

คู่มือเขียน

# Windows 8 Apps

ด้วย VB 2012 & VC# 2012 สำหรับผู้เริ่มต้น



Visual Studio

เนื้อหาละเอียด ชัดเจน และเข้าใจอย่างแท้จริง

ตัวอย่าง Workshop พร้อมใช้งาน

ศึกษาและใช้งานภาษาสคริปต์ XAML ร่วมกับ VB 2012 & VC# 2012

คู่มือเขียน Windows 8 Apps  
ด้วย VB 2012 และ VC# 2012 สำหรับผู้เริ่มต้น

คู่มือเขียน Windows 8 Apps

ด้วย VB 2012 และ VC# 2012 สำหรับผู้เริ่มต้น

ผู้แต่ง : นายศุภชัย สมพานิช

เนื้อหา : 18 บท (618 หน้า)

พอนต์ : [www.f0nt.com/](http://www.f0nt.com/)

วันที่พิมพ์ : กันยายน 2555

ครั้งที่พิมพ์ : 1st Edition

ลิขสิทธิ์ : สงวนลิขสิทธิ์ตาม พ.ร.บ. ลิขสิทธิ์ พ.ศ. 2537

ห้ามลอกเลียนแบบ, ดัดแปลง, ทำซ้ำ, เผยแพร่ ไม่ว่าส่วนหนึ่งส่วนใดของหนังสือเล่มนี้ หรือด้วยวิธีการใดๆ ก็ตาม นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้เขียนเท่านั้น

คุณผู้อ่านสามารถสอบถามปัญหา-ติชมเนื้อหาหนังสือเล่มนี้ได้ที่

<https://www.facebook.com/thaivb.net>

Visual Studio เป็นเครื่องหมายทางการค้าของบริษัท Microsoft Coporation จำกัด  
โปรแกรมต่างๆ ตามที่อ้างในหนังสือเล่มนี้ เป็นเครื่องหมายการค้าของบริษัทนั้นๆ

## คำนำ

การพัฒนา Windows 8 Apps ใน Visual Studio 2012 ด้วย VB 2012 & VC# 2012 ถือว่าเป็นเรื่องใหม่มากสำหรับโปรแกรมเมอร์สาย .NET มาพร้อมกับการเปิดตัวของระบบปฏิบัติการ Windows 8 และอุปกรณ์แท็บเล็ตของไมโครซอฟท์ที่เรียกว่า Surface

เนื้อหาที่นำเสนอในหนังสือเล่มนี้ เหมาะสำหรับคุณผู้อ่านที่เป็นมือใหม่ในโลกของ .NET และโปรแกรมเมอร์สาย .NET เดิมที่ต้องการพัฒนา Windows 8 Apps ด้วย VB 2012 & VC# 2012 โดยที่ผู้เขียนนำเสนอเนื้อหาครบถ้วนทั้ง 2 ภาษา กล่าวคือ

เนื้อหาเริ่มตั้งแต่ปูพื้นฐานระดับเบื้องต้น, ทำความรู้จักกับโครงสร้างของ Windows 8 Apps ไปจนกระทั่งถึงการสร้าง Windows 8 Apps ได้จริง โดยการยกตัวอย่างประกอบทุกหัวข้อ เพื่อให้เห็นผลการทำงานชัดเจน ง่ายต่อการทำความเข้าใจ

สุดท้ายนี้ ผู้เขียนขอขอบคุณส่วนช่วยเหลือใน MSDN ของไมโครซอฟท์ที่ช่วยแก้ไขปัญหาระหว่างการแต่งหนังสือเล่มนี้

หากคุณผู้อ่านท่านใด ต้องการสอบถามปัญหาเนื้อหาของหนังสือเล่มนี้ สามารถโพสถามปัญหาได้ที่แฟนเพจของผู้เขียนที่ <https://www.facebook.com/thaivb.net> โดยที่ผู้เขียนขอยกเลิกการถาม-ตอบปัญหาผ่านระบบ e-mail ของผู้เขียนทั้งหมด

ศุภชัย สมพานิช

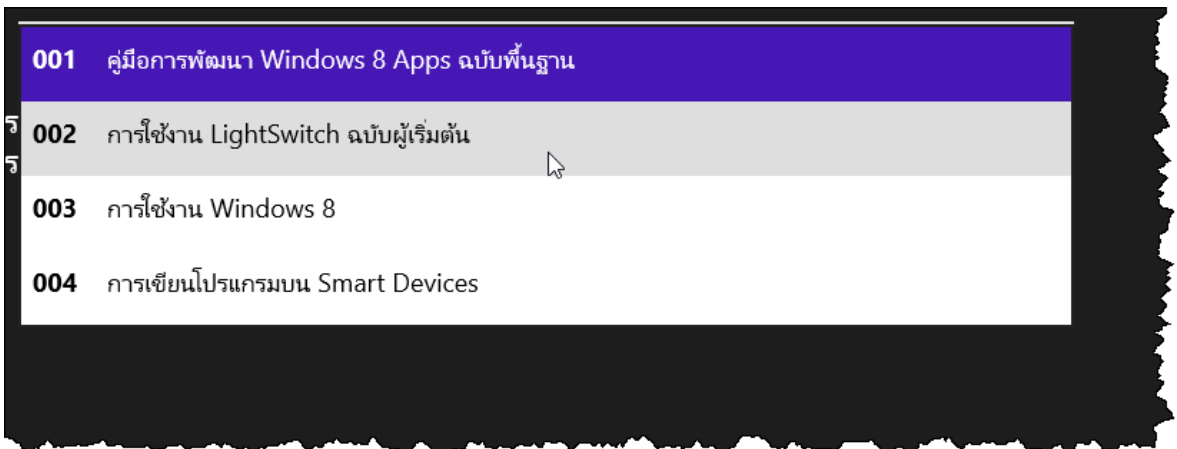
กันยายน 2555

## ตัวอย่างที่น่าสนใจในหนังสือเล่มนี้

การสร้างปุ่มควบคุมแสดงคลิปวิดีโอระดับ FullHD ในส่วนแสดงผลของ Windows 8 Apps



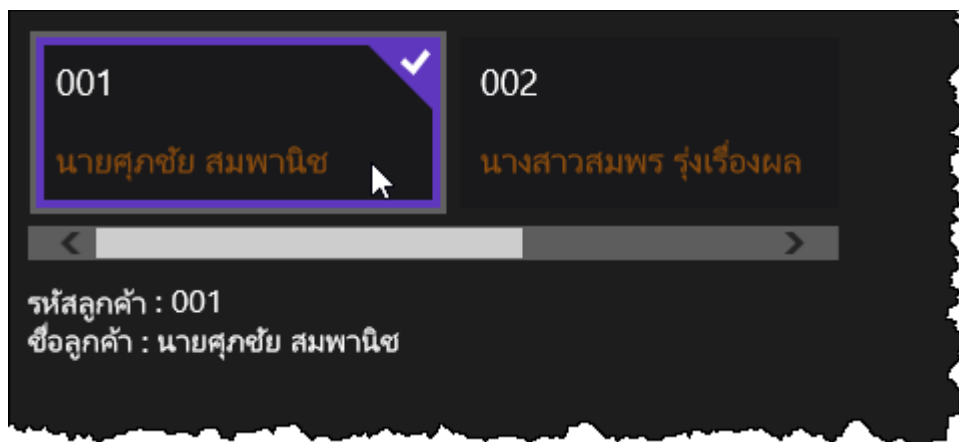
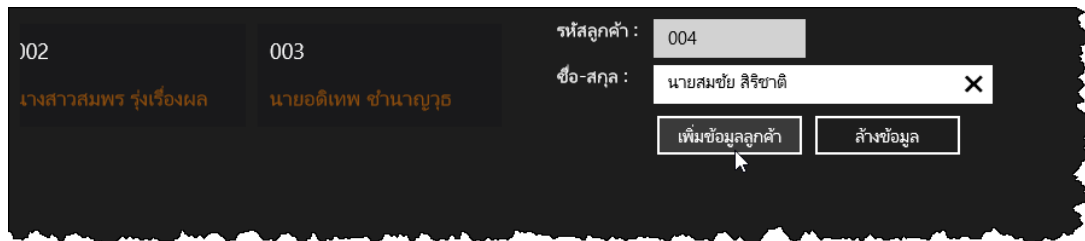
การแก้ไขส่วนแสดงผลในคอนโทรล ComboBox แบบที่ 1



