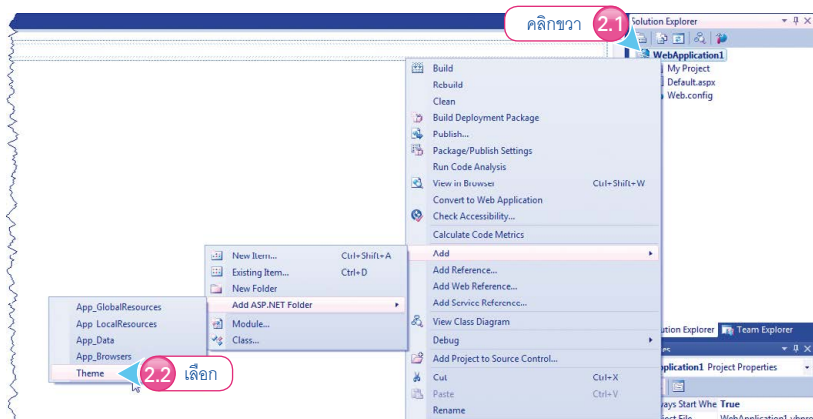
จากรูปที่ 1-10 เว็บฟอร์มแรกที่เริ่มต้น หรือเป็นหน้าแรกในโฮมเพจของเราควรตั้งชื่อว่า Default.aspx ส่วนเว็บฟอร์มหน้าอื่นๆ ก็แล้วแต่ความต้องการ

2. การเพิ่มไฟล์เดอร์พิเศษต่างๆ ของ **ASP.NET 4.0** เพื่อใช้ในโฮมเพจ โดยปกติแล้วการสร้างเว็บไซต์ขึ้นมาจะต้องมีการสร้างไฟล์เดอร์ต่างๆ ขึ้นมาเพื่อทำหน้าที่เฉพาะอย่าง เช่น เก็บรูปภาพ, เก็บสคริปต์ต่างๆ, เก็บฐานข้อมูล เป็นต้น

ใน ASP.NET 4.0 ถือว่าไฟล์เดอร์เหล่านี้เป็นไฟล์เดอร์พิเศษที่หน้าต่าง Solution Explorer ซึ่งสามารถเพิ่มเข้ามาในโปรเจกต์ได้โดยเลือกชื่อโปรเจกต์ คลิกขวาแล้วเลือกคำสั่ง Add > Add ASP.NET Folder เพื่อสร้างไฟล์เดอร์พิเศษขึ้นมาในโปรเจกต์ ดังนี้

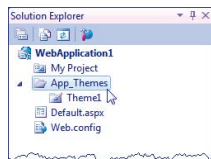
รูปที่ 1-11

แสดงการเพิ่มไฟล์เดอร์
App_Themes เข้ามา
ในโปรเจกต์



รูปที่ 1-11 (ต่อ)

แสดงการเพิ่มโฟลเดอร์
App_Themes เข้ามา
ในโปรเจกต์



หน้าที่โฟลเดอร์แต่ละประเภทผู้เขียนจะกล่าวถึงอีกครั้งเมื่อมีการใช้งานหรือเนื้อหาเกี่ยวข้อง

3. ที่ไฟล์ Web.Config มีสิ่งที่ควรแก้ไขก็คือ ในขณะที่โปรเจกต์อยู่ระหว่างพัฒนายังไม่เสร็จสมบูรณ์ ให้ผู้อ่านกำหนดแอตทริบิวต์ `debug="true"` เพื่อให้ VS 2010 มีการ debugging ได้ทำงานในขณะที่ยังโปรเจกต์ด้วย

รูปที่ 1-12

การกำหนดแอตทริบิวต์
debug เท่ากับ true



ผู้อ่านจะกำหนดค่าแอตทริบิวต์
debug เท่ากับ false ก็ต่อเมื่อโปรเจกต์ได้
พัฒนาเสร็จสมบูรณ์ไม่มีการแก้ไขอีกแล้ว

WebMatrix Platform

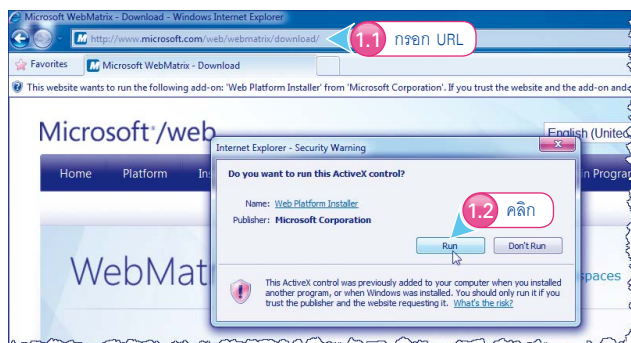
อีกช่องทางหนึ่งที่ผู้อ่านสามารถเรียกใช้ ASP.NET 4.0 นอกเหนือไปจากการติดตั้ง Visual Studio 2010 ตามปกติ นั่นคือ โครงการ WebMatrix ของไมโครซอฟท์

WebMatrix เป็นช่องทางที่ไมโครซอฟท์ต้องการให้นักพัฒนา Web Application มีคู่มือแนวทางที่ดี ไม่ว่าจะเป็นการติดตั้งเครื่องมือในการพัฒนา, การติดตามความคืบหน้าล่าสุดต่างๆ ที่เกี่ยวข้อง เห็นได้ว่าเป็นคำถามที่ผู้อ่านคุ้นเคยกันเป็นอย่างดีไม่ว่าจะเป็นเราต้องติดตั้งอะไรบ้าง ?, โปรแกรมที่ติดตั้งอยู่ในเครื่องของเราเป็นเวอร์ชันล่าสุดแล้วหรือยัง ? หรือแม้กระทั่งมือใหม่ที่ไม่ทราบว่าจะต้องเริ่มต้นอย่างไร WebMatrix มีคำตอบให้ ซึ่งมีขั้นตอนการใช้งาน ดังนี้

