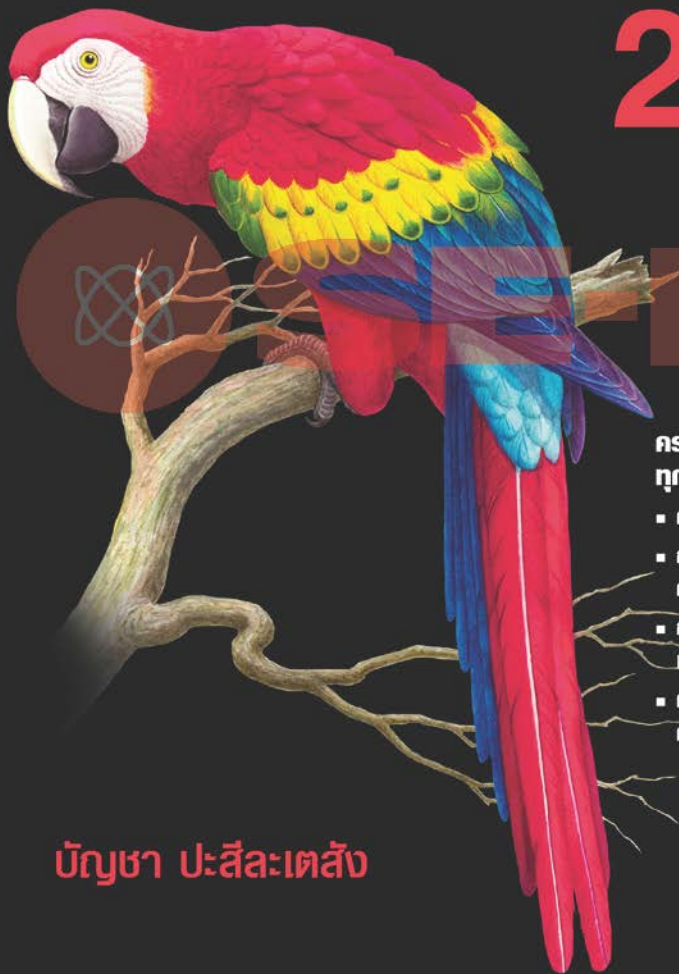


พัฒนาแอปพลิเคชันด้วย

# Visual C#

## 2008



### ครอบคลุมการใช้งานกับ VC# ทุกเวอร์ชัน

- ความรู้พื้นฐานทั่วไปของ Visual C#
- การใช้คอนโทรลตั้งแต่ขั้นพื้นฐานจนถึงคอนโทรลระดับสูง
- การจัดการฐานข้อมูลด้วย ADO.NET และ Crystal Reports
- ตัวอย่างโปรเจกต์: Web#, Text# และตัวอย่างระบบงาน App#

บัญชา ปะสีละเตสัง

# คำนำ



Visual C# นั้นเป็นภาษาที่ผ่านการพัฒนามาแล้วหลายเวอร์ชันนับตั้งแต่เวอร์ชัน 2002, 2003, 2005 และ 2008 ตามลำดับ โดยในแต่ละเวอร์ชันนั้นก็จะมีการปรับปรุงประสิทธิภาพการทำงาน และเพิ่มเติมเครื่องมือที่ช่วยอำนวยความสะดวกในการเขียนโปรแกรมเข้ามาเรื่อยๆ ซึ่งก็จะช่วยให้เรามีแนวทางที่หลากหลายในการพัฒนาโปรแกรมขึ้นด้วยเช่นกัน แต่อย่างไรก็ตาม ถึงแม้สิ่งที่อยู่ใน Visual C# จะดีเยี่ยมเพียงไร แต่หากเราซึ่งเป็นผู้ใช้ขาดความรู้และความเข้าใจการนำไปประยุกต์ใช้งานแล้ว ก็คงจะไม่มีประโยชน์อันใดเลย

ด้วยเหตุนี้ผู้เขียนจึงได้พยายามรวบรวมความรู้ในภาษา Visual C# ที่มีอยู่ทั้งหมด จากประสบการณ์ที่เคยใช้งานมาแล้วหลายเวอร์ชันมาไว้ในหนังสือเล่มนี้ โดยมุ่งเน้นให้ผู้อ่านได้เข้าใจหลักการที่จำเป็นของการใช้ Visual C# ควบคู่ไปกับแนวทางการนำไปประยุกต์ใช้งาน จึงได้มีตัวอย่างโปรแกรมที่สอดคล้องกับเนื้อหาปกติไปเป็นช่วงๆ ซึ่งจะช่วยให้ทักษะการเขียนโปรแกรมให้เพิ่มขึ้นอย่างรวดเร็ว และช่วยให้ผู้อ่านมองเห็นแนวทางในการนำไปประยุกต์ใช้งานได้ชัดเจนมากยิ่งขึ้น

ดังนั้นผู้เขียนจึงหวังเป็นอย่างยิ่งว่า เนื้อหาต่างๆ ที่มีอยู่ในหนังสือเล่มนี้ จะช่วยให้ผู้อ่านเกิดแนวคิดในการนำสิ่งต่างๆ ที่มีอยู่ใน Visual C# มาสร้างสรรค์เป็นแอปพลิเคชันที่สมบูรณ์แบบ และสามารถนำไปใช้งานได้จริง

**บัญชา ปะสีละเต็ล**

banchar\_pa@yahoo.com

<http://www.jobgle.com>

# สารบัญ



<b>ตอนที่ 1 : การติดตั้งและใช้งานทั่วไป .....</b>	<b>17</b>
เกี่ยวกับภาษา C# .....	17
การติดตั้ง VC# 2008 .....	18
การเขียนโปรแกรมแบบ Windows Forms Application .....	22
ส่วนประกอบที่สำคัญของโปรแกรม .....	23
การบันทึกและเปิดโปรเจกต์ .....	26
โค้ดของโปรแกรมและการรันโปรแกรม .....	28
พื้นฐานการเขียนโปรแกรมกับอีเวนต์ .....	29
การปรับแต่ง Options .....	30
การพิจารณาข้อผิดพลาดเบื้องต้น .....	32
<b>ตอนที่ 2 : พื้นฐานของ C# .....</b>	<b>35</b>
คำสั่งวนของ C# .....	35
องค์ประกอบพื้นฐานที่ควรรู้จัก .....	36
บล็อก {...} .....	36
เครื่องหมายสิ้นสุดคำสั่ง(;) .....	37
คำอธิบาย .....	37

การแสดงข้อความด้วย MessageBox .....	38
ชนิดของข้อมูล .....	38
ตัวแปร .....	40
การประกาศค่าของตัวแปร .....	40
การกำหนดค่าให้แก่ตัวแปร .....	40
ค่าคงที่ .....	42
โอเปอเรเตอร์ .....	42
โอเปอเรเตอร์สำหรับการคำนวณ .....	42
โอเปอเรเตอร์สำหรับการกำหนดค่า .....	42
โอเปอเรเตอร์สำหรับการเปรียบเทียบ .....	44
โอเปอเรเตอร์ทางตรรกะ .....	44
Ternary Operator .....	45
ลำดับความสำคัญของโอเปอเรเตอร์ใน C# .....	45
เทคนิคเพิ่มเติมเกี่ยวกับสตริง .....	46
การเชื่อมต่อสตริง .....	46
การขึ้นบรรทัดใหม่ .....	47
การแปลงชนิดข้อมูล .....	48
Control Statement .....	49
คำสั่ง if .....	49
คำสั่ง switch...case .....	54
คำสั่ง for .....	56
คำสั่ง while .....	58
คำสั่ง do...while .....	59
คำสั่ง break .....	59
คำสั่ง continue .....	60
บล็อกและขอบเขตของตัวแปร .....	60
<b>unit 3 : การเขียนโปรแกรมแบบ OOP เบื้องต้น .....</b>	<b>63</b>
คลาส .....	63
โมดิไฟเออร์ .....	65
เมธอด .....	66

เมธอดที่ไม่มีการส่งค่ากลับ .....	67
เมธอดที่มีการส่งค่ากลับ .....	68
การกำหนดโมดิฟายเออร์สำหรับเมธอด .....	70
การเรียกใช้เมธอด .....	70
เทคนิคการกำหนดพารามิเตอร์ .....	73
Value Parameter .....	73
Reference Parameter .....	74
Output Parameter .....	75
Parameter Array .....	75
เมธอดแบบ Overloading .....	76
ออบเจกต์และอินสแตนซ์ .....	77
ฟิลด์ .....	81
พรีอ็อปเพอเรเตอร์ .....	82
สมาชิกแบบสแตติก .....	85
คอนสตรัคเตอร์ .....	87
การสืบทอด .....	90
หลักในการสืบทอด .....	90
โมดิฟายเออร์ <i>protected</i> .....	91
โมดิฟายเออร์ <i>virtual</i> และ <i>override</i> .....	92
โมดิฟายเออร์ <i>sealed</i> .....	94
การใช้คีย์เวิร์ด <i>this</i> และ <i>base</i> .....	96
การใช้คีย์เวิร์ด <i>is</i> และการคาสต์ออบเจกต์ .....	98
Partial Class .....	99
นามสเปซ .....	101
<b>บทที่ 4 : ข้อมูลและคลาสพื้นฐานที่สำคัญของ .NET .....</b>	<b>105</b>
ข้อมูลชนิดอาร์เรย์ .....	105
อาร์เรย์ 1 มิติ .....	105
อาร์เรย์หลายมิติ .....	108
Jagged Array .....	111
ข้อมูลชนิดอินยูเมอเรชั่น .....	116