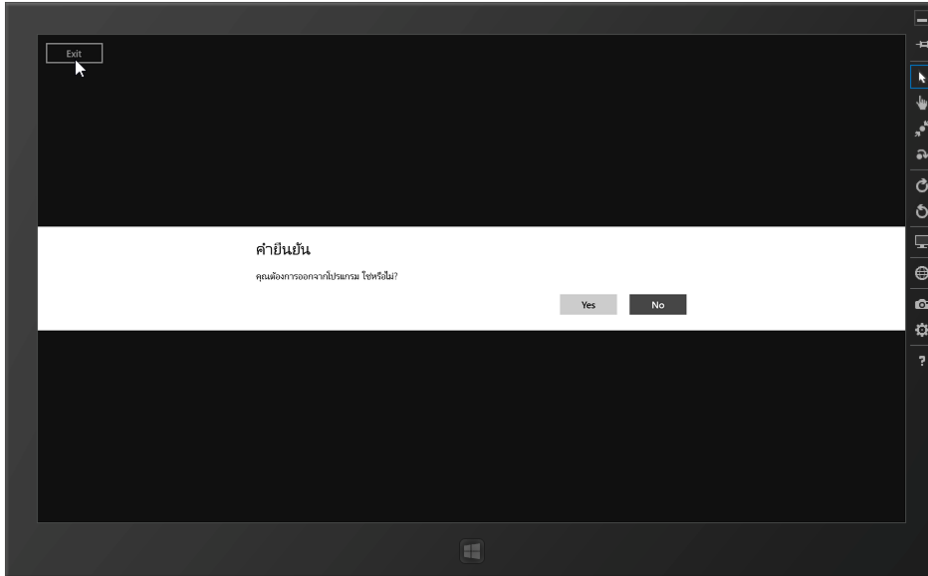
## การแก้ไขส่วนแสดงผลในคอนโทรล ComboBox แบบที่ 2