1. เข้าเว็บไซต์ www.asp.net/webmatrix คลิกไอคอนของโครงการ WebMatrix เพื่อดาวน์โหลดตัวติดตั้งผ่านเว็บ หมายความว่า เครื่องของผู้อ่านต้องเชื่อมต่ออินเทอร์เน็ตเพื่ออัปเดตเครื่องให้มีความทันสมัยได้ตลอดเวลา โดยที่ไมโครซอฟท์เป็นผู้นำทาง ดังนี้

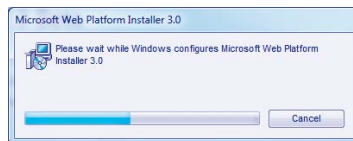
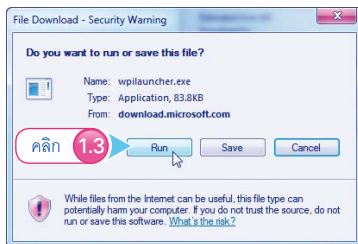
รูปที่ 1-13

การดาวน์โหลดและ
ติดตั้ง WebMatrix ใน
เครื่องของผู้เขียน



รูปที่ 1-13 (ต่อ)

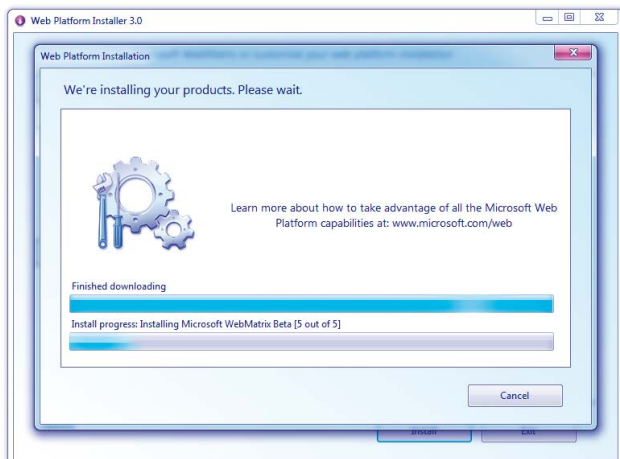
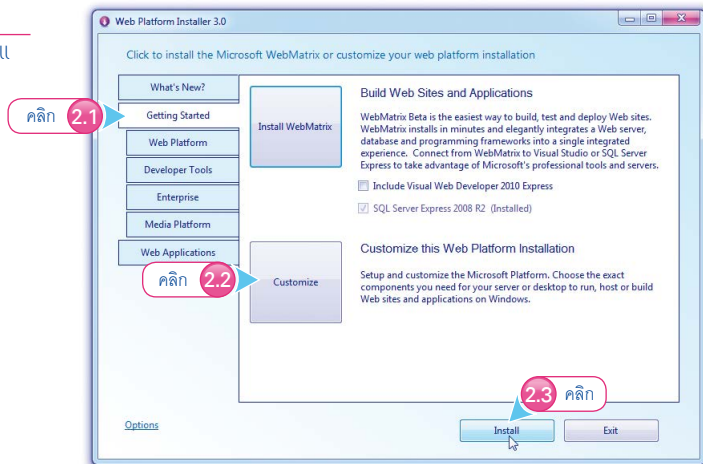
การดาวน์โหลดและติดตั้ง WebMatrix ในเครื่องของผู้เขียน



2. การติดตั้งโปรแกรมและคอมโพเนนต์ต่างๆ เท่าที่ปัจจุบันมีอยู่สามารถทำได้ 2 วิธีคือ
 - ◆ **Install WebMatrix** เพื่อดาวน์โหลดและติดตั้งโปรแกรม, คอมโพเนนต์ต่างๆ ที่ไม่ใคร่ซอฟต์แวร์กำหนดไว้ให้ในขั้นต้นว่าควรจะติดตั้งอะไรบ้าง มีผลดีในแง่ที่ว่าถ้าผู้อ่านเป็นผู้ใช้มือใหม่แล้วไม่ทราบว่าจะต้องติดตั้งอะไรบ้าง ทางเลือกนี้เป็นคำตอบที่ดีที่สุดสำหรับผู้อ่าน