เนมสเปซของ .NET Framework .....	119
คลาส String .....	121
Numeric Format String .....	123
การแปลง String Number .....	124
<i>การใช้เมธอด Parse()</i> .....	124
<i>การใช้เมธอด TryParse()</i> .....	125
<i>การใช้คลาส Convert</i> .....	126
คลาส Math .....	126
คลาส Random .....	128
คลาส DateTime .....	128
คลาส MessageBox .....	131

## unit 5 : วินโดวส์แอปพลิเคชันและคอนโทรลพื้นฐาน ..... 135

การออกแบบและจัดวางคอนโทรล .....	135
การกำหนดพรีอปรเตอร์ดีของคอนโทรล .....	137
พรีอปรเตอร์พื้นฐานของคอนโทรลทั่วไป .....	138
เมธอด InitializeComponent() .....	142
อีเวนต์ของคอนโทรล .....	143
<i>ตัวจัดการอีเวนต์</i> .....	144
<i>การใช้เครื่องมือสร้างตัวจัดการอีเวนต์</i> .....	146
<i>การลบหรือยกเลิกตัวจัดการอีเวนต์</i> .....	147
<i>พารามิเตอร์ของตัวจัดการอีเวนต์</i> .....	147
คอนโทรล Button .....	150
คอนโทรล Label และ LinkLabel .....	151
คอนโทรล TextBox .....	153
คอนโทรล CheckBox .....	158
คอนโทรล RadioButton และ GroupBox .....	159
คอนโทรลประเภทรายการตัวเลือก .....	162
คอนโทรล ComboBox .....	165
คอนโทรล ListBox .....	169
คอนโทรล CheckedListBox .....	171

<b>unit 6 : คอนโทรลและคอมโพเนนต์ที่น่าสนใจอื่นๆ.....</b>	<b>177</b>
คอนโทรล MonthCalendar และ DateTimePicker .....	177
คอนโทรล NumericUpDown .....	179
คอนโทรล TrackBar .....	182
คอนโทรล ProgressBar .....	186
คอนโทรล TabControl .....	188
การจัดเก็บรูปภาพด้วย ImageList .....	192
คอนโทรล TreeView .....	193
คอนโทรล ListView .....	198
ลักษณะเกี่ยวกับคอมโพเนนต์ .....	208
คอมโพเนนต์ Timer .....	209
คอมโพเนนต์ NotifyIcon .....	212
คอมโพเนนต์ ErrorProvider .....	214
<b>unit 7 : ลักษณะสำคัญอื่นๆ ของ Windows Forms Application .....</b>	<b>217</b>
การสร้างเมนูให้กับฟอร์ม .....	217
การสร้างรายการเมนูและเมนูย่อย .....	219
การกำหนดรูปภาพให้แก่เมนู .....	221
รายการเมนูแบบ ComboBox และ TextBox .....	222
การจัดการอีเวนต์ของเมนู .....	222
การสร้าง Popup Menu .....	223
การสร้างแถบสถานะ (Status Bar) .....	224
แถบสถานะแบบ StatusLabel .....	224
แถบสถานะแบบ ProgressBar .....	225
แถบสถานะแบบ DropDownButton และ SplitButton .....	226
การสร้างทูลบาร์ .....	229
การใช้ปุ่ม Button กับ ToolStrip .....	229
การวางคอนโทรลอื่นๆ บน ToolStrip .....	232
การจัดการอีเวนต์สำหรับ ToolStrip .....	234
การใช้หลายๆ ฟอร์มในโปรแกรม .....	235
เทคนิคการรับส่งข้อมูลระหว่างฟอร์ม .....	237

แอปพลิเคชันแบบ MDI .....	240
การเพิ่มคอนโทรลในขณะรันโปรแกรม .....	245
คอนเท้นเนอร์และคอลเล็กชัน Controls .....	248
อีเวนต์เกี่ยวกับคีย์บอร์ด .....	250
อีเวนต์เกี่ยวกับเมาส์ .....	252
อีเวนต์การ Drag-Drop .....	253

**unit 8 : ตัวอย่างโปรเจกต์: Web# ..... 259**

คอนโทรล WebBrowser .....	259
ลักษณะองค์ประกอบของแอปพลิเคชัน .....	261
แนวคิดในการพัฒนาแอปพลิเคชัน .....	262
การกำหนดและตรวจสอบ URL .....	262
การรับข้อมูล URL จากผู้ใช้ .....	264
เมื่อเกิดอีเวนต์ Navigated .....	265
เมื่อเกิดอีเวนต์ ProgressChanged .....	266
เมื่อเกิดอีเวนต์ StatusTextChanged .....	267
เมื่อเกิดอีเวนต์ DocumentTitleChanged .....	267
เมื่อเกิดอีเวนต์ DocumentCompleted .....	268
เมื่อเกิดอีเวนต์ NewWindow .....	268
การสืบค้นข้อมูล .....	270
การควบคุมปุ่มอื่นๆ และเมนู .....	272
การออกแบบและพัฒนาแอปพลิเคชันที่สมบูรณ์ .....	274

**unit 9 : การตรวจสอบและจัดการข้อผิดพลาด ..... 285**

การใช้งาน Debugging Tools .....	285
ตั้ง Break Point .....	286
ทดสอบ Step Into .....	287
ทดสอบ Step Over .....	288
ทดสอบ Step Out .....	288
การตรวจสอบค่าของตัวแปร .....	289



ลักษณะของการเกิดข้อผิดพลาด .....	289
Exception Handling .....	291
การใช้คำสั่ง finally .....	293
การใช้คำสั่ง throw .....	295

## บทที่ 10: การวาดกราฟิกด้วย GDI+ ..... 299

คลาสพื้นฐานทางด้านกราฟิก .....	299
คลาส Point .....	299
คลาส Pen .....	300
คลาส SolidBrush .....	301
คลาส Rectangle .....	301
การสร้างออบเจกต์ Graphics สำหรับการวาด .....	301
การวาดเส้นแบบต่างๆ .....	303
การวาดเส้นตรง .....	303
การวาดเส้นตรงหลายเส้น .....	304
การวาดเส้นโค้ง .....	304
การวาดเส้นโค้งแบบ Bezier .....	305
การวาดรูปทรงแบบต่างๆ .....	306
การวาดรูปสี่เหลี่ยม .....	306
การระบายสีลงในรูปสี่เหลี่ยม .....	307
การวาดวงกลมและวงรี .....	308
การวาดรูปหลายเหลี่ยม (Polygon) .....	309
การวาดส่วนโค้ง .....	309
เทคนิคการวาดกราฟิกโดยใช้เมาส์ .....	310
การวาดสตริง .....	316
การวาดรูปภาพ .....	317
การโหลดภาพด้วยคลาส Image, Bitmap และ Icon .....	318
การย่อ/ขยายรูปภาพ .....	319
การหมุนรูปภาพ .....	321

**บทที่ 11: ไลอ็อกและเทคนิคการพิมพ์ ..... 327**

ColorDialog .....	327
FontDialog .....	330
FolderBrowserDialog .....	333
OpenFileDialog .....	336
SaveFileDialog .....	341
เทคนิคการพิมพ์ .....	345
หลักการเบื้องต้นของการพิมพ์ .....	345
การพิมพ์ข้อความหลายบรรทัด .....	348
การพิมพ์หลายๆ หน้า .....	350
การพิมพ์รูปภาพ .....	352
การตั้งค่าหน้ากระดาษ (Page Setup) .....	353
การกำหนดช่วงการพิมพ์ .....	355
การแสดงตัวอย่างก่อนพิมพ์ (Print Preview) .....	356

**บทที่ 12: ระบบไฟล์และสตรีม ..... 361**

การอ้างถึงตำแหน่งพาท .....	361
คลาส Path .....	363
คลาส Directory .....	365
คลาส File .....	368
คลาส StreamReader .....	370
คลาส StreamWriter .....	372
คลาส FileStream .....	375
คลาส BinaryReader และ BinaryWriter .....	376
คลาส BinaryWriter .....	377
คลาส BinaryReader .....	377

**บทที่ 13: ตัวอย่างโปรแกรม: Text# ..... 379**

คอนโทรล RichTextBox .....	379
องค์ประกอบของโปรแกรม Text# .....	381
แนวทางการเขียนโปรแกรม .....	383

ข้อกำหนดเบื้องต้น .....	383
การตรวจสอบการเปลี่ยนแปลงของเอกสาร .....	384
การบันทึกข้อมูลแบบ Save และ Save As .....	385
การเปิดไฟล์ .....	387
การสร้างเอกสารใหม่ .....	388
การออกจากโปรแกรม .....	389
การค้นหา .....	390
การแทนที่ .....	393
การเขียนโปรแกรมสำหรับส่วนอื่นๆ .....	395
การสร้างโปรเจกต์ Text# .....	396