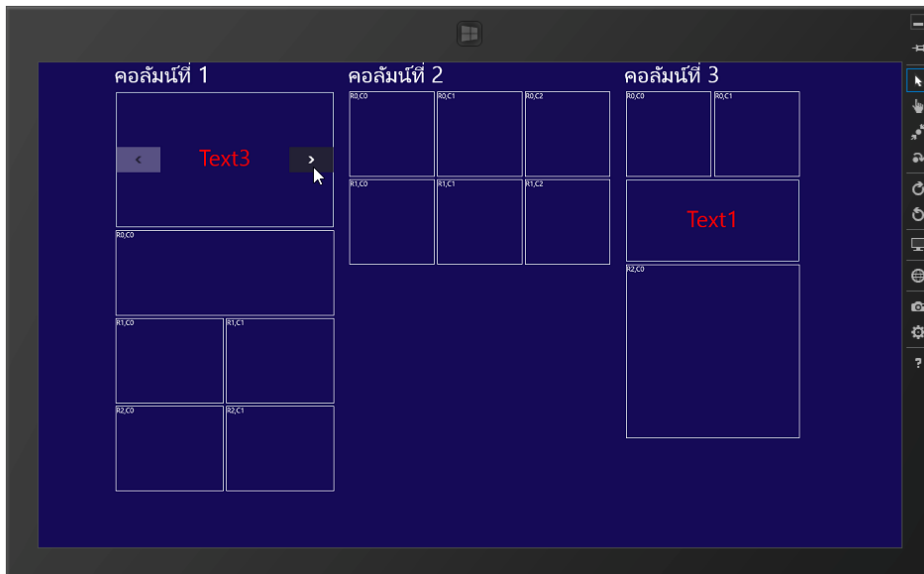
## การทำงานกับคอนโทรล GridView



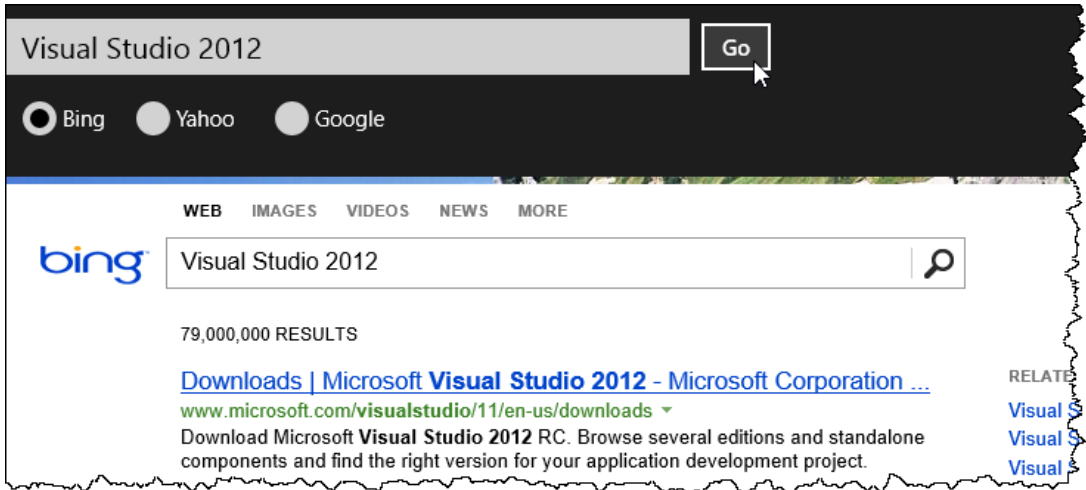
## การสร้างไดอะล็อกแจ้งผู้ใช้งาน



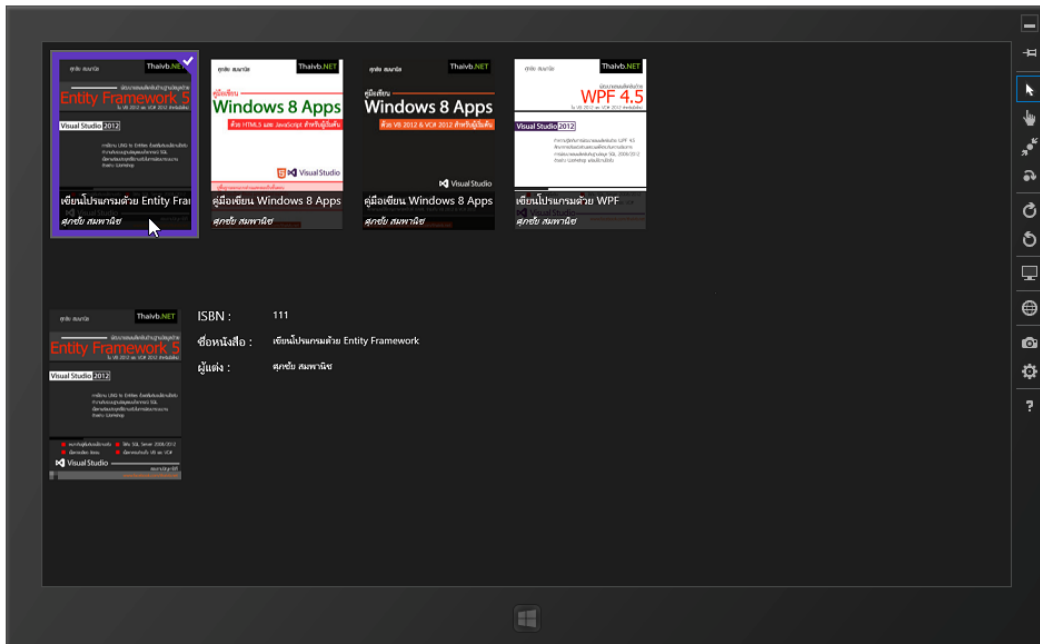
## ส่วนแสดงผลแบบ Windows Store



## การค้นหาข้อมูลจาก Bing, Yahoo และ Google ด้วย Windows 8 Apps



## การแสดงความคมและรูปภาพในคอนโทรล GridView





## สารบัญ

<b>บทที่ 1 ก้าวแรกเข้าสู่โลกของ Windows 8 Apps</b>	<b>1</b>
บทนำ	1
ทำความรู้จักกับ Windows 8 ในขั้นต้น	2
รูปแบบของ App ในยุคปัจจุบัน	4
แนวทางของโปรแกรมเมอร์ในยุค .NET Framework 4.5	5
การพัฒนา Windows 8 Apps ด้วย Visual Studio 2012 Express for Windows 8	17
การขอสิทธิชั่วคราวในการพัฒนา Windows 8 Apps	23
การขอหรือต่ออายุ Developer license	27
โครงสร้างของ Windows 8 Apps	28
รูปแบบการนำเสนอตัวอย่างในหนังสือเล่มนี้	31
รูปแบบการลงโค้ด (VB 2012 & VC# 2012) และสคริปต์ XAML ในหนังสือเล่มนี้	32
การจัดระเบียบโค้ดหรือสคริปต์ใน VS 2012	33
สรุปท้ายบท	34
<b>บทที่ 2 พื้นฐานการพัฒนา Windows 8 Apps</b>	<b>35</b>
บทนำ	35
Windows 8 Apps แรกของคุณ	36
การสร้างเหตุการณ์ในสคริปต์ XAML	46
การทดสอบรัน Windows 8 Apps	53
ทำความรู้จักกับ Simulator ในขั้นต้น	54
การกำหนดเพจเริ่มต้น	61
วิธีการปิด Windows 8 Apps ที่ถูกต้อง	62
การค้นหาคุณสมบัติต่างๆ	65
การอ้างอิงคอนโทรลหรืออีลีเมนต์	65
การเลือกโฟกัสคอนโทรลที่สนใจอยู่	69
การยกเลิกแก้ไขคุณสมบัติ	71
การ Copy คอนโทรลด้วยปุ่ม Alt บนคีย์บอร์ด	72

การจัดตำแหน่งคอนโทรลด้วยคุณสมบัติด้าน Alignment	73
การจัดระยะห่างด้วยคุณสมบัติ Margin	76
สรุปท้ายบท	78
<b>บทที่ 3 การจัดส่วนแสดงผลด้วยคอนโทรลประเภท Layout</b>	<b>79</b>
บทนำ	79
พื้นฐานการใช้งานคอนโทรล Grid	80
การรวมคอลัมน์หรือรวมแถวในคอนโทรล Grid	85
การปรับแต่ง Layout ของคอนโทรล Grid	87
การเปลี่ยนสีพื้นหลังให้กับตัวบรรจุ	89
การจัดเรียงคอนโทรลตามแนวนอนหรือแนวตั้งด้วยคอนโทรล StackPanel	96
การเปลี่ยนตัวบรรจุด้วยคำสั่งใน VS 2012	99
การจัดกลุ่มคอนโทรลด้วยคำสั่งใน VS 2012	100
การควบคุมตัวบรรจุ	102
การสร้างส่วนแสดงผลแบบพื้นที่อิสระด้วยคอนโทรล Canvas	106
การซ้อนกันของวัตถุในส่วนแสดงผลแบบ Canvas	108
การแสดงผลข้อมูลแบบมีแถบเลื่อนด้วยคอนโทรล ScrollViewer	110
สรุปท้ายบท	112
<b>บทที่ 4 การใช้งานคอนโทรลพื้นฐาน</b>	<b>113</b>
บทนำ	113
การสร้างช่องป้อนข้อความด้วยคอนโทรล TextBox	114
การแสดงผลข้อความหลายบรรทัดในคอนโทรล TextBox	116
การสร้างปุ่มที่มีรูปและข้อความ	118
พื้นฐานการแสดงผลข้อความแบบหุ้หุ้หุ้ด้วยคอนโทรล RichTextBlock	119
การสร้างเนื้อหาแบบหุ้หุ้หุ้ในคอนโทรล RichTextBlock	121
การสร้างตัวเลือกด้วยคอนโทรล RadioButton	123
การตรวจสอบ Content ของคอนโทรล RadioButton กรณีที่มีการปรับแต่ง	124
การสร้างตัวเลือกด้วยคอนโทรล CheckBox	129

การสร้างปุ่มกดค้างด้วยคอนโทรล RepeatButton	133
การสร้างเบราว์เซอร์ขนาดเล็กด้วยคอนโทรล WebView	137
พื้นฐานการเล่นคลิป์วิดีโอด้วยคอนโทรล MediaElement	143
การสร้างเครื่องเล่นคลิป์วิดีโอด้วยคอนโทรล MediaElement	145
สรุปท้ายบท	153
<b>บทที่ 5 การใช้งานคอนโทรลประเภท list</b>	<b>154</b>
บทนำ	154
การใช้งานคอนโทรล ComboBox	155
พื้นฐานการใช้งานคอนโทรล ListBox	160
การแสดงผลรูปภาพในรายการของคอนโทรล ListBox	168
การเพิ่มและลบรายการในคอนโทรล ListBox	175
สรุปท้ายบท	179
<b>บทที่ 6 การปรับแต่งคอนโทรลด้วย Style</b>	<b>180</b>
บทนำ	180
การใช้งาน Style	181
การกำหนดขอบเขตการใช้งานของ Resources	184
การกำหนดขอบเขตการใช้งาน Resources ระดับ Local	185
การกำหนดขอบเขตการใช้งาน Resources ระดับ Container	186
การใช้งาน Resources ที่มีขอบเขตระดับ Application	186
การเรียกใช้แบบภายใน	187
การเรียกใช้แบบภายนอก	188
การกำหนดธีมใน Windows 8 Apps ตามที่เราต้องการ	190
การควบคุมธีมโดยการเขียนโค้ด	192
สรุปท้ายบท	198
<b>บทที่ 7 การผูกติดข้อมูล (Data Binding) ใน Windows 8 Apps</b>	<b>199</b>
บทนำ	199
การผูกติดข้อมูลระหว่างคอนโทรลของ Windows 8 Apps	200

รูปแบบการผูกติดข้อมูล	201
การสร้างธาตุผสมสี	205
การผูกติดข้อมูลจากคลาสแบบปกติ	210
การผูกติดข้อมูลโดยการเขียนโค้ด	217
สรุปท้ายบท	222
<b>บทที่ 8 การแก้ไขส่วนแสดงผลในคอนโทรลผ่านระบบ Template</b>	<b>223</b>
บทนำ	223
การผูกติดข้อมูลกับคลาสโดยการแก้ไข Template	224
การแก้ไข Template ในขณะออกแบบ	236
การปรับแต่งรายการในคอนโทรล ComboBox	248
พื้นฐานการแสดงผลข้อมูลในคอนโทรล ListView แบบแก้ไข Template	255
สรุปท้ายบท	266
<b>บทที่ 9 ระบบไฟล์ใน Windows 8 Apps</b>	<b>267</b>
บทนำ	267
การเรียกระบบจัดเก็บข้อมูลใน Windows 8	268
แนวความคิดของการเขียนโปรแกรมแบบซัดจ์หะวะได้ (Asynchronous Programming)	268
พื้นฐานการสร้าง, เขียนและอ่านไฟล์ใน Windows 8 Apps	269
ทำความเข้าใจกับไฟล์เดอ์ระบบในชั้นต้นของ Windows 8	283
การเปิดไฟล์ pdf หรือไฟล์ประเภทอื่นๆ ด้วย Windows 8 Apps	286
การอ่านพาสของไฟล์เดอ์ Assets และ Common	292
การค้นหาไฟล์	295
การเพิ่มประสิทธิภาพค้นหาไฟล์ด้วย Advanced Query Syntax	304
ทำงานกับระบบจัดเก็บข้อมูลด้วยคอนโทรล FileOpenPicker	309
สรุปท้ายบท	316
<b>บทที่ 10 ทำงานกับรูปภาพ</b>	<b>317</b>
บทนำ	317
การแสดงรูปภาพด้วยคอนโทรล Image	318