รูปที่ 1-14

กรณีเลือก Install WebMatrix

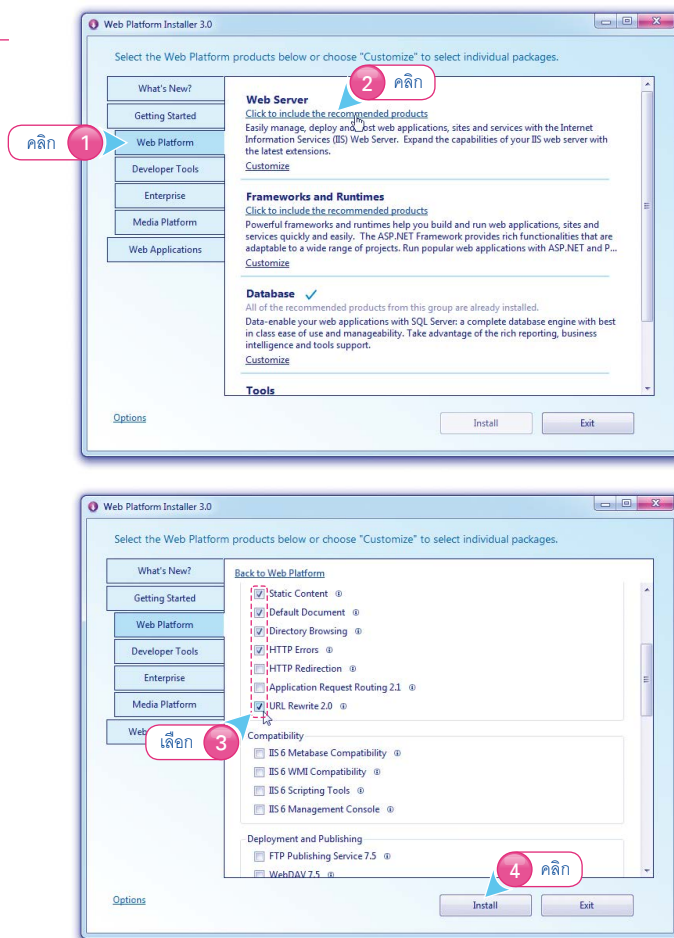


จากรูปที่ 1-14 รายการที่ติดตั้งขึ้นอยู่กับว่า เครื่องของผู้อ่านในปัจจุบันมีการติดตั้งอะไรอยู่บ้างก็จะมี การอัปเดตโปรแกรมต่างๆ เท่าที่เกี่ยวข้องให้โดยอัตโนมัติ การอัปเดตผ่านทาง Web Matrix มีผลกับ VS 2010 ของผู้อ่านด้วยเช่นกัน

- ◆ **Customize** เป็นการเลือกโปรแกรมและส่วนประกอบ, คอมโพเนนต์ต่างๆ ที่ต้องการติดตั้งในเครื่อง ซึ่งผู้อ่านต้องทราบข้อมูลข่าวสารพอสมควรว่า สิ่งที่น่าสนใจต้องติดตั้งอะไรบ้าง ผู้อ่านควรติดตาม ข่าวสารของ ASP.NET อยู่เสมอ แม้ว่า ASP.NET 4.0 เป็นเวอร์ชันหลักที่ออกมากับ .NET 4.0 ก็ตาม แต่ก็จะมีการอัปเดตตัวภาษา, องค์ประกอบอยู่ตลอดเวลา

รูปที่ 1-15

กรณีเลือก
Customize



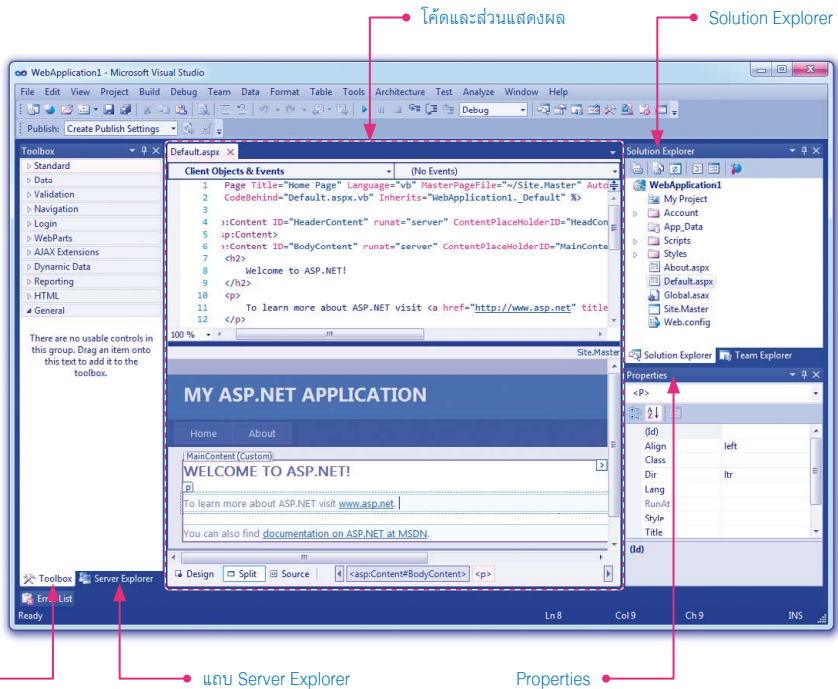
จากรูปที่ 1-15 แน่หนอนว่าต่อไปในอนาคตรายการตัวเลือกต่างๆ ย่อมที่จะแตกต่างไปจากขณะนี้ ผู้เขียนแต่งต้นฉบับหนังสือเล่มนี้ แต่ประเด็นที่ต้องการนำเสนอก็คือ ช่องทางที่นักพัฒนา Web Application ด้วย .NET Framework ควรจะทราบไว้ก่อนว่า ความก้าวหน้าในปัจจุบันมีตลอดเวลา โดยเฉพาะอย่างยิ่งผู้อ่านที่ต้องการเป็นผู้พัฒนาเว็บก็ต้องตื่นตัวอยู่ตลอดเวลาเพื่อต้อนรับสิ่งใหม่ๆ อยู่เสมอ

ทำความรู้จักกับ VS 2010 IDE แบบ Web Application

เราจะมาลองทำความรู้จักกับสภาพแวดล้อมของ VS 2010 แบบ Web Application ในขั้นตอนดังรูป

รูปที่ 1-16

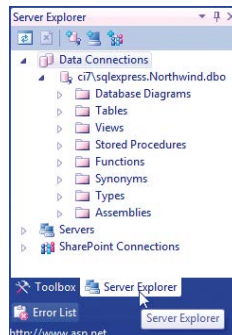
สภาพแวดล้อมของ
VS 2010 แบบ Web
Application



1. **แถบ Toolbox** เป็นแถบเครื่องมือเก็บเว็บคอนโทรลต่างๆ ที่ผู้อ่านสามารถเรียกใช้งานได้ แยกออกเป็นแต่ละแท็บตามหน้าที่
2. **แถบ Server Explorer** ทำหน้าที่แสดงการเชื่อมต่อกับฐานข้อมูล SQL Server ที่ผู้อ่านเคยติดต่อหรือเคยใช้งานกับโปรเจกต์ปัจจุบัน