#### บทที่ 14: การจัดการฐานข้อมูลด้วย ADO.NET ..... 411

เกี่ยวกับ ADO.NET .....	411
การเชื่อมต่อกับ Access ด้วย OLEDB .....	412
การเชื่อมต่อกับ SQL Server .....	417
การเชื่อมต่อกับ SQL Server ผ่าน OLEDB .....	417
การเชื่อมต่อกับ SQL Server ด้วย SQL Managed Provider .....	418
การแสดงผลข้อมูลจาก DataSet .....	420
การแสดงผลข้อมูลด้วย DataGridView .....	422
การทำ Data Binding .....	424
การอ่านข้อมูลด้วย DataReader .....	429
การอ่านข้อมูลชนิดรูปภาพ .....	431
การกรองข้อมูลด้วย DataView .....	435
Master-Details Form .....	436

#### บทที่ 15: การใช้ SQL และพารามิเตอร์ ..... 441

ความรู้เพิ่มเติมเกี่ยวกับ Command .....	441
การกำหนดคำสั่ง SQL .....	441
การประมวลผลคำสั่ง SQL .....	442
คำสั่ง SELECT .....	445
การคำนวณข้อมูลในคอลัมน์ .....	445

การป้องกันการเลือกซ้ำด้วย DISTINCT .....	446
คำสั่ง WHERE .....	446
โอเปอเรเตอร์ในการเปรียบเทียบ .....	447
การใช้วงเล็บ () และ AND/OR ร่วมกัน .....	448
การใช้ BETWEEN...AND และ NOT BETWEEN...AND .....	450
การใช้ IN และ NOT IN .....	450
การใช้ LIKE และ Wildcard .....	451
การเรียงลำดับผลลัพธ์ด้วย ORDER BY .....	452
การจัดกลุ่มข้อมูลด้วย GROUP BY .....	452
ฟังก์ชันในการสรุปผล (Aggregate Functions) .....	453
ข้อมูลชนิดวันเวลา .....	455
การใช้ข้อมูลจากหลายตาราง .....	455
การอ้างถึงตารางและคอลัมน์ .....	455
การคัดเลือกข้อมูลโดยเทียบจาก WHERE clause .....	456
LEFT JOIN .....	457
การปรับปรุงข้อมูล .....	458
การเพิ่มข้อมูล .....	458
การแก้ไขข้อมูล .....	459
การลบข้อมูล .....	459
ข้อควรระวังเกี่ยวกับกฎ Referential Integrity .....	460
ADO.NET กับการปรับปรุงข้อมูล .....	461
การใช้ Parameter .....	462
การเพิ่มข้อมูลรูปภาพลงในตาราง .....	465

## บทที่ 16: การสร้างรายงานด้วย Crystal Reports ..... 471

เริ่มต้นการใช้ Crystal Reports .....	471
การจัดวางฟิลด์ข้อมูล .....	474
การปรับแต่งลักษณะการแสดงผล .....	476
การเพิ่มเติมรูปแบบด้วยออบเจกต์ของ Crystal Reports .....	480
รายงานที่มีหลายหน้า .....	482
การแบ่งกลุ่มข้อมูล .....	483

การแสดงผลรูป .....	486
การนับจำนวน .....	486
การหาผลรวม .....	489
การใช้คำสั่ง SQL และพารามิเตอร์ .....	489
การใช้คอนโทรล CrystalReportViewer .....	492
การส่งพารามิเตอร์จาก C# ไปยัง Crystal Reports .....	493

## unit 17: ตัวอย่างระบบงาน: App# ..... 499

ภาพรวมของแอปพลิเคชัน .....	499
ฐานข้อมูลสำหรับโปรเจกต์ .....	503
ส่วนของข้อมูลลูกค้า .....	504
ส่วนของข้อมูลสินค้า .....	513
ส่วนของข้อมูลการสั่งซื้อ .....	522
ส่วนของการทำงานใบส่งสินค้า (Invoice) .....	528
รวมทุกส่วนเข้าด้วยกันใน MDI Form .....	539

## unit 18: การสร้างชุดติดตั้งและแจกจ่ายโปรแกรม ..... 543

การสร้างชุดติดตั้งเบื้องต้น .....	543
พรีอปเพอริตี้ของการติดตั้ง .....	550
การกำหนดไฟล์ใน File System .....	552
การเพิ่มเติมส่วน User Interface .....	554
การเพิ่มเติมไดอะล็อก .....	556
ทดสอบการติดตั้ง .....	557

## unit 19: แอปพลิเคชันแบบ WPF ..... 559

ลักษณะของแอปพลิเคชันแบบ WPF .....	559
สคริปต์ XAML .....	561
การจัดวางคอนโทรลบนคอนเทนเนอร์ .....	562
คอนเทนเนอร์แบบ Grid .....	562
คอนเทนเนอร์แบบ StackPanel .....	563
การใช้หลายๆ คอนเทนเนอร์ .....	564



พรีอปเทอร์ดีและแอดตริบิวต์ .....	565
การสร้างตัวจัดการอีเวนต์ .....	565
การเขียนโปรแกรมกับคอนโทรลของ WPF .....	567

**บทที่ 20: การใช้ C# กับเว็บแอปพลิเคชัน ASP.NET ..... 573**

เริ่มต้นการสร้างแอปพลิเคชัน ASP.NET .....	573
Web Form และ Server Controls .....	575
การเขียนโปรแกรมกับ Server Controls .....	577
การจัดส่งอีเมล .....	580
การเก็บข้อมูลแบบ Cookies .....	583
การจัดเก็บข้อมูลแบบ Session .....	586
การเชื่อมต่อกับฐานข้อมูล .....	589
การแสดงผลข้อมูลด้วย GridView .....	590



SE-ED

# การติดตั้ง และการใช้งานทั่วไป



Visual C# หรือ VC# เป็นชุดเครื่องมือในการพัฒนาแอปพลิเคชันด้วยภาษา C# โดยมีการปรับปรุงแก้ไขมาแล้วหลายเวอร์ชันตั้งแต่ปี ค.ศ. 2002, 2003, 2005 และ 2008 ตามลำดับ โดยในแต่ละเวอร์ชันก็ได้มีการเพิ่มความสามารถและเครื่องมือที่จะช่วยอำนวยความสะดวกในการเขียนโปรแกรมเข้ามาเรื่อยๆ และการเปลี่ยนแปลงที่สำคัญคือการปรับปรุงรูปแบบวิธีการเขียนโปรแกรมให้ง่ายและสะดวกสบายยิ่งกว่าเดิม ดังนั้นแม้เราจะเพิ่งเริ่มต้นศึกษาการเขียนโปรแกรม ก็สามารถที่จะเรียนรู้ได้อย่างรวดเร็ว แต่ก่อนที่จะเข้าสู่เนื้อหาของ การเขียนโปรแกรมจริง ในบทนี้จะกล่าวถึงลักษณะการใช้งานเบื้องต้นขององค์ประกอบต่างๆ ที่มีใน VC# ก่อน โดยเริ่มตั้งแต่การติดตั้งไปจนถึงการปรับแต่งสภาพแวดล้อมต่างๆ และลักษณะอื่นๆ ที่จำเป็นต้องรู้ก่อนเข้าสู่การเขียนโปรแกรมจริงในบทต่อไป

## เกี่ยวกับภาษา C#

C# เป็นภาษาคอมพิวเตอร์ที่พัฒนาโดยบริษัท ไมโครซอฟต์ โดยใช้รากฐานของภาษา C/C++ เป็นหลัก ดังนั้นรูปแบบโครงสร้างทางภาษาของ C# จึงคล้ายกับ C/C++ แต่ได้ลดความสลับซับซ้อนลง นอกจากนี้ก็ยังได้แก้ไขปัญหาและข้อบกพร่องหลายประการที่มีอยู่ใน C++ ให้หมดไป จึงทำให้ภาษา C# นั้นกลายเป็นภาษาที่เรียนรู้ได้ง่าย และสามารถทำงานได้อย่างมีประสิทธิภาพ

เนื่องจาก C# นั้นเกิดขึ้นมาพร้อมกับเทคโนโลยี .NET ดังนั้นการทำงานของ C# จึงขึ้นกับ .NET Framework เป็นหลัก โดยมีชุดเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันด้วยภาษา C# เรียกว่า Visual C# ซึ่งสามารถใช้ในการพัฒนาแอปพลิเคชันทั่วๆ ไปในระดับเดียวกับ Visual Basic เพื่อให้โปรแกรมเมอร์ที่คุ้นเคยกับรูปแบบโครงสร้างในแบบ C/C++ สามารถเลือกใช้ VC# แทน VB ได้