การลงสีข้อความด้วยรูปภาพ	319
การกำหนดความเข้ม-จางด้วยคุณสมบัติ Opacity	320
การเปลี่ยนรูปภาพโดยการเขียนโค้ด	322
สรุปท้ายบท	325
<b>บทที่ 11 การออกแบบส่วนแสดงผลใน windows 8 Apps</b>	<b>326</b>
บทนำ	326
การตรวจสอบความละเอียดของส่วนแสดงผล	327
การสร้างส่วนแสดงผลแบบช่องเปลี่ยนเนื้อหาได้ด้วยคอนโทรล FlipView	330
การสร้างอัลบั้มรูปด้วยคอนโทรล FlipView	336
การสร้างส่วนแสดงผลหลายแบบ (Multiple View) ด้วยคอนโทรล FlipView	339
การสร้างส่วนแสดงผลแบบเปลี่ยนหลายหน้าได้	355
การรับ-ส่งข้อมูลระหว่างเพจ	359
การรักษาสถานะข้อมูลด้วยระบบแคช (Cache)	367
การสร้างส่วนแสดงผลแบบตารางสี่เหลี่ยมด้วยคอนโทรล GridView	371
การเพิ่มรายการในคอนโทรล GridView โดยการเขียนโค้ด	381
สรุปท้ายบท	390
<b>บทที่ 12 ระบบเมนูและไดอะล็อก</b>	<b>391</b>
บทนำ	391
การแสดงผลข้อมูลแบบ Popup ด้วยคอนโทรล Popup	392
การสร้างเมนูจากการคลิกขวา (ContextMenu)	395
พื้นฐานการใช้งาน AppBar	398
การปรับแต่งแถบ AppBar โดยใช้ Style สำเร็จรูป	403
การสร้างเมนูให้กับแถบ AppBar	408
การสร้างกรอบข้อความด้วยคลาส AlertDialog	413
การใช้ AlertDialog แบบมีการตรวจสอบปุ่มที่ถูกเลือก	415
การสร้าง AlertDialog ก่อนออกจาก App	419
สรุปท้ายบท	421

<b>บทที่ 13 ทำความรู้จักกับ Blend for Visual Studio 2012</b>	<b>422</b>
บทนำ	422
การใช้งาน VS 2012 ร่วมกับ Blend	424
การปรับแต่งสภาพแวดล้อมของ Blend	429
ทำความรู้จักกับ Artboard	430
การซูม Artboard	434
โหมดการทำงานใน Blend	434
วิธีการรันโปรเจกต์ใน Blend	436
การกำหนดคุณสมบัติต่างๆ ของคอนโทรล	436
การปรับค่าให้กับคุณสมบัติที่เป็นตัวเลข	437
คุณสมบัติที่เกี่ยวข้องกับระบบสี	437
วิธีการเลือกสี	440
การใช้งาน Style ใน Blend	443
การเรียกใช้ Style เดิมที่มีอยู่	449
การเปลี่ยนขอบเขตเรียกใช้งานของ Style	451
สรุปท้ายบท	454
<b>บทที่ 14 การใช้งานแหล่งข้อมูล XML ใน Windows 8 Apps</b>	<b>455</b>
บทนำ	455
โครงสร้างของภาษา XML	456
ความหมายของคำว่าแท็ก (Tag) กับอีลีเมนต์ (Element)	456
กฎ กติกาเบื้องต้นสำหรับการสร้างแหล่งข้อมูล XML	457
พื้นฐานการใช้งานแหล่งข้อมูล XML ด้วยคลาส XElement	460
ข้อความที่มีรูปแบบ XML	463
การแสดงผลข้อมูลจากไฟล์ XML ด้วยคลาส XDocument	465
การสร้างแหล่งข้อมูล XML โดยอาศัยคลาส XDocument	474
การทำงานกับโครงสร้างของแหล่งข้อมูล XML	479
การจัดการแหล่งข้อมูล XML ที่เก็บอยู่ในคลาส XElement	483

สรุปท้ายบท	487
<b>บทที่ 15 การใช้ประโยชน์จากแหล่งข้อมูล XML ด้วย LINQ to XML</b>	<b>488</b>
บทนำ	488
พื้นฐานการใช้งาน LINQ to XML	489
การควิรีข้อมูลจากไฟล์ XML ภายนอก (*.xml)	489
การควิรีข้อมูลที่อยู่ในรูปแบบ XML	496
การแสดงผลข้อมูล XML ร่วมกัน	498
ค่า Null ใน XML	503
การควิรีข้อมูลจากแหล่งข้อมูล XML แบบระบุลำดับอีลีเมนต์	509
การควิรีข้อมูลจาก XML แบบแอ็ททริบิวต์	519
การสร้างแหล่งข้อมูล XML จากอาร์เรย์	525
การสร้างแหล่งข้อมูล XML จากคลาส	526
การแสดงผลแหล่งข้อมูล XML ด้วย HTML	534
สรุปท้ายบท	540
<b>บทที่ 16 การแสดงผลข้อมูลใน XML ด้วย XSLT</b>	<b>541</b>
บทนำ	541
การแสดงผลแหล่งข้อมูล XML ด้วยภาษา XSLT แบบไม่มีเงื่อนไข	542
สรุปท้ายบท	555
<b>Workshop 1 : การสร้างหน้า Login</b>	<b>556</b>
การสร้างหน้า Login	556
<b>Workshop 2 : การสร้าง Windows 8 Apps แบบเปลี่ยนสีพื้นหลังได้</b>	<b>560</b>
การสร้าง Windows 8 Apps แบบเปลี่ยนสีพื้นหลังได้	560
<b>Workshop 3 : การสร้างส่วนแสดงผลแบบ Windows Store</b>	<b>565</b>
การสร้างส่วนแสดงผลแบบ Windows Store	565
<b>Workshop 4 : การค้นหาข้อมูลผ่าน Bing, Yahoo และ Google</b>	<b>574</b>
การค้นหาข้อมูลผ่าน Bing, Yahoo และ Google	574
<b>Workshop 5 : การแสดงผลข้อมูลและรูปภาพในคอนโทรล GridView</b>	<b>582</b>

การแสดงผลข้อมูลและรูปภาพในคอนโทรล GridView	582
<b>บทที่ 17 การปรับแต่งโปรเจกต์ Windows 8 Apps</b>	<b>600</b>
บทนำ	600
การปรับแต่ง Windows 8 Apps ในไฟล์ Package.appxmanifest	601
การกำหนดหน้าจอ Splash screen	603
การกำหนดไฟล์รูปภาพโลโก้	604
การกำหนดข้อความกำกับโลโก้	609
การกำหนดรูปแบบส่วนแสดงผล	611
การสร้าง App Package	612
สรุปท้ายเล่ม	618





# Thaivb.NET

.NET • iOS • Android

## บทที่ 1 ก้าวแรกเข้าสู่โลกของ Windows 8 Apps

### บทนำ

ในปัจจุบัน ไมโครซอฟท์เปิดตัวระบบปฏิบัติการตัวใหม่ มีชื่อเรียกว่า **“Windows 8”** ความพิเศษของระบบปฏิบัติการรุ่นใหม่ก็คือ เป็นระบบปฏิบัติการ Windows รุ่นแรกที่ถูกสร้างขึ้นมาให้ทำงานได้ทั้งบนคอมพิวเตอร์ (PC), Notebook และอุปกรณ์แท็บเล็ต (Tablet) โดยที่ไมโครซอฟท์เรียกอุปกรณ์แท็บเล็ตของตัวเองว่า **เซอร์เฟส (Surface)**

หนังสือ “คู่มือเขียน Windows 8 Apps ด้วย VB 2012 และ VC# 2012 สำหรับผู้เริ่มต้น” เล่มนี้เป็นการนำเสนอพื้นฐานการพัฒนา Windows 8 Apps โดยการใช้ภาษา VB 2012 & VC# 2012 ซึ่งเป็นภาษาที่ได้รับนิยมอยู่ในปัจจุบัน

เนื้อหาที่นำเสนอในหนังสือเล่มนี้ เริ่มต้นตั้งแต่ระดับพื้นฐาน ไล่ไปจนกระทั่งถึงการกำหนดรายละเอียดต่างๆ เพื่อให้ Windows 8 Apps ของคุณ ตรงกับมาตรฐานที่ไมโครซอฟท์กำหนดไว้ ซึ่งถือเป็นรากฐานที่มีความสำคัญเป็นอย่างมาก ก่อนที่จะพัฒนาเข้าสู่ระดับการสร้าง Windows 8 Apps เพื่อขายจริงใน Windows Store ในลำดับต่อไป