รูปที่ 1-17

แถบ Server Explorer



จากรูปที่ 1-17 พบว่าโปรเจกต์ปัจจุบันเคยติดต่อกับฐานข้อมูล Northwind

3. **โค้ดและส่วนแสดงผลของเว็บฟอร์ม** เป็นพื้นที่สำหรับแสดงผลของแต่ละเว็บเพจที่ผู้อ่านเป็นผู้ออกแบบขึ้นมา แบ่งออกเป็น 3 แบบคือ
 - ◆ **Design** หมายถึง แสดงเว็บเพจในโหมดออกแบบ
 - ◆ **Split** หมายถึง แสดงเว็บเพจในโหมดออกแบบและแสดงโค้ดด้วย โดยการแบ่งหน้าจอออกเป็น 2 ส่วน
 - ◆ **Source** หมายถึง แสดงโค้ดของเว็บเพจปัจจุบัน
4. **หน้าต่าง Solution Explorer** ทำหน้าที่แสดงโครงสร้างเว็บเพจและรายการไฟล์เดอร์ต่างๆ ของโปรเจกต์ปัจจุบันที่เปิดขึ้นมา
5. **หน้าต่าง Properties** หมายถึง คุณสมบัติต่างๆ ของคอนโทรล, อิลิเมนต์ หรือออบเจกต์ปัจจุบันที่กำลังโฟกัสอยู่

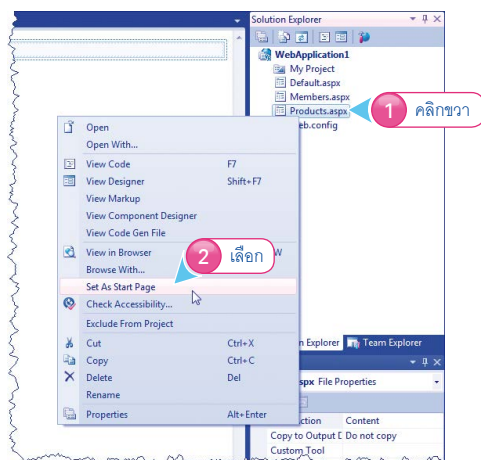
การกำหนดเว็บฟอร์มเริ่มต้น

โดยปกติแล้วการสร้างเว็บไซต์ขึ้นมาจะประกอบไปด้วยเว็บฟอร์มหลายเว็บฟอร์มในเวลาเดียวกัน โดยปกติแล้วมักที่กำหนดให้เว็บฟอร์ม Default.aspx เป็นหน้าแรก เป็นหน้าเริ่มต้นของเว็บไซต์

แต่ถ้าผู้อ่านต้องการกำหนดให้เว็บฟอร์มอื่นเป็นเว็บฟอร์มเริ่มต้น สามารถทำได้โดยคลิกขวาแล้วเลือกคำสั่ง Set As Start Page เพื่อกำหนดให้เว็บฟอร์مدังกล่าวเป็นเว็บฟอร์มเริ่มต้นดังรูป

รูปที่ 1-18

การกำหนดให้เว็บฟอร์มที่ชื่อว่า Products.aspx เป็นฟอร์มเริ่มต้น



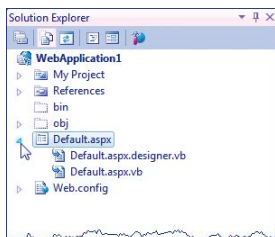
จากรูปที่ 1-18 เมื่อผู้อ่านสั่งรันโปรเจกต์เว็บฟอร์มที่ชื่อว่า Products.aspx จะถูกสั่งให้รันขึ้นมาเป็นลำดับแรก

ทำความรู้จักกับโครงสร้างเบื้องต้นของโปรเจกต์ชนิด ASP.NET 4.0

ได้การทำงานของภาษา VB 2010 เก็บอยู่ในไฟล์ *.vb ส่วนภาษา VC# 2010 เก็บอยู่ในไฟล์ *.cs ของแต่ละโปรเจกต์แสดงอยู่ในหน้าต่าง Solution Explorer

รูปที่ 1-19

รายการไฟล์ที่ปรากฏใน
หน้าต่าง Solution
Explorer ของ ASP.NET 4.0
ของภาษา VB 2010

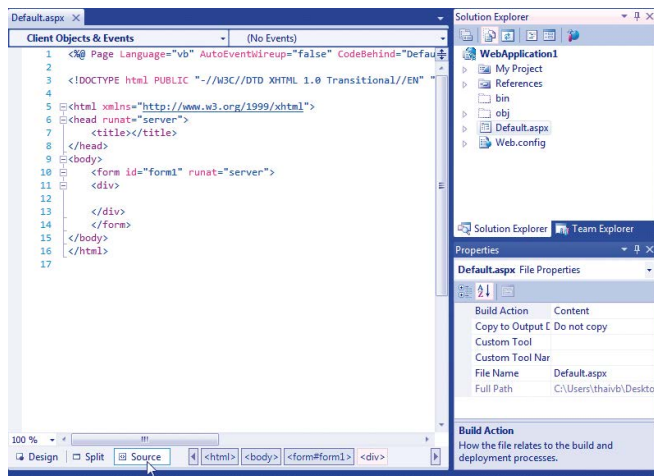


จากรูปที่ 1-19 ให้ผู้อ่านคลิกปุ่ม  เพื่อแสดงรายการไฟล์ทั้งหมดของโปรเจกต์ปัจจุบัน โดยที่

- ◆ ไฟล์ **Default.aspx** ทำหน้าที่เป็นส่วนแสดงผลในหน้าเว็บเพจ สคริปต์ต่างๆ ที่เกิดขึ้นในไฟล์นี้เกิดจากการเรียกใช้งานเว็บคอนโทรลต่างๆ ถูกสร้างขึ้นมาโดยอัตโนมัติด้วย VS 2010