ในปัจจุบันภาษา C# กำลังได้รับความนิยมมากขึ้นเรื่อยๆ โดยเฉพาะอย่างยิ่งโปรแกรมเมอร์ที่เคยเขียนด้วยภาษาอื่นๆ เช่น C++, Java, Delphi หรือแม้กระทั่งผู้ที่เคยใช้ VB มาก่อนก็ตาม ต่างก็หันมาใช้ C# กันมากขึ้น เพราะโครงสร้างของ C# นั้น สั้น, กระชับ และเข้าใจได้ง่ายกว่า ซึ่งไม่เพียงแต่จะใช้สร้างแอปพลิเคชันบน Windows เท่านั้น แต่ยังสามารถสร้างแอปพลิเคชันอื่นๆ ได้อีกหลากหลาย เช่น Web Application (ASP.NET), Smart Device, WPF, Silverlight เป็นต้น ดังนั้นการศึกษาเรียนรู้ภาษา C# จึงมีช่องทางให้เรานำไปใช้งานได้มากมายและนับว่าคุ้มค่าเป็นอย่างยิ่ง

## การติดตั้ง VC# 2008

การติดตั้ง VC# 2008 มีให้เลือก 2 รูปแบบคือ

- **แบบติดตั้งเฉพาะ VC# อย่างเดียว** หรือเรียกว่า VC# 2008 Express Edition แต่การติดตั้งแบบนี้ จะใช้งานได้ไม่สมบูรณ์ จึงเหมาะกับการศึกษาเรียนรู้ในเบื้องต้นเท่านั้น
- **แบบติดตั้งจากชุด Visual Studio 2008** ซึ่งการติดตั้งแบบนี้จะทำให้เราสามารถใช้เครื่องมือต่างๆ ได้อย่างครบสมบูรณ์ (ยกเว้น Visual Studio Express Edition) ซึ่งถ้าเราจะใช้ Visual Studio จริงๆ ก็ควรใช้ Professional Edition ไปเลยดีกว่า แต่รุ่นนี้ไม่มีให้ดาวน์โหลด ต้องซื้อชุดจริงมาติดตั้งเท่านั้น

สำหรับในที่นี้จะกล่าวถึงเฉพาะรูปแบบการติดตั้งและใช้งานผ่านชุด Visual Studio เท่านั้น แต่ผู้ที่ใช้ VC# Express ก็สามารถนำไปใช้ได้เกือบทั้งหมด ยกเว้นในบทหลังๆ บางบทเท่านั้น

สำหรับความต้องการของระบบสำหรับ Visual Studio 2008 Professional มีดังนี้

Processor	1.6 GHz or faster
Operating System	Windows XP SP2, Vista all editions (except Starter Edition), Windows Server 2003 SP1 or R2
RAM	384 MB (768 for Vista)
Disk Space	2.2 GB
Drive	DVD ROM

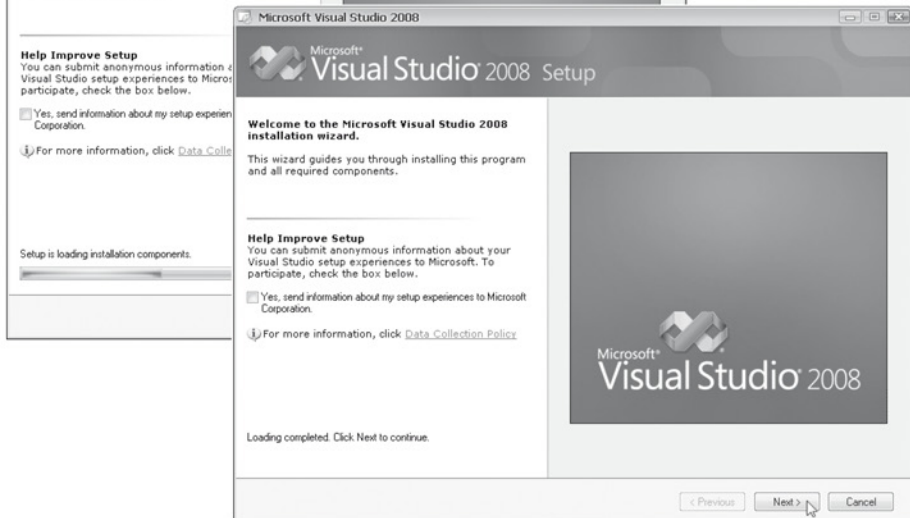
สำหรับขั้นตอนการติดตั้งที่น่าสนใจมีดังนี้

1. Visual Studio 2008 Professional Edition จะอยู่ในรูปแบบ DVD ดังนั้นเครื่องที่จะติดตั้งต้องสามารถอ่านแผ่น DVD ได้ หลังใส่แผ่นเข้าไป ก็จะเข้าสู่การติดตั้งทันที ตามรูปต่อไปนี่

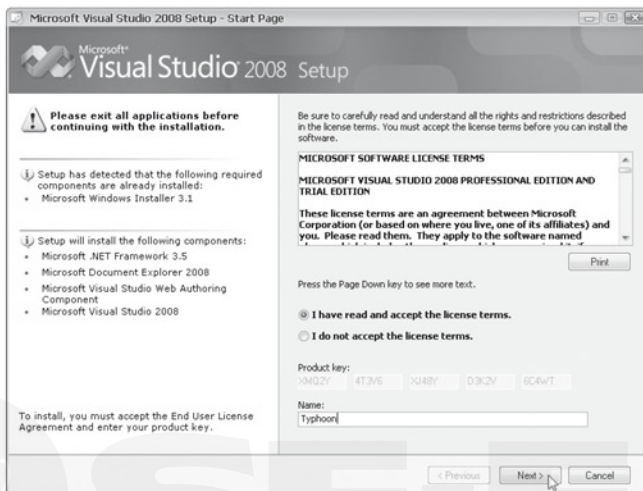




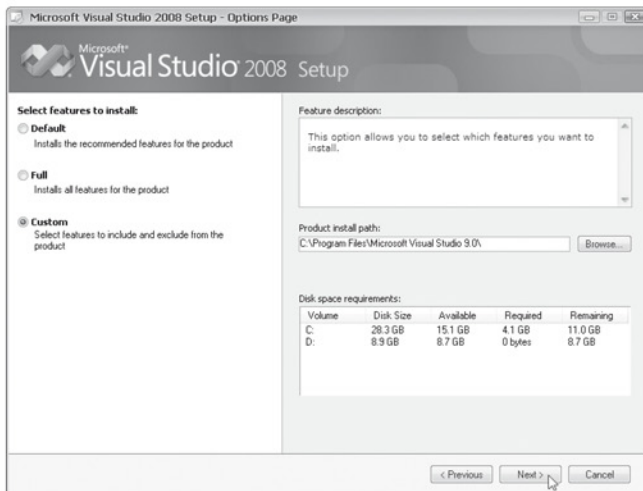
2. จากนั้นจะเข้าสู่การโหลดคอมโพเนนต์สำหรับการติดตั้ง หลังโหลดเสร็จ ให้เลือกตัวเลือก I have read and accept ... แล้วคลิกปุ่ม Next เพื่อไปยังขั้นตอนต่อไป



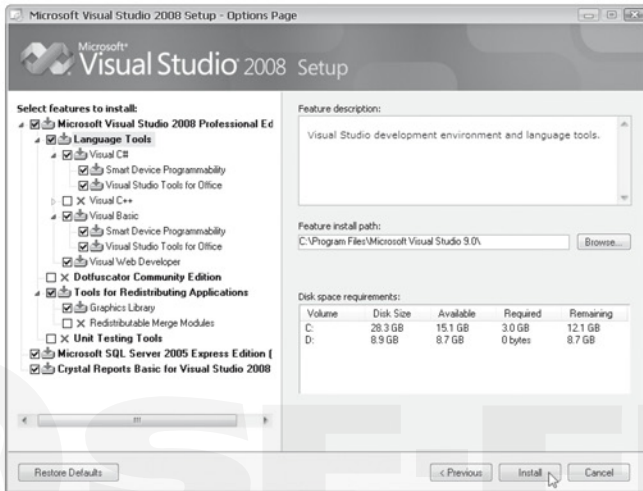
3. หลังโหลดเสร็จ ให้เลือกตัวเลือก I have read and accept ... แล้วคลิกปุ่ม Next เพื่อไปยังขั้นตอนต่อไป



4. เมื่อเข้าสู่ขั้นตอนการเลือกรูปแบบการติดตั้ง แนะนำให้เลือกแบบ Custom เพื่อให้เราสามารถกำหนดสิ่งที่จะติดตั้งลงไปเองได้ แล้วคลิกปุ่ม Next



5. เมื่อมาถึงการเลือกสิ่งที่จะติดตั้ง สิ่งสำคัญที่เราต้องเลือกอยู่แล้วคือ Visual C# นอกจากนี้สิ่งที่เราเลือกเพิ่มเติม เช่น SQL Server Express, Crystal Reports เป็นต้น ซึ่งสิ่งเราเลือกในขั้นตอนนี้เป็นเพียงบางส่วนเท่านั้น ยังมีส่วนที่ถูกต้องเพิ่มเติมเข้าไปในขณะติดตั้งอีกหลายอย่าง แล้วคลิกปุ่ม Install