## ทำความรู้จักกับ Windows 8 ในขั้นต้น

เราสามารถแยก Windows 8 ออกเป็น 2 แพลตฟอร์ม คือ

1.บนคอมพิวเตอร์ เรียกว่า **Windows 8 Pro** รันบนสถาปัตยกรรมซีพียูแบบ x86 (32 บิต) และ x64 (64 บิต) ที่คุ้นเคยกันเป็นอย่างดี คือ ซีพียูของอินเทล (Intel) และเอเอ็มดี (AMD) **โดยที่ Windows 8 Pro ที่รันบนซีพียูตระกูล x86/x64 นี้สามารถใช้งาน Windows 8 ได้ทั้งแบบ Desktop Mode กับ Modern UI Style Mode**



รูปที่ 1-1 แสดง Windows 8 แบบ Modern UI Style Mode

จากรูปที่ 1-1 แอปพลิเคชันที่ติดตั้งบน Windows 8 แบบนี้ ผู้เขียนขอเรียกว่า **Windows 8 Apps**



รูปที่ 1-2 แสดง Windows 8 แบบ Desktop Mode

จากรูปที่ 1-2 เป็นสภาพแวดล้อมของ Windows ที่เรารู้จักกันเป็นอย่างดีมาตั้งแต่ Windows 7 และเวอร์ชันก่อนหน้านี้

2.บนอุปกรณ์แท็บเล็ต สำหรับ Windows 8 ที่รันอยู่บนแท็บเล็ต ยังสามารถแยกออกเป็น 2 แบบ คือ

- รันบนสถาปัตยกรรมซีพียูแบบ x86/x64 ส่งผลให้คุณสามารถใช้ Windows 8 ได้ทั้งแบบ Desktop Mode กับ Modern UI Style Mode ยังคงเรียกว่า Windows 8 Pro เช่นเดิม
- รันบนสถาปัตยกรรมซีพียูแบบ ARM เรียกว่า *Windows RT* สำหรับแท็บเล็ตที่ใช้ซีพียูแบบ ARM สามารถรัน Windows RT ได้เพียงอย่างเดียวเท่านั้น โดยที่ Windows RT ก็คือ Windows 8 ที่มี Modern UI Style Mode เป็นแกนหลัก ถูกสร้างขึ้นมาให้ทำงานกับอุปกรณ์แท็บเล็ตเท่านั้น ส่งผลให้โปรแกรมสาย .NET จึงมีพื้นที่ใหม่ที่ไม่เคยมีมาก่อน นั่นคือ การเขียนโปรแกรมที่เรียกว่า

Windows 8 Apps ที่สามารถรันได้ทั้งบน PC, Notebook และอุปกรณ์แท็บเล็ตที่ติดตั้ง Windows 8 หรือ Windows RT นั่นเอง

## รูปแบบของ App ในยุคปัจจุบัน

เพื่อให้คุณมองเห็นภาพในยุคปัจจุบันชัดเจนขึ้นว่า เรากำลังทำอะไรกันอยู่ ผู้เขียนขอแยกรูปแบบ App ออกเป็น 5 แบบ ดังนี้

**1.Command App** หรือแอปพลิเคชันที่ผู้ใช้ต้องมีการป้อนคำสั่งด้วยการป้อนคำสั่ง **command** เราคุ้นเคยกันดีในชื่อ Console Application ในยุคของ DOS รวมไปถึงแอปพลิเคชันบางประเภท ที่รันอยู่บนระบบปฏิบัติการ Server ที่ยังคงอาศัยการป้อนคำสั่งแบบข้อความโดยตรง

**2.Click App** เราคุ้นเคยกับแอปพลิเคชันประเภทนี้มากที่สุด ใช้เมาส์เคอร์เซอร์คลิกไอคอนต่างๆ สั่งงานให้แอปพลิเคชันทำงาน

**3.Touch App** เราเข้ามาสู่ยุคที่โทรศัพท์มือถือ และแท็บเล็ตก็มีแอปพลิเคชันต่างๆ อยู่ด้วยเช่นกัน เรียกสั้นๆ ว่า **App** คุณสั่งการให้ App เหล่านี้ทำงาน โดยการสัมผัสหน้าจอ เพราะความที่นิ้วมือของคุณ มีขนาดใหญ่กว่าเคอร์เซอร์ของเมาส์ จึงทำให้ผู้ที่ต้องการเข้ามาพัฒนา App ในส่วนนี้ ต้องแก้ไขส่วนแสดงผลใหม่ให้รองรับกับการสัมผัสด้วยนิ้วมือของผู้ใช้แทนเคอร์เซอร์ของระบบคลิก

**4.Motion App** หรือ App ที่จับการเคลื่อนไหวของผู้ใช้ แล้วแปลงการเคลื่อนไหวดังกล่าว ไปเป็นคำสั่งใน App เพื่อสั่งการต่างๆ ตามที่ได้กำหนดไว้ สำหรับคุณผู้อ่านที่ชอบเล่นเกม Console ของ Xbox คุ้นเคยกันเป็นอย่างดี กับ App กลุ่มนี้

**5.Voice App** หรือ App ที่สั่งการด้วยคำพูดของมนุษย์ เรากำลังเริ่มต้นเข้าสู่ยุคของการสั่ง App ให้ทำงานด้วยคำพูดของเรา โดยที่เราไม่จำเป็นต้องเข้าไปแตะต้อง หรือสัมผัสใดๆ เลยกับอุปกรณ์ Hardware ที่เราตั้งใจอยู่ คุ้นเคยกันเป็นอย่างดีกับคุณผู้อ่านที่ชอบดูหนังประเภทไซไฟต่างๆ ที่ตัวละครมักจะพูดสั่งให้ซอฟต์แวร์ต่างๆ ทำงาน

## แนวทางของโปรแกรมเมอร์ในยุค .NET Framework 4.5

การอัปเดตของไมโครซอฟท์ในยุคของ .NET 4.5 และการมาของ Windows 8 ถือเป็น การปรับปรุงครั้งใหญ่อีกครั้งหนึ่ง ให้คุณดูรูปที่ 1-3 ประกอบคำอธิบาย

Desktop Application      Windows Forms Application  
WPF Application  
LightSwitch

Web Application      ASP.NET  
ASP.NET Web Matrix  
Silverlight

Windows 8 Apps      XAML (VB, VC#)  
JavaScript, HTML 5

รูปที่ 1-3 แสดงประเภทแอปพลิเคชันของ .NET 4.5 ในขั้นต้น

การพัฒนาแอปพลิเคชันก่อนหน้านี้ (เวอร์ชัน 4.0 และเวอร์ชันเก่ากว่า) โปรแกรมเมอร์สาย .NET มีพื้นที่ให้คุณทำงาน 2 ส่วนใหญ่ๆ คือ