รูปที่ 1-20

แสดงแท็บ Source



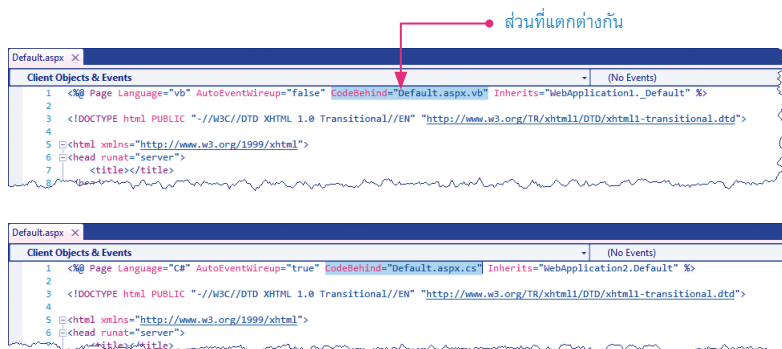
ถ้าผู้อ่านโฟกัสที่ไฟล์นามสกุล *.aspx ในหน้าต่าง Solution Explorer แล้วคลิกแท็บ Source เป็นการแสดงสคริปต์ ASP.NET ของไฟล์ *.aspx ที่ผู้อ่านเลือก

- ◆ ไฟล์ **Default.aspx.vb** (หรือไฟล์ **Default.aspx.cs**) ทำหน้าที่เก็บโค้ดการทำงานของภาษา VB 2010 เป็นพื้นที่ที่ผู้อ่านต้องเขียนโค้ดขึ้นมานั่นเอง ส่วนโปรเจกต์แบบภาษา VC# 2010 มีลักษณะเช่นเดียวกับแบบภาษา VB 2010 แตกต่างกันตรงที่ว่า ไฟล์ที่เก็บโค้ดการทำงานของภาษามีสกุล *.cs นั่นคือ Default.aspx.cs เราเรียกโครงสร้างโค้ดดังกล่าวว่า การเขียนโค้ดแบบ Code-Behind

ความแตกต่างระหว่างสคริปต์ ASP.NET 4.0 (*.aspx) ของภาษา VB 2010 กับภาษา VC# 2010 นั่นคือ การอ้างอิงไฟล์ที่เก็บโค้ดทำงานที่ระบุไว้ในบรรทัดแรกดังรูป

รูปที่ 1-21

การอ้างอิงไฟล์เก็บโค้ดการทำงานระหว่างภาษา VB 2010 กับภาษา VC# 2010 ที่บรรทัดแรก



ข้อแตกต่างก็คือ การอ้างอิงไฟล์ที่เก็บโค้ดทำงานในไฟล์ *.aspx จะอ้างอิงตามชื่อไฟล์ของภาษานั้นๆ เช่น ภาษา VB 2010 ระบุไฟล์นามสกุล *.vb ส่วนภาษา VC# 2010 ระบุไฟล์นามสกุล *.cs

ผู้เขียนขอละในฐานที่เข้าใจว่า ถ้ามีการลงสคริปต์ของ ASP.NET 4.0 (*.aspx) จะลงแต่เฉพาะส่วนที่เกี่ยวข้องกับโค้ดทำงานเท่านั้น แต่ขอให้ผู้อ่านทราบว่ามี ความแตกต่างกันเฉพาะบรรทัดแรกเท่านั้น ดังตัวอย่างต่อไปนี้

ASP.NET 4.0

```
<body>
  <form id="form1" runat="server">
    <div>
      ...
    </div>
  </form>
</body>
```

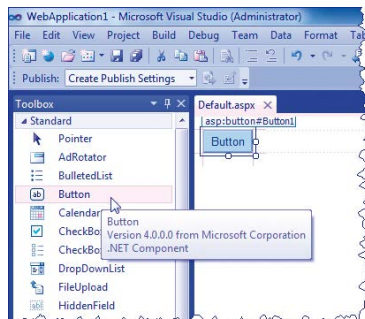
ตัวอย่างสคริปต์ข้างต้น หมายความว่า ต้องการแสดงสคริปต์บางส่วนที่มีความสำคัญในไฟล์ *.aspx ที่ผู้เขียนคาดว่าเกี่ยวข้องกับโค้ดการทำงาน ซึ่งอยู่ระหว่างอิลิเมนต์ <body>...</body> เท่านั้น ผู้เขียนขอละในฐานเข้าใจที่ว่าสคริปต์บรรทัดอื่นๆ ที่ไม่ได้นำมาแสดงในหนังสือไม่มีความเกี่ยวข้อง หรือไม่มีผลมากนักต่อประเด็นที่น่าเสนอ

ส่วนข้อความที่กำกับกรอบสีเหลี่ยมบอกให้ผู้อ่านทราบว่า โค้ดหรือสคริปต์ที่อยู่ในกรอบสีเหลี่ยมดังกล่าวเป็นของสคริปต์ ASP.NET 4.0 หรือภาษา VB 2010 หรือ VC# 2010 ฯลฯ ทำให้ผู้อ่านแยกแยะได้ว่า โค้ดที่ลงในหนังสือหมายถึงโค้ดส่วนใด

ข้อแตกต่างที่สำคัญอีกประการหนึ่งคือ เหตุการณ์ของเว็บคอนโทรลต่างๆ ที่ผู้อ่านใช้งานมีความแตกต่างกันระหว่างภาษา VB 2010 กับ VC# 2010 เล็กน้อยกล่าวคือ