6. จากการเลือกในข้อที่ 5 ก็จะเข้าสู่ขั้นตอนการติดตั้ง ไปจนกว่าจะแล้วเสร็จ และคลิกปุ่ม Finish



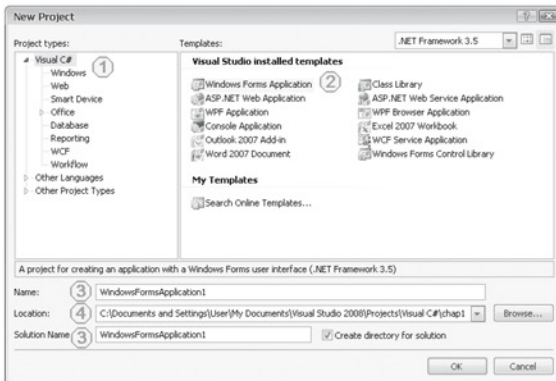


7. ถ้าต้องการเพิ่มหรือลบคอมโพเนนต์อื่นๆ ในภายหลัง ให้เริ่มทำเหมือนกับการติดตั้งใหม่

## การเขียนโปรแกรมแบบ Windows Forms Application

ในขั้นตอนการสร้าง, ออกแบบและพัฒนาโปรแกรม จะเรียกชุดองค์ประกอบนี้ว่า "โปรเจกต์ (Project)" ซึ่งภายในโปรเจกต์เราสามารถสร้างโปรแกรมที่จะนำไปใช้งานหรือแอปพลิเคชันได้หลายแบบ เช่น Windows Forms, WPF, Console Application, Web Application (ASP.NET) เป็นต้น แต่ในที่นี้จะกล่าวถึงเฉพาะ Windows Forms Application เพื่อมุ่งสู่การเขียนโปรแกรมแบบ Visual เป็นหลัก ส่วน WPF และ Web จะกล่าวถึงในบทที่ 19 และ 20 ตามลำดับ

การสร้าง Windows Forms Application มีขั้นตอนดังนี้ โดยเริ่มจากเมื่อเปิดเข้าสู่ Visual Studio หรือ VC# Express แล้วเลือกที่เมนู File > New > Project ... จะได้ดังรูปต่อไปนี้

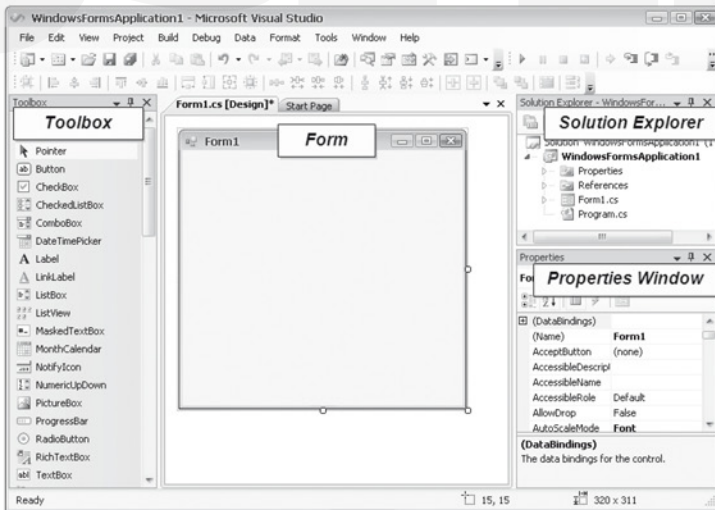


1. ที่ช่องเลือกชนิดโปรเจกต์ (Project types) ให้เลือก Visual C# หรือ Windows
2. ที่ช่อง Templates ให้เลือก Windows Forms Application
3. กำหนดชื่อโปรเจกต์ (Name) ซึ่งเราจะตั้งชื่ออะไรก็ได้ แต่ควรสื่อความหมายว่าโปรเจกต์นั้นเกี่ยวกับอะไร และปกติแล้วชื่อโปรเจกต์กับชื่อ Solution เรามักตั้งเป็นชื่อเดียวกัน
4. ช่อง Location เป็นการกำหนดตำแหน่งที่จัดเก็บโปรเจกต์ ทั้งนี้โปรแกรมจะกำหนดตำแหน่งที่เป็นค่า default ให้อยู่แล้ว ซึ่งถ้าเราไม่ชอบตำแหน่งนี้ก็เปลี่ยนโดยคลิกที่ปุ่ม Browse... สำหรับในที่นี้ผู้เขียนได้สร้างไดเรกทอรีเพิ่มขึ้นมาจากตำแหน่งดีฟอลต์คือ Visual C#\chapX สำหรับเก็บโปรเจกต์ที่เป็น VC# และแยกในแต่ละบทออกจากรัน

หลังจากกำหนดค่าเสร็จเรียบร้อยแล้วคลิกปุ่ม OK ก็จะเข้าสู่หน้าจอหลักของการสร้างโปรเจกต์ ดังรายละเอียดในหัวข้อถัดไป


## ส่วนประกอบที่สำคัญของโปรแกรม

เมื่อเราเข้าสู่โปรแกรมจะมีลักษณะดังรูปต่อไป ซึ่งเป็น Visual Studio แต่ถึงแม้เราจะใช้ VC# Express ก็จะเป็นแบบเดียวกัน



จากรูปส่วนประกอบหลัก 4 อย่างมีรายละเอียดที่น่าสนใจ ดังนี้

## Toolbox

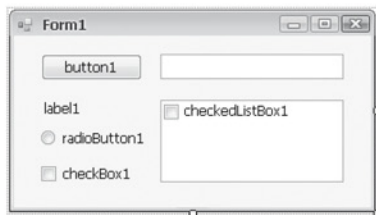
ทูลบ็อกซ์ (ถ้าทูลบ็อกซ์ไม่ปรากฏให้คลิกที่ปุ่ม  บนทูลบาร์หรือเลือกจากเมนู View > Toolbox) เป็นส่วนที่ใช้ในการแสดงคอนโทรลต่างๆ ที่จะนำมาวางบนฟอร์ม แต่เนื่องจากคอนโทรลมีเป็นจำนวนมาก ดังนั้นจึงได้มีการแบ่งออกเป็นกลุ่มๆ เช่น Common Controls, Data เป็นต้น โดยแต่ละกลุ่มจะแยกด้วยแท็บ หากเราต้องการใช้คอนโทรลในกลุ่มใด ก็คลิกที่แท็บของกลุ่มนั้น เพื่อขยายออกดังรูป



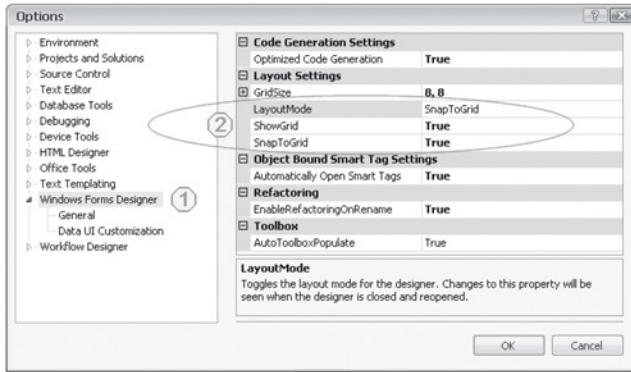
## Form

ฟอร์ม (ถ้าฟอร์มไม่ปรากฏให้เลือกจากเมนู View > Designer) เป็นส่วนประกอบพื้นฐานที่สำคัญอย่างยิ่งของการพัฒนาแอปพลิเคชันแบบ Visual เพราะต้องใช้ในการจัดวางคอนโทรลต่างๆ เพื่อติดต่อกับผู้ใช้

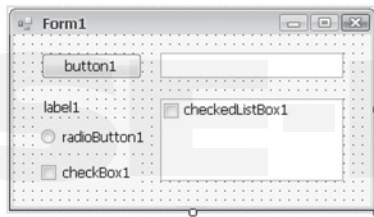
เช่น หากเราเลือกคอนโทรลบางตัวจากทูลบ็อกซ์มาวางลงบนฟอร์ม จะได้ดังนี้



ทั้งนี้การวางคอนโทรลลงบนฟอร์มนั้น หากไม่มีเส้นกริดช่วยเราจะกะตำแหน่งค่อนข้างยาก เพราะโปรแกรมไม่ได้กำหนดให้แสดงเส้นกริดเป็นค่าเริ่มต้นไว้ให้ แต่ถ้าเราต้องการก็สามารถสั่งให้แสดงได้ โดยเลือกจากเมนู Tool > Options... เมื่อปรากฏหน้าจอ Options ในกรอบด้านซ้ายให้เลือก Windows Forms Designer แล้วรายการในกรอบด้านขวาให้กำหนด LayoutMode เป็น SnapToGrid และ ShowGrid เป็น True ดังรูปด้านบนในหน้าถัดไป




แล้วในการเปิดโปรเจกต์ครั้งต่อไป ฟอรัมจะแสดงกริดให้เราเห็นตลอด ดังเช่นรูปต่อไปนี้




สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับฟอรัม เราจะได้ศึกษาเพิ่มเติมเรื่อยๆ ในบทต่อไป