1. Desktop Application
2. Web Application

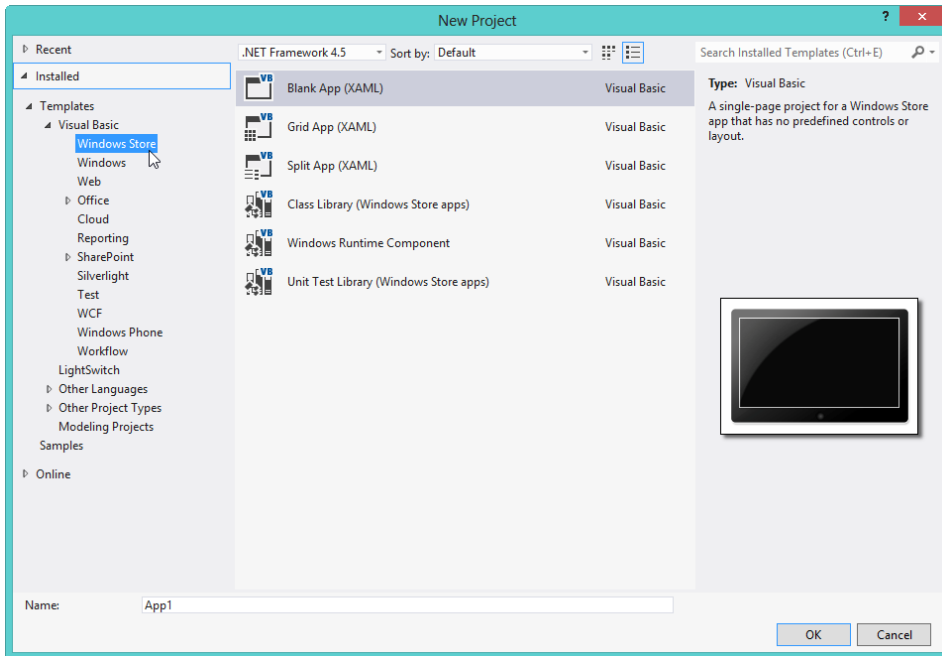
การพัฒนาแอปพลิเคชันทั้ง 2 สาย เป็นพื้นที่เดิมที่เราคุ้นเคยกันเป็นอย่างดี ขึ้นอยู่กับความสนใจของคุณว่า จะเลือกเดินไปในเส้นทางใด ก็แล้วแต่ความชอบ ความถนัดของแต่ละบุคคล

แต่ในยุคปัจจุบัน .NET Framework ถูกพัฒนาไปถึงเวอร์ชัน 4.5 และเข้าสู่ยุคของ Windows 8 โปรแกรมเมอร์สาย .NET มีพื้นที่ใหม่ขึ้นมาอีก 1 แห่ง ทำงานบนอุปกรณ์ Tablet

เป็นที่ทราบกันดีว่า ในยุคปัจจุบัน ผู้คนโดยทั่วไปต่างคุ้นเคยกับการใช้ **แท็บเล็ต (Tablet)** และ **สมาร์ทโฟน (Smart Phone)** หลายคนเปลี่ยนพฤติกรรมจากที่เคยทำบนเครื่องคอมพิวเตอร์ มาทำบนอุปกรณ์เหล่านี้แทน เช่น การเล่นอินเทอร์เน็ต, การเล่นเกมส์, การใช้แอปพลิเคชันต่างๆ ฯลฯ เป็นต้น

ในโลกของ .NET เราเรียกแอปพลิเคชันต่างๆ ที่รันบนอุปกรณ์เหล่านี้ว่า **Windows 8 Apps** และเรียกอุปกรณ์เหล่านี้โดยรวมว่า **Smart Devices**

กล่าวได้อีกนัยหนึ่งว่า ถ้าคุณต้องการพัฒนาแอปพลิเคชันบน Smart Devices เหล่านี้ด้วย .NET คุณต้องพัฒนาแอปพลิเคชันแบบ Windows 8 Apps นั่นเอง



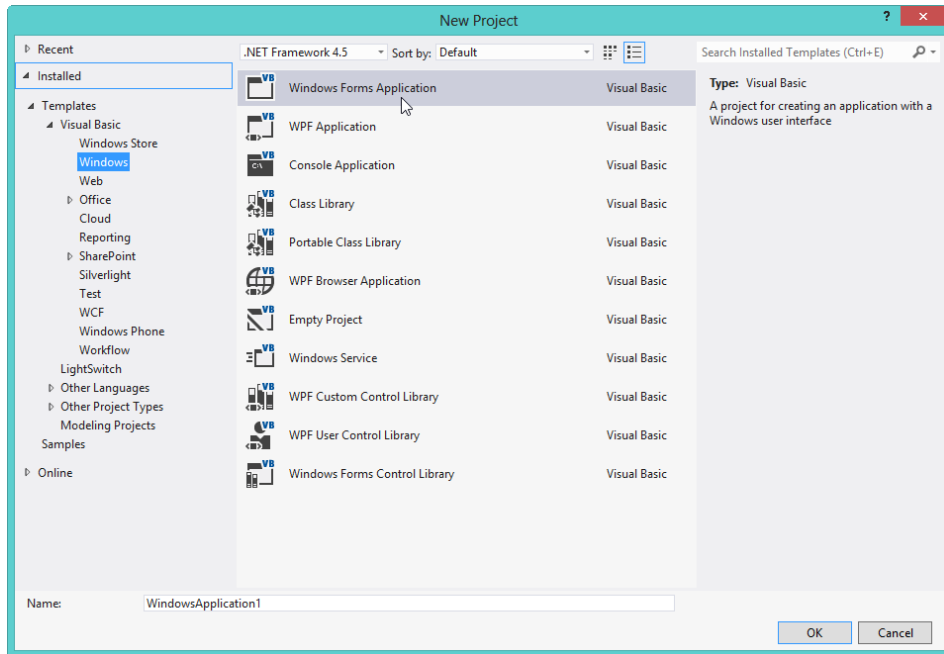
รูปที่ 1-4 แสดงโปรเจกต์แบบ Windows 8 Apps ใน Visual Studio 2012 บน Windows 8

ไมโครซอฟท์ออกแบบให้ Windows 8 ทำงานได้ทั้ง PC (คอมพิวเตอร์) และ Tablet ซึ่งเป็นพื้นที่แห่งใหม่ของโปรแกรมเมอร์สาย .NET ทำงานร่วมกับ .NET Framework 4.5

ความสำคัญอีกประการหนึ่ง ก็คือ คุณต้องทราบให้ได้ก่อนว่า คุณกำลังอยู่ส่วนไหนในโลกของ .NET 4.5 บน Windows 8 กล่าวคือ

## 1.Desktop Application คุณมีทางเลือก 3 ทาง คือ

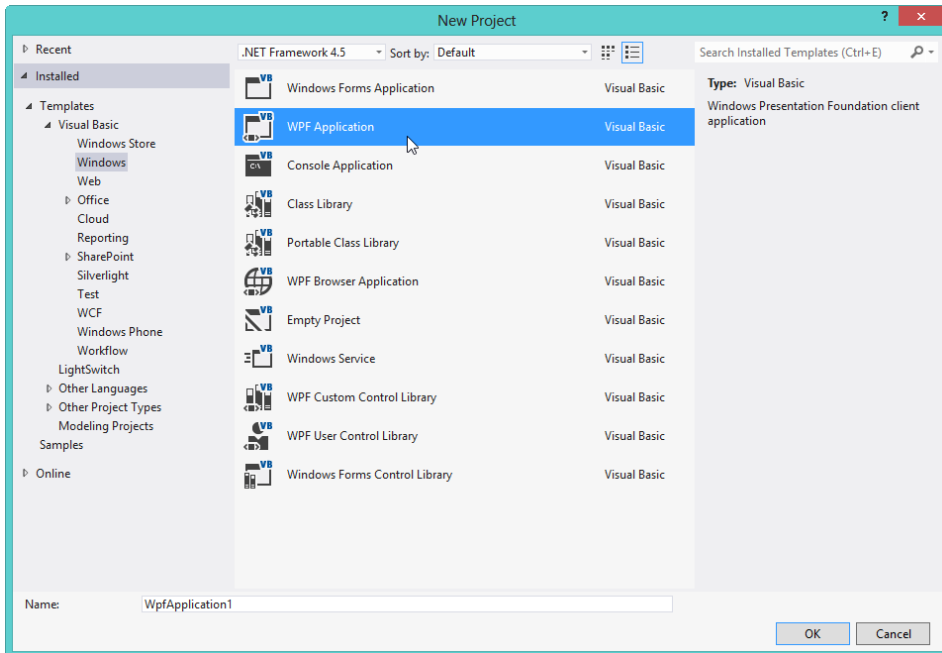
- **Windows Forms Application** พื้นที่เดิมที่หลายคนคุ้นเคยกันเป็นอย่างดีกับการพัฒนาแอปพลิเคชันที่มีส่วนแสดงผลแบบมาตรฐาน สามารถพัฒนาด้วยภาษา Visual Basic หรือ Visual C#



รูปที่ 1-5 แสดงโปรเจกต์แบบ Windows Forms Application ใน Visual Studio 2012 บน Windows 8