รูปที่ 1-22

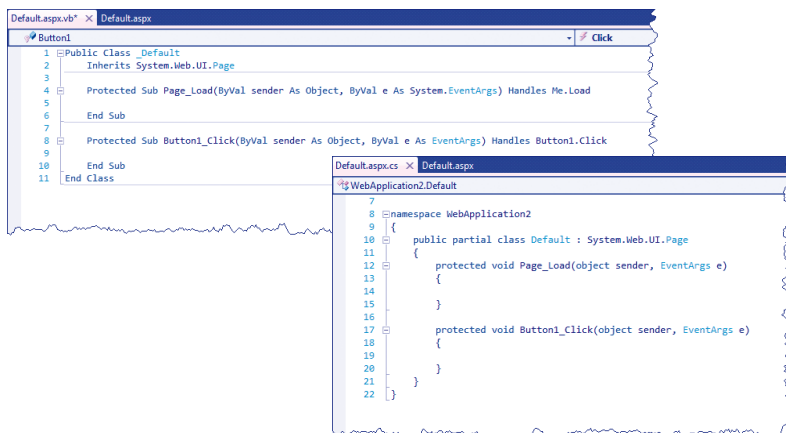
การเรียกใช้เว็บ
คอนโทรล Button ของ
ASP.NET 4.0



เมื่อผู้อ่านดับเบิลคลิกที่เว็บคอนโทรล Button ส่งผลให้ VS 2010 สร้างเหตุการณ์ Button1_Click() ขึ้นมาให้อ่านในไฟล์ Default.aspx.vb (.cs)

รูปที่ 1-23

แสดงโค้ดทำงานของ
เหตุการณ์ Button1_
Click() ของภาษา VB
2010 และภาษา VC#
2010



ข้อแตกต่างที่เกิดขึ้นในสคริปต์ของ ASP.NET 4.0 (*.aspx) นั่นคือ ในภาษา VC# 2010 มีการอ้างอิงเหตุการณ์ Button1_Click() ไว้ด้วย ส่วนภาษา VB 2010 ไม่มีการอ้างอิงแต่อย่างใดจึงเป็นข้อแตกต่างระหว่างสคริปต์ ASP.NET 4.0 ของทั้ง 2 ภาษา

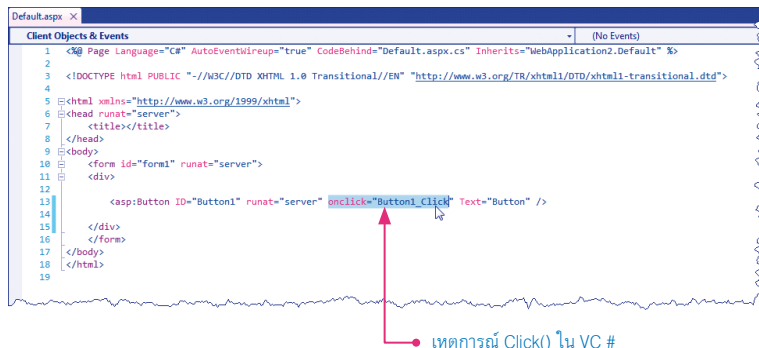
รูปที่ 1-24

การอ้างอิงเหตุการณ์
Button1_Click ในไฟล์
*.aspx ระหว่างภาษา
VB 2010 กับภาษา
VC# 2010



รูปที่ 1-24 (ต่อ)

การอ้างเหตุการณ์
Button1_Click ในไฟล์
*.aspx ระหว่างภาษา
VB 2010 กับภาษา
VC# 2010



วิธีการทดสอบตัวอย่างสคริปต์ของ ASP.NET 4.0

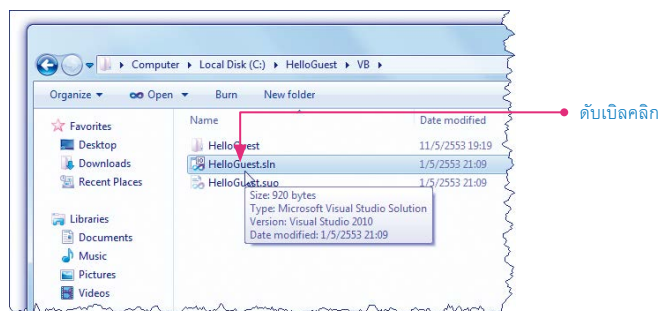
สมมติว่าโปรเจกต์ที่ผู้อ่านสนใจเก็บอยู่ที่พาร์ C:\ มีวิธีทดสอบสคริปต์ ASP.NET 4.0 ที่น่าสนใจอยู่ 2 วิธีคือ

◆ **กรณีที่ 1** โปรเจกต์ดังกล่าวมีไฟล์ Solution (*.sln)

1. ดับเบิลคลิกไฟล์ *.sln เพื่อเปิดโปรเจกต์ชนิด Web Application ใน VS 2010

รูปที่ 1-25

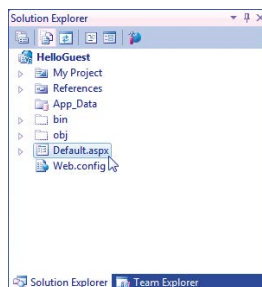
กรณีการเปิด Web
Application ใน
VS 2010



2. แสดงรายชื่อเว็บฟอร์มของโปรเจกต์ปัจจุบันที่แสดงอยู่ในหน้าต่าง Solution Explorer ของ VS 2010 IDE