## Solution Explorer

Solution Explorer (ถ้าไม่ปรากฏให้คลิกที่ปุ่ม  บนทูลบาร์หรือเลือกจากเมนู View > Solution Explorer) เป็นส่วนที่แสดงองค์ประกอบต่างๆ ที่มีในโปรเจกต์ สำหรับส่วนนี้เราจะได้ใช้งานในบางกรณี เช่น ดูรายละเอียดของส่วนประกอบบางอย่างที่อยู่ในโปรเจกต์ หรือใช้สลับไปมาระหว่างมุมมองการออกแบบและโค้ดของโปรแกรมตามที่จะกล่าวถึงในลำดับต่อไป

## Properties Window

Properties Window (ถ้าไม่ปรากฏให้คลิกที่ปุ่ม  บนทูลบาร์หรือเลือกจากเมนู View > Properties Window) เป็นส่วนที่ใช้ในการแสดงคุณสมบัติของคอนโทรลหรือคอมโพเนนต์ต่างๆ ที่วางอยู่บนฟอรัม เช่น ขนาด, ตำแหน่ง, สี, ฯลฯ โดยเราสามารถปรับคุณสมบัติแต่ละอย่างให้เป็นไปตามที่เราต้องการได้

สำหรับรายละเอียดการใช้ Properties Window จะกล่าวถึงอีกครั้งในเรื่องคอนโทรลในบทต่อไป

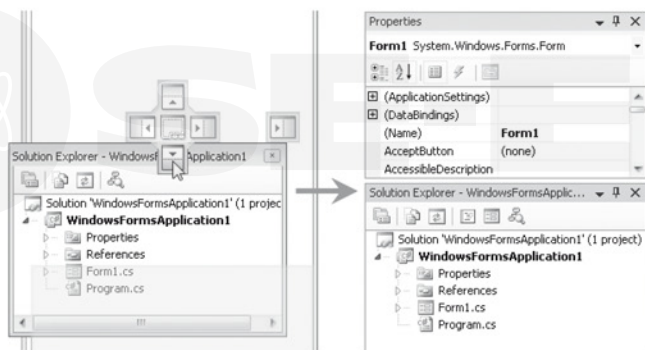
## การปรับย้ายตำแหน่ง

สำหรับแต่ละส่วนที่ได้กล่าวมานี้ ได้แก่ Toolbox, Solution Explorer และ Properties Window การจัดวางตำแหน่งที่โปรแกรมกำหนดไว้ให้ก็จะเป็นตัวตั้งต้นที่นำเสนอไปแล้ว แต่หากเราต้องการย้ายไปจัดวางใหม่แบบที่เราต้องการก็ให้ทำดังนี้

คลิกที่ไตเติลบาร์ของส่วนนั้นค้างไว้ แล้วลากออกมาตำแหน่งเดิมของมัน จะปรากฏภาพเป็นจุดให้เราเลือกที่จะวางจุดไหน ในลักษณะดังนี้




ถ้าเราจะวางจุดไหนก็นำเมาส์ที่เราลากไตเติลบาร์ของส่วนนั้นมาวางที่ขงที่ต้องการ เช่น ถ้าลาก Solution Explorer มาวางที่ช่องด้านล่าง จะส่งผลให้ Solution Explorer ย้ายมาอยู่ข้างล่าง Properties Window แทน ดังรูป



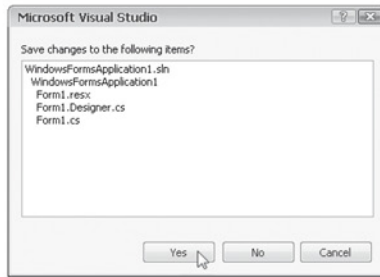
## การบันทึกและเปิดโปรเจกต์

โดยปกติโปรเจกต์ของ VC# ที่เราสร้างขึ้นจะไม่ได้ประกอบด้วยไฟล์เพียงไฟล์เดียว แต่องค์ประกอบต่างๆ จะถูกแยกย่อยออกจากกันไปอยู่กันคนละไฟล์ บางครั้งจึงมักเกิดปัญหาเมื่อเราไปแก้ไขที่ส่วนใดส่วนหนึ่งของโปรเจกต์ แล้วบันทึกการเปลี่ยนแปลงไม่ครบทุกส่วน เพื่อเป็นการแก้ปัญหาเหล่านี้ เมื่อเราต้องการบันทึกโปรแกรมจึงได้เตรียมทางเลือกไว้คือ

- บันทึกเฉพาะส่วนที่ถูกเปิดอยู่ในขณะให้เลือกที่ปุ่มทูลบาร์  หรือกด <Ctrl + S>
- บันทึกการเปลี่ยนแปลงทุกไฟล์ในโปรเจกต์ให้เลือกที่ปุ่มทูลบาร์ 

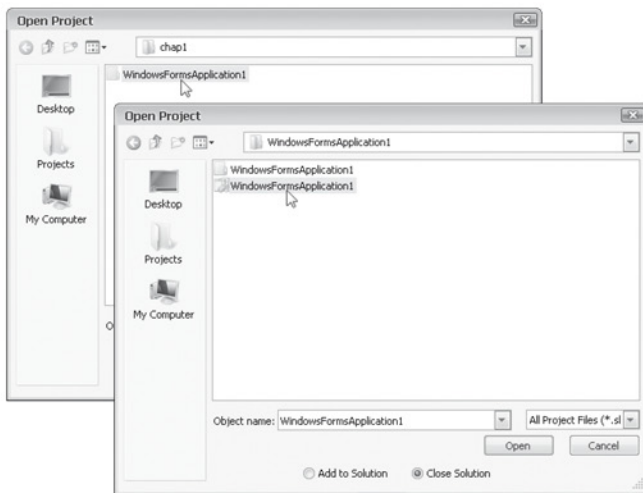


- หรือถ้าเราเปิดโปรเจกต์โดยยังไม่ได้บันทึกการเปลี่ยนแปลง โปรแกรมจะแสดงไดอะล็อกแจ้งให้ทราบว่ามีส่วนใดบ้างที่เกิดเปลี่ยนแปลงที่เรายังไม่ได้บันทึก ดังรูปต่อไป ถ้าเราจะบันทึกก็ให้ตอบ Yes



## การเปิดโปรเจกต์

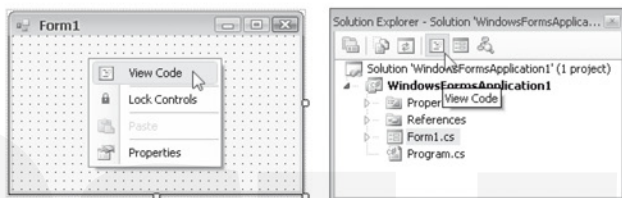
ถ้าจะเปิดโปรเจกต์ที่ได้บันทึกเอาไว้แล้ว ให้เลือกจากเมนู File > Open > Project/Solution... หลังจากนั้นเมื่อไดอะล็อกปรากฏขึ้น ให้เราเลือกตำแหน่งที่เราได้จัดเก็บโปรเจกต์เอาไว้ หากเราเปิดโปรเจกต์เป็นครั้งแรก ตำแหน่งที่โปรแกรมเลือกไว้ให้อาจจะยังไม่ถูกต้อง ก็ให้เราคลิกค้นหาจากตำแหน่งที่ได้จัดเก็บโปรเจกต์เอาไว้ เช่น กรณีของผู้เขียนเก็บโปรเจกต์ที่ 1 ไว้ที่ C:\Document and Settings\...\My Documents\Visual Studio 2008\Projects\Visual C#\chap1 ถ้าคลิกเข้าไปจะเป็นชื่อโปรเจกต์ที่เราได้สร้างขึ้น ให้ดับเบิลคลิกเข้าไปในโฟลเดอร์นั้น จะพบชื่อ Solution ให้ดับเบิลคลิกที่ไฟล์ Solution นั้น เพื่อเปิดโปรเจกต์




## กิตติของโปรแกรมและการรับโปรแกรม

การที่โปรแกรมจะทำงานตามข้อกำหนดของเราได้ ต้องอาศัยการเขียนโค้ดเป็นตัวควบคุม แต่ในบทนี้เราจะยังไม่ลงลึกไปถึงรายละเอียดการเขียนโปรแกรม เพียงแต่จะแนะนำว่าหากเราจะเขียนโค้ดของโปรแกรมจะต้องทำอย่างไร เพื่อเป็นแนวทางในการนำไปใช้ในบทต่อๆ ไป ทั้งนี้หากเราจะเขียนโค้ด หลังเข้าสู่โปรเจกต์ ก็อาจใช้วิธีใดวิธีหนึ่งต่อไปนี้

- กด <F7>
- หรือคลิกขวาที่ฟอร์มแล้วเลือก View Code
- หรือจะเลือกจากปุ่มบน Solution Explorer