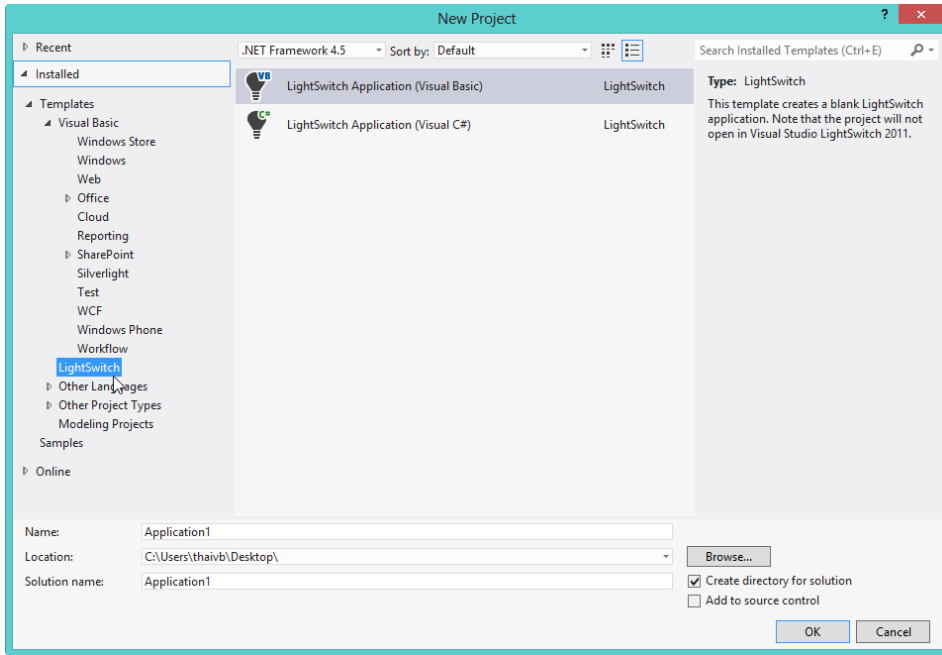


- **WPF Application** ในกรณีที่คุณต้องการพัฒนาแอปพลิเคชันที่สามารถแก้ไขส่วนแสดงผลให้ตรงตามความต้องการของคุณ โปรเจกต์ประเภทนี้ คือ คำตอบของคุณ โดยการแบ่งส่วนแสดงผล (อยู่ในความรับผิดชอบของ *สคริปต์ XAML*) กับส่วนการทำงาน (โค้ดภาษา Visual Basic หรือ Visual C#) ออกจากกันอย่างชัดเจน



รูปที่ 1-6 แสดงโปรเจกต์แบบ WPF Application ใน Visual Studio 2012 บน Windows 8

- **LightSwitch** พัฒนาแอปพลิเคชันบนแนวความคิดที่ว่า เขียนโค้ดให้น้อยที่สุด หรือไม่ต้องเขียนโค้ดเลย โดยใช้ระยะเวลาพัฒนาไม่นาน เรียกแนวความคิดนี้ว่า **Rapid Application Development (RAD)** ผลงานที่ได้ออกมา เพียงพอต่อการใช้งานจริงอีกด้วย

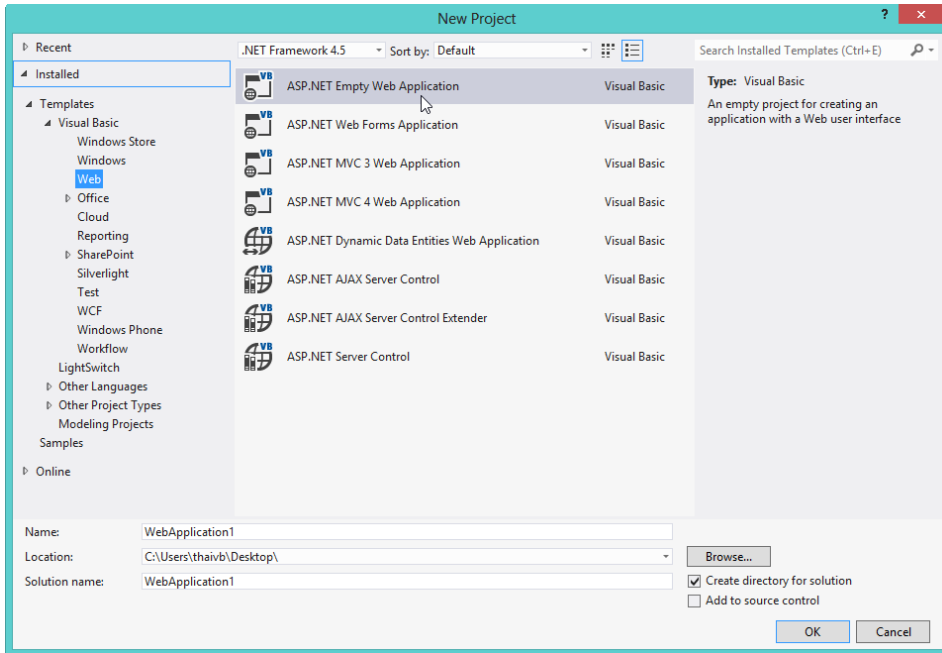


รูปที่ 1-7 แสดงโปรเจกต์แบบ LightSwitch ใน Visual Studio 2012 บน Windows 8

จากรูปที่ 1-7 โปรเจกต์แบบ LightSwitch ของ VS 2012 ครอบคลุมทั้ง Desktop Application (VB 2012 กับ VC# 2012) และ Web Application (ตามมาตราฐาน HTML5) ดังนั้นจึงเป็นอีก 1 ทางเลือกที่สำคัญมาก สำหรับโปรแกรมเมอร์สาย .NET ที่ต้องเรียนรู้

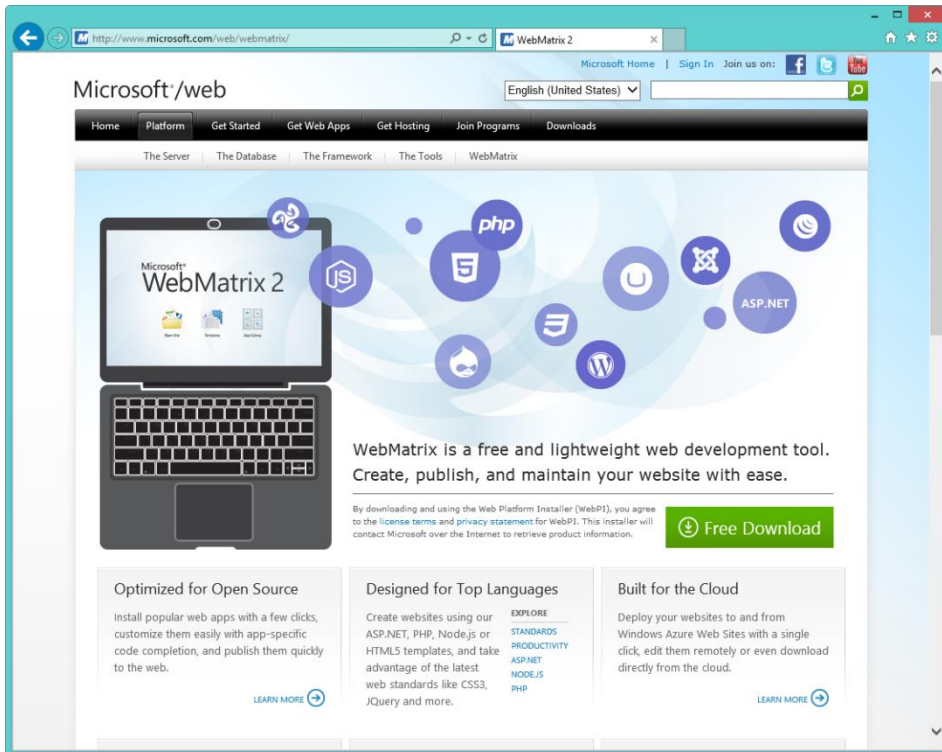
## 2.Web Application คุณมีทางเลือก 3 ทาง คือ