รูปที่ 1-26

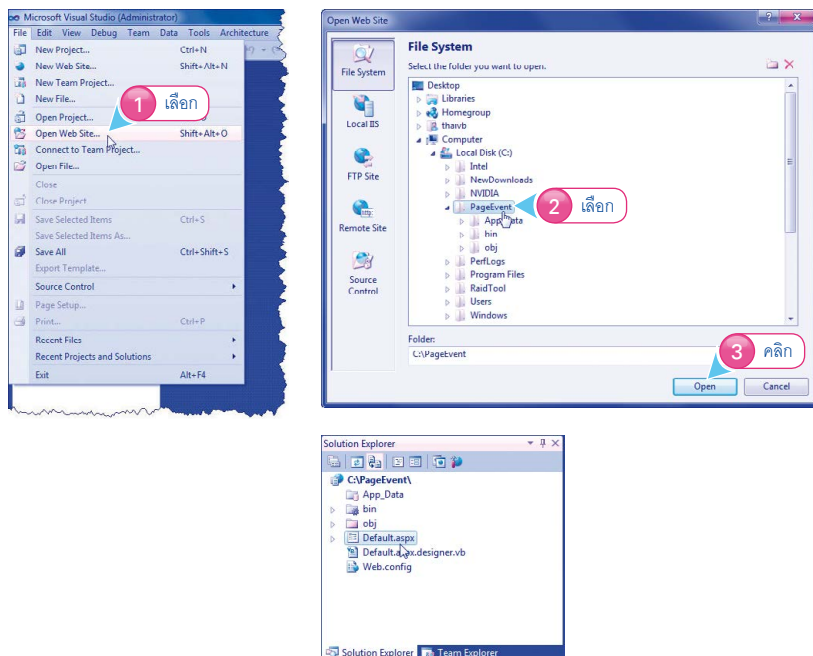
แสดงเว็บฟอร์มของ
โปรเจกต์ปัจจุบัน



- ◆ **กรณีที่ 2** โปรเจกต์ดังกล่าวไม่มีไฟล์ Solution มีแต่รายชื่อไฟล์ของ ASP.NET เช่น *.vb, *.cs, *.aspx ฯลฯ ให้ผู้อ่านเปิด VS 2010 แบบว่างเปล่าขึ้นมาก่อน จากนั้นคลิกเมนู File > Open Web Site... เลือกพารที่จัดเก็บไฟล์ต่างๆ ของ ASP.NET สมมติว่าจัดเก็บอยู่ที่พาร C:\PageEvent

รูปที่ 1-27

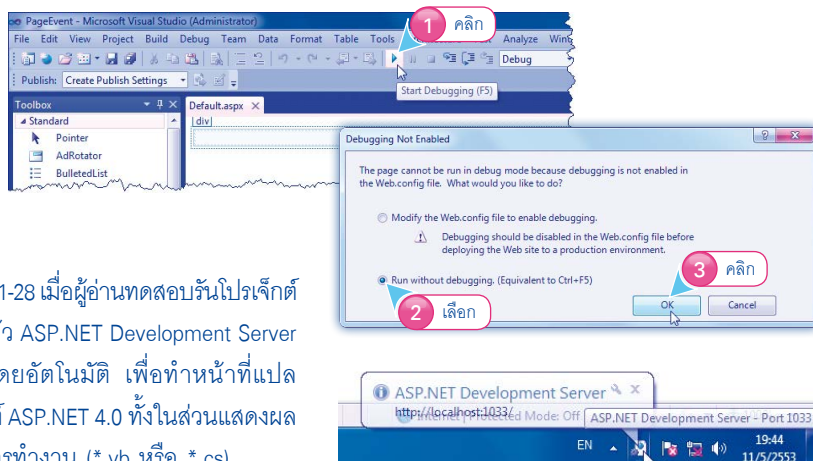
กรณีมีแต่รายชื่อไฟล์
ของ ASP.NET 4.0



การทดสอบสคริปต์ ASP.NET 4.0 ทั้ง 2 วิธีนี้ภายใต้ ASP.NET Development Server โดยที่ไม่จำเป็นต้องอาศัย IIS แต่อย่างใด ซึ่งมีความสะดวกต่อนักพัฒนาเป็นอย่างยิ่ง เพราะผู้อ่านสามารถทดสอบผลการทำงานได้ทันที

รูปที่ 1-28

แสดง ASP.NET
Development
Server ถูกรันขึ้นมา
โดยอัตโนมัติ



จากรูปที่ 1-28 เมื่อผู้อ่านทดสอบรันโปรเจกต์ใน VS 2010 IDE แล้ว ASP.NET Development Server ถูกสั่งให้รันขึ้นมาโดยอัตโนมัติ เพื่อทำหน้าที่แปลชุดคำสั่งของสคริปต์ ASP.NET 4.0 ทั้งในส่วนแสดงผล (*.aspx) และได้การทำงาน (*.vb หรือ *.cs)

การทดสอบสคริปต์ของ ASP.NET 4.0 มี 2 วิธีคือ

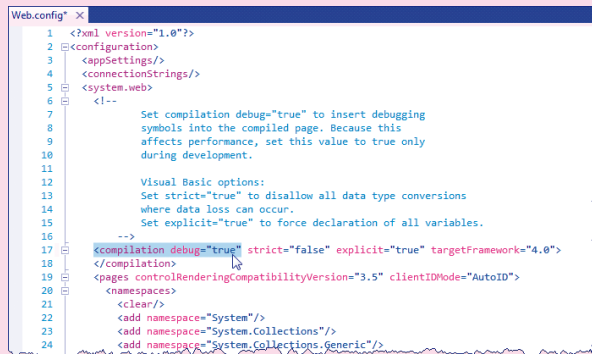
1. **ตัวเลือก Modify the Web.config file to enable debugging** หมายถึง ทดสอบสคริปต์ ASP.NET และได้การทำงานว่า มีข้อผิดพลาดเกิดขึ้นหรือไม่ ถ้ามีให้ไปยังจุดที่เกิดข้อผิดพลาด เพื่อ debug ข้อผิดพลาดดังกล่าวด้วย
2. **ตัวเลือก Run without debugging** หมายถึง ทดสอบสคริปต์ ASP.NET และได้การทำงานว่า มีข้อผิดพลาดเกิดขึ้นหรือไม่ ถ้ามีให้ไปยังจุดที่เกิดข้อผิดพลาด แต่ไม่มีการ debug ข้อผิดพลาดดังกล่าว