และหากต้องการเปลี่ยนกลับมาที่มุมมองออกแบบ (ฟอร์ม) ก็สามารถทำได้หลายวิธี เช่น

- กด <Shift + F7>
- หรือขณะที่อยู่ในมุมมองของโค้ดให้คลิกขวาแล้วเลือก View Designer
- หรือเลือกจากปุ่ม  บน Solution Explorer
- หรือเราสามารถสลับไปมาระหว่างมุมมองออกแบบและโค้ดโดยเลือกจากแท็บ ดังรูปด้านขวามือ



เมื่อเราเปิดเข้าสู่มุมมองการเขียนโค้ดของโปรแกรม จะมีโค้ดบางส่วนที่โปรแกรมสร้างไว้ให้ล่วงหน้าแล้ว ดังนี้

```

Start Page  Form1.cs [Design]*  Form1.cs*
Form1
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1 {
    public partial class Form1 : Form {
        public Form1() {
            InitializeComponent();
        }
    }
}

```

## การรันโปรแกรมและการยกเลิกการรัน

การรัน (Run) ก็คือการสั่งให้โปรแกรมทำงานตามโค้ดที่เราเขียนเอาไว้ ซึ่งเราสามารถสั่งรันได้หลายวิธี เช่น

- กดปุ่ม <F5>
- คลิกที่ปุ่มลูกศรบนทูลบาร์ ดังรูป 

เมื่อสั่งรัน โดยปกติฟอร์มจะปรากฏขึ้น หากเราต้องการยกเลิกการรันให้คลิกเลือกจากปุ่ม Stop  บนทูลบาร์ดังรูป

## การใช้ IntelliSense ใน Code Editor

การเขียนโค้ดใน Visual C# นั้นค่อนข้างสะดวกสบาย เพราะมีเครื่องมือเข้ามาช่วยหลายอย่าง ซึ่งตัวที่ล้ำคณ IntelliSense ซึ่งเป็น Popup ที่แสดงรายการสมาชิกที่เราสามารถเติมลงไปในโค้ดตรงนั้นได้ ดังเช่นในรูปต่อไปนี้



เมื่อ IntelliSense ปรากฏขึ้น รายการที่น่าจะตรงกับที่เรากำลังจะใส่ลงไปจะเป็นแถบสี หากเราต้องการใช้ก็กด <Tab> รายการนั้นก็จะถูกนำมาเติมลงใน Code Editor ให้โดยอัตโนมัติ ซึ่งระบบ IntelliSense นั้นนอกจากจะช่วยให้เราเขียนโค้ดได้ง่ายและรวดเร็วขึ้นแล้ว ยังช่วยลดข้อผิดพลาดลงไปได้มากทีเดียว เพราะสิ่งที่เป็นไปได้ที่จะเติมลงตรงนั้นจะถูกแสดงให้เห็นใน IntelliSense แล้ว แต่ไม่ได้หมายความว่าเราเลือกรายการจาก IntelliSense มาเติมแล้วจะไม่เกิดผิดพลาดเลย นอกจากนี้ก็ไม่ได้หมายความว่าต้องเลือกโค้ดจาก IntelliSense เท่านั้น เพราะถึงแม้จะมี IntelliSense ปรากฏขึ้น แต่เราอาจกำหนดค่าลงไปเองโดยไม่เลือกใช้รายการใดๆ ใน IntelliSense นั้นเลยก็ได้

## พื้นฐานการเขียนโปรแกรมกับอีเวนต์

การเขียนโปรแกรมแบบ Visual นั้นส่วนใหญ่จะเป็นการทำงานเพื่อตอบสนองต่อเหตุการณ์บางอย่างที่เกิดขึ้น ซึ่งต่อไปในหนังสือเล่มนี้จะเรียกเหตุการณ์แบบนี้ว่า "อีเวนต์ (Event)" เช่น เมื่อโหลดฟอร์ม หรือคลิกที่ฟอร์มแล้วจะทำอะไรต่อไป เป็นต้น ดังนั้นแนวคิดหลักๆ ของการเขียนโปรแกรมจึงประกอบด้วย



สำหรับอีเวนต์ที่สามารถเกิดขึ้นได้ตามแต่ชนิดคอนโทรลที่เราเลือกใช้ในการออกแบบ แต่ในเบื้องต้นนี้จะแนะนำเพียงอีเวนต์เดียวคือ อีเวนต์ที่เกิดเมื่อโหลดฟอร์ม ทั้งนี้เพื่อให้เราสามารถนำไปใช้ประกอบการศึกษาในช่วงบทแรกๆ ที่ยังไม่ได้เข้าสู่เนื้อหาเกี่ยวกับคอนโทรล โดยมีแนวทางโดยสังเขปต่อไปนี้

## อีเวนต์การโหลดฟอร์ม

อีเวนต์การโหลดฟอร์มจะเกิดเมื่อเปิดฟอร์มขึ้นมาครั้งแรก และเกิดได้เพียงครั้งเดียวจนกว่าจะเปิดฟอร์มขึ้นมาใหม่ ซึ่งเราอาจจะเลือกใช้เวนต์นี้กับสิ่งที่ต้องการกระทำเพียงครั้งเดียวและทำทันทีเมื่อแสดงฟอร์ม ถ้าเราจะเลือกใช้เวนต์นี้ให้ “ดับเบิลคลิก” ที่ตัวฟอร์ม โปรแกรมจะเข้าสู่โหมดการเขียนโค้ดพร้อมสร้างตัวจัดการอีเวนต์ (Event Handler) ดังนี้

```

namespace WindowsFormsApplication1 {
    public partial class Form1 : Form {
        public Form1() {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e) {
        }
    }
}

```

### หมายเหตุ:

- ❖ ตัวจัดการอีเวนต์นี้ เราจะพิมพ์ลงไปเองไม่ได้ มิฉะนั้นจะไม่มีผลต่อการทำงาน ทั้งนี้เนื่องจากการจัดการอีเวนต์ตามแบบ C# นั้นยังรายละเอียดบางประการที่ถูกซ่อนเอาไว้ ซึ่งเราจะได้ศึกษาในบทต่อไป แต่ในเบื้องต้นนี้ให้ใช้การ “ดับเบิลคลิกที่ฟอร์ม” เพื่อสร้างตัวจัดการอีเวนต์ไปก่อน

## การปรับแต่ง Options

เนื่องจากการเขียนโปรแกรมนั้นจะมียังมีองค์ประกอบหลายอย่าง เช่น ตัวแปร, คีย์เวิร์ด, คำอธิบาย, คำสั่งทั่วไป เป็นต้น โดยปกติแล้วโปรแกรมจะกำหนดรูปแบบมาให้แล้วบางส่วน เช่น คีย์เวิร์ดใช้สีน้ำเงิน, คำอธิบายใช้สีเขียว เป็นต้น แต่รูปแบบในบางลักษณะสำหรับโปรแกรมเมอร์แต่ละคนอาจแตกต่างกัน เช่น บางคนถนัดวางบล็อก {} (หรือ open brace/close brace) ในลักษณะแบบนี้

```

private void test()
{
    if (...)
    {
        for (...)
        {
            :
        }
    }
}

```

แต่บางคนอาจนัดวาง open brace ไว้แบบนี้

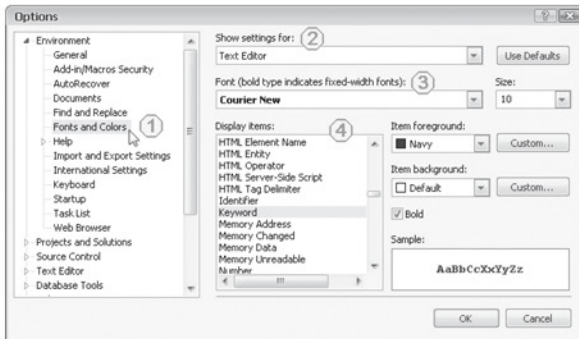
```
private void test() {
    if (...) {
        for (...) {
            .
        }
    }
}
```

หรือกรณีอื่นๆ อีกหลายอย่าง เช่น การให้คีย์เวิร์ดเป็นตัวหนา, การกำหนดสีสำหรับไอเท็มแต่ละอย่าง หรือแม้กระทั่งลักษณะการเว้นช่องว่าง เป็นต้น ซึ่งสิ่งเหล่านี้เราสามารถกำหนดรูปแบบได้หมด ทั้งนี้การกำหนดรูปแบบที่ตรงกับความต้องการหรือความเคยชินของตัวเอง จะทำให้สามารถศึกษาโปรแกรมได้เข้าใจมากยิ่งขึ้น

เนื่องจากการปรับแต่งนั้นมีมากมายหลายอย่าง แต่ในที่นี้จะนำเสนอเฉพาะรูปแบบที่น่าสนใจ เพราะบางอย่างค่าดีฟอลต์ที่โปรแกรมกำหนดไว้ให้ก็ถือว่าเหมาะสมแล้ว หรือผู้อ่านสามารถใช้แนวทางที่แนะนำไว้ไปใช้ค้นหาเองได้

## การปรับแต่งลักษณะตัวอักษรและสี

ให้คลิกที่เมนู Tool > Options... และที่หน้าจอ Options ให้ขยายโหนด Environment > Fonts and Colors ดังรูป