Note

ผู้เขียนเลือกตัวเลือกที่ 2 เพื่อทดสอบโปรแกรมที่ไมจำเป็นต้องมีการ debugging แต่อย่างไรก็ตาม ในกรณีที่ผู้อ่านเลือกทดสอบแบบตัวเลือกที่ 1 ส่งผลให้เกิดการแก้ไขค่าของแอตทริบิวต์ debug="true" ในไฟล์ Web.config ผู้อ่านสามารถแก้ไขค่ากลับมาเป็น false เพื่อให้เป็นทางเลือกอย่างเดิมได้เช่นกัน ดังรูปที่ 1-29

รูปที่ 1-29

แสดงค่าของแอตทริบิวต์ debug="true"



```

1 <?xml version="1.0"?>
2 <configuration>
3   <appSettings/>
4   <connectionStrings/>
5   <system.web>
6     <!--
7       Set compilation debug="true" to insert debugging
8       symbols into the compiled page. Because this
9       affects performance, set this value to true only
10      during development.
11
12      Visual Basic options:
13      Set strict="true" to disallow all data type conversions
14      where data loss can occur.
15      Set explicit="true" to force declaration of all variables.
16    -->
17    <compilation debug="true" strict="false" explicit="true" targetFramework="4.0">
18  </compilation>
19  <pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID">
20    <namespaces>
21      <clear/>
22      <add namespace="System"/>
23      <add namespace="System.Collections"/>
24      <add namespace="System.Collections.Generic"/>
  
```

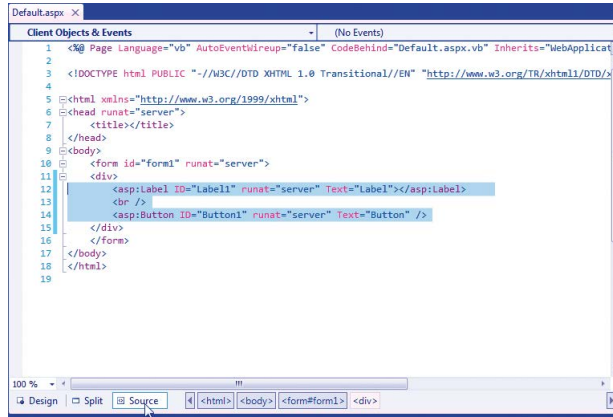
ข้อแตกต่างที่สำคัญของการทดสอบทั้ง 2 วิธีก็คือ ถ้าผู้อ่านมั่นใจว่า สคริปต์ ASP.NET และได้การทำงานของผู้่านทำงานสมบูรณ์ไม่มีอะไรต้องแก้ไขอีกแล้ว ให้เลือกกรนแบบที่ 2 เพราะว่าเป็นต้อง debugging ได้การทำงานแต่อย่างใด

ความสัมพันธ์ระหว่างเว็บคอนโทรลของ ASP.NET กับ ภาษา HTML

เมื่อผู้อ่านใช้งานเว็บคอนโทรลต่างๆ เพื่อสร้างส่วนแสดงผลในเว็บเพจ (*.aspx) เราจะมาดูว่า เว็บคอนโทรลที่ผู้อ่านใช้งานถูกแปลงเป็นภาษา HTML อย่างไร ให้ผู้อ่านดูรูปที่ 1-30 ประกอบคำอธิบาย

รูปที่ 1-30

การใช้งานเว็บ
คอนโทรล Label กับ
เว็บคอนโทรล Button
ในสคริปต์ของ ASP.
NET 4.0



ให้ผู้อ่านคลิกแท็บ Source พบว่า เว็บคอนโทรลที่นำมาใช้งานถูกแปลงเป็นอิลิเมนต์ที่ขึ้นต้นด้วยคำว่า asp เสมอ แตกต่างจากอิลิเมนต์ของภาษา HTML แสดงด้วยชื่ออิลิเมนต์โดยตรง เช่น

- ◆ **อิลิเมนต์
** เป็นของภาษา HTML ทำหน้าที่ขึ้นบรรทัดใหม่ เมื่อพบราเซอร์พบอิลิเมนต์นี้ก็จะขึ้นบรรทัดใหม่ 1 บรรทัดต่อ 1 อิลิเมนต์ ราเซอร์สามารถแปลงหน้าที่อิลิเมนต์ของภาษา HTML ได้โดยตรง
- ◆ **อิลิเมนต์ <asp:Label>...</asp:Label>** เป็นอิลิเมนต์ที่เกิดจากการใช้งานเว็บคอนโทรล Label ของภาษา ASP.NET 4.0 (*.aspx) พบว่ามีแอตทริบิวต์ที่ชื่อว่า runat="server" ทำหน้าที่บอกราเซอร์ว่าอิลิเมนต์นี้รันที่ฝั่ง server
- ◆ **อิลิเมนต์ <asp:Button>...</asp:Button>** เป็นอิลิเมนต์ที่เกิดจากการใช้งานเว็บคอนโทรล Button ของภาษา ASP.NET 4.0

หมายความว่าเว็บคอนโทรลทุกตัวที่ผู้อ่านเรียกใช้งานต้องแปลงเป็นภาษา HTML ที่ฝั่ง server เสียก่อน แล้วค่อยส่งกลับมาให้เบราว์เซอร์แสดงผลต่ออีกทีหนึ่ง

สรุปท้ายบท

ผู้เขียนนำเสนอเนื้อหาเบื้องต้นต่างๆ ที่เกี่ยวข้องกับ ASP.NET 4.0 เพื่อให้ผู้อ่านทราบว่า ผู้อ่านกำลังอยู่ส่วนไหนของ .NET 4.0 Framework