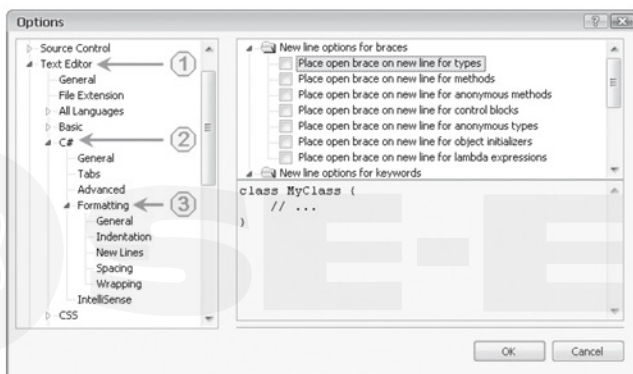
ถ้าเราจะปรับแต่ง ก็ให้ทำดังนี้ (ตามหมายเลขในรูป)

1. ที่กรอบด้านซ้าย เลือก Environment > Fonts and Colors
2. ในกรอบด้านขวา ที่ช่อง Show Settings for: ถ้าจะปรับแต่งสำหรับการเขียนโค้ดให้เลือก Text Editor

3. ที่ช่อง Font ให้เลือกชนิดฟอนต์ที่จะใช้ พร้อมทั้งกำหนดขนาดในช่อง Size
4. ในช่อง Display Items เป็นการเลือกว่าปรับแต่งส่วนใดของโค้ด ซึ่งตัวเลือกที่น่าสนใจ เช่น Plain Text, Comment, Keyword เป็นต้น แล้วเลือกลักษณะ Item foreground (สีฟอนต์), Item background (สีพื้นหลัง) และบางรายการเราสามารถทำตัวหนา (Bold) ได้

## การปรับแต่ง Text Editor

เป็นการปรับแต่งรูปแบบของโค้ดเพื่อให้โค้ดส่วนที่โปรแกรมสร้าง ให้เรามีรูปแบบที่ใกล้เคียงกับที่เราต้องการมากที่สุด โดยมีลำดับการเลือกตามรูปต่อไปนี้



เลือกรูปแบบรายการจากตัวเลือกของ Formatting จากนั้นก็เลือกหรือยกเลิกรายการที่กรอบด้านขวา

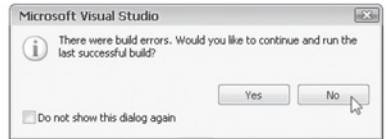
นอกจากการปรับแต่งตามที่กล่าวมาแล้ว ยังสามารถปรับแต่งลักษณะอื่นๆ ได้หลายรูปแบบตามที่เราเห็นในรูป ซึ่งผู้อ่านสามารถทดลองปรับดูเองได้

## การพิจารณาข้อผิดพลาดเบื้องต้น

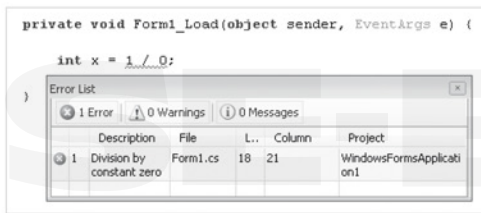
ข้อผิดพลาดคือสิ่งที่เกิดขึ้นได้เสมอในการเขียนโปรแกรม โดยเฉพาะอย่างยิ่งผู้ที่เพิ่งเริ่มต้นศึกษาการเขียนโปรแกรม มักจะพบกับปัญหามากมาย แต่ถ้าผ่านการเขียนไปสักระยะปัญหาจะลดลงเรื่อยๆ และสามารถหาทางแก้ปัญหาได้เร็วขึ้น สำหรับ Code Editor ของ VC# มีเครื่องมือที่ช่วยตรวจสอบข้อผิดพลาดเบื้องต้นได้ตั้งแต่ตอนเขียนโค้ด โดยหากเป็นข้อผิดพลาดทางไวยากรณ์ (Syntax Error) เช่น เขียนคำสั่งผิด จะแสดงข้อผิดพลาดให้เห็นทันที โดยจะปรากฏเส้นหยักใต้คำสั่งที่เขียนผิดนั้น เช่น

```
private void Form1_Load(object sender, EventArgs e) {
    MessageBox.Show("Hello, World!");
}
```

ถ้ามีข้อผิดพลาดเกิดขึ้นเช่นนี้ แล้วเราสั่งรันโปรแกรม จะเกิดคำเตือนในลักษณะดังนี้ ให้เราเลือก No เพื่อกลับไปแก้ไขให้ถูกต้องก่อน หากเราเลือก Yes จะหมายถึงให้ย้อนกลับไปใช้การรันครั้งล่าสุดที่ไม่เกิดข้อผิดพลาด



แต่อย่างไรก็ตามยังมีข้อผิดพลาดในอีกลักษณะที่เราไม่สามารถตรวจสอบได้จาก Code Editor เพราะข้อผิดพลาดแบบนี้ไม่ได้เกิดจากการเขียนคำสั่งผิด แต่เกิดจากการใช้เขียนคำสั่งที่ไม่สามารถประมวลผลได้ ซึ่งข้อผิดพลาดจะถูกแสดงออกมาให้เห็นเมื่อเรารันโปรแกรม เช่นกรณีต่อไปนี้



เมื่อเกิดข้อผิดพลาดเช่นนี้ เราต้องหยุดรันโปรแกรมก่อนตามวิธีที่กล่าวมาแล้ว เพื่อแก้ไขให้ถูกต้องเสียก่อน

สิ่งที่ได้กล่าวถึงทั้งหมดในบทนี้ เป็นเพียงแนวทางการใช้งานทั่วไป เพราะเพื่อจะเป็นพื้นฐานสำหรับการศึกษาในบทต่อไป ทั้งนี้ในบางเรื่องผู้เขียนจะไม่กล่าวถึงแบบละเอียดนัก เพราะจำเป็นต้องใช้ความรู้อย่างอื่นๆ มาพิจารณาร่วมด้วย ซึ่งในบทต่อไป เราจะเข้าสู่เนื้อหาของการพัฒนาโปรแกรมด้วย Visual C# อย่างแท้จริง ก็จะได้กล่าวเพิ่มเติมในบางเรื่องอีก



# พัฒนาแอปพลิเคชันด้วย Visual C# 2008

Visual C# 2008 ได้มีการปรับปรุงประสิทธิภาพการทำงานและเพิ่มเติมเครื่องมือที่ช่วยอำนวยความสะดวกในการเขียนโปรแกรมเข้ามาหลายรูปแบบ ทำให้เรามีแนวทางในการพัฒนาโปรแกรมที่หลากหลายขึ้น ซึ่งในหนังสือเล่มนี้ มุ่งเน้นให้ผู้อ่านได้เข้าใจหลักการที่จำเป็นของการใช้ Visual C# ควบคู่ไปกับแนวทางการนำไปประยุกต์ใช้งาน ด้วยตัวอย่างโปรแกรมต่างๆ รูปแบบ ซึ่งจะช่วยให้ทักษะการเขียนโปรแกรมและมองเห็นแนวทางที่จะนำไปประยุกต์ใช้งานได้ชัดเจนมากยิ่งขึ้น

## เนื้อหาประกอบด้วย

- บทที่ 1 การติดตั้งและการใช้งานทั่วไป
- บทที่ 2 พื้นฐานของ C#
- บทที่ 3 การเขียนโปรแกรมแบบ OOP เบื้องต้น
- บทที่ 4 ข้อมูลและคลาสพื้นฐานที่สำคัญของ .NET
- บทที่ 5 วินโดวส์แอปพลิเคชันและคอนโทรลพื้นฐาน
- บทที่ 6 คอนโทรลและคอมโพเนนต์ที่น่าสนใจอื่นๆ
- บทที่ 7 ลักษณะสำคัญอื่นๆ ของ Win Forms App.
- บทที่ 8 ตัวอย่างโปรแกรม: Web#
- บทที่ 9 การตรวจสอบและจัดการข้อผิดพลาด
- บทที่ 10 การวาดกราฟิกด้วย GDI+
- บทที่ 11 โดอะล็อกและเทคนิคการพิมพ์
- บทที่ 12 ระบบไฟล์และสตรีม
- บทที่ 13 ตัวอย่างโปรแกรม: Text#
- บทที่ 14 การจัดการฐานข้อมูลด้วย ADO.NET
- บทที่ 15 การใช้ SQL และพารามิเตอร์
- บทที่ 16 การสร้างรายงานด้วย Crystal Reports
- บทที่ 17 ตัวอย่างระบบงาน: App#
- บทที่ 18 การสร้างชุดติดตั้งและแจกจ่ายโปรแกรม
- บทที่ 19 แอปพลิเคชันแบบ WPF
- บทที่ 20 การใช้ C# กับเว็บแอปพลิเคชัน ASP.NET

www.se-ed.com

หนังสือเล่มนี้สำหรับ		
ผู้เริ่มต้น	ระดับกลาง	ระดับสูง
คอมพิวเตอร์ / การเขียนโปรแกรม		

ISBN 978-974-212-967-5

9 789742 129675
299 บาท