- **ASP.NET** เป็นการพัฒนาเว็บแอปพลิเคชันโดยการแยกส่วนแสดงผล (\*.aspx) กับโค้ดการทำงาน (ภาษา Visual Basic (\*.vb) หรือภาษา Visual C# (\*.cs)) ออกจากกัน



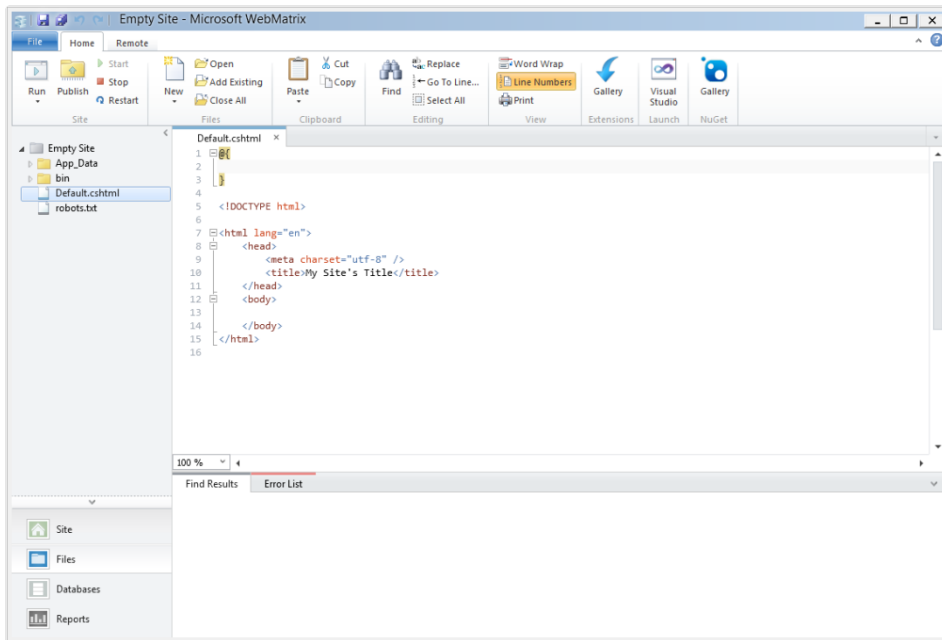
รูปที่ 1-8 แสดงโปรเจกต์ของ ASP.NET ใน Visual Studio 2012 บน Windows 8

- **ASP.NET Web Matrix** ไมโครซอฟท์สร้างโครงการที่เรียกว่า **Web Matrix** ขึ้นมา เพื่อเป็นอีก 1 ทางเลือกสำหรับนักพัฒนาเว็บแอปพลิเคชันด้วย ASP.NET ข้อแตกต่างที่สำคัญ ก็คือ ไมโครซอฟท์ รวมส่วนแสดงผลเข้ากับโค้ดการทำงาน แยกออกเป็น 2 กรณี คือ ภาษา Visual Basic (\*.vbhtml) และภาษา Visual C# (\*.cshtml)



รูปที่ 1-9 แสดงโครงการ Web Matrix ของไมโครซอฟท์

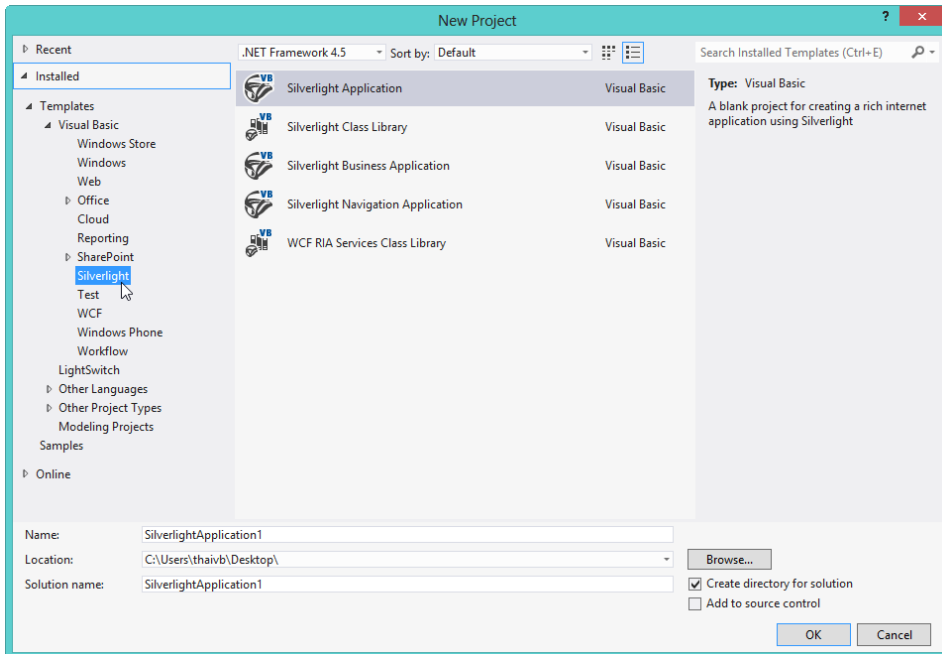
เครื่องมือที่ใช้พัฒนา ASP.NET Web Matrix ไมโครซอฟท์ปล่อยให้ นักพัฒนาสามารถ download ฟรีได้ที่ <http://www.microsoft.com/web/webmatrix/> เราเรียก ASP.NET Web Matrix อีกชื่อหนึ่งว่า ASP.NET ที่รันด้วย **Razor**



รูปที่ 1-10 แสดงสภาพแวดล้อมของ ASP.NET Web Matrix

ความสำคัญของโครงการ ASP.NET Web Matrix ก็คือ เป็นพื้นฐานของการพัฒนาเว็บแอปพลิเคชันด้วย ASP.NET MVC 4.0 โดยมีโครงสร้างการทำงานแบ่งออกเป็น 3 ส่วน คือ **Model - View - Controller**

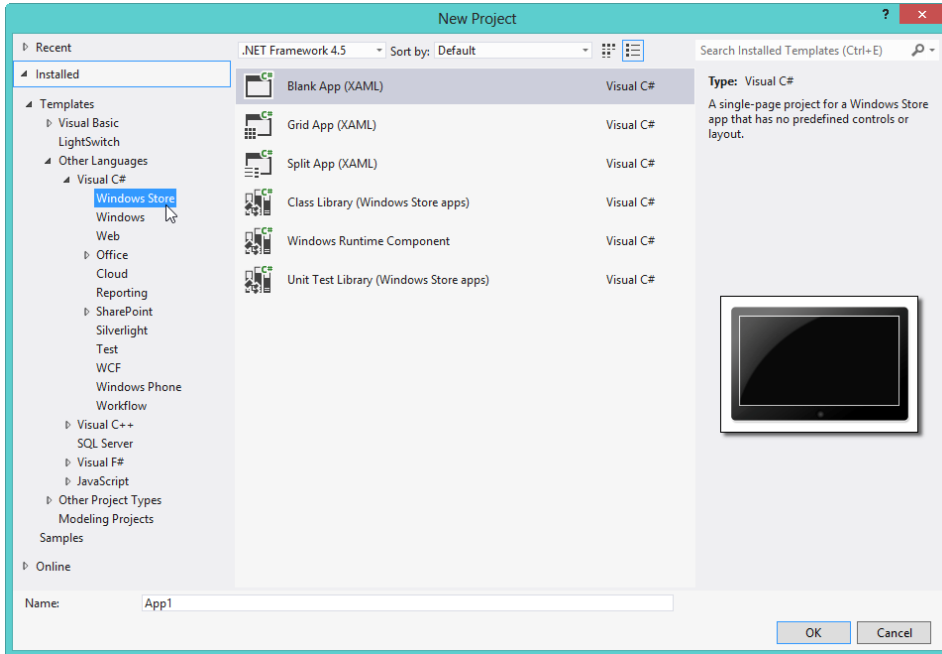
- **Silverlight** เรียกอีกอย่างหนึ่งว่า WPF Application ที่รันบนเว็บ ในอนาคตไมโครซอฟท์จะสนับสนุน HTML5 มากกว่า Silverlight



รูปที่ 1-11 แสดงโปรเจกต์แบบ SilverLight ใน Visual Studio 2012 บน Windows 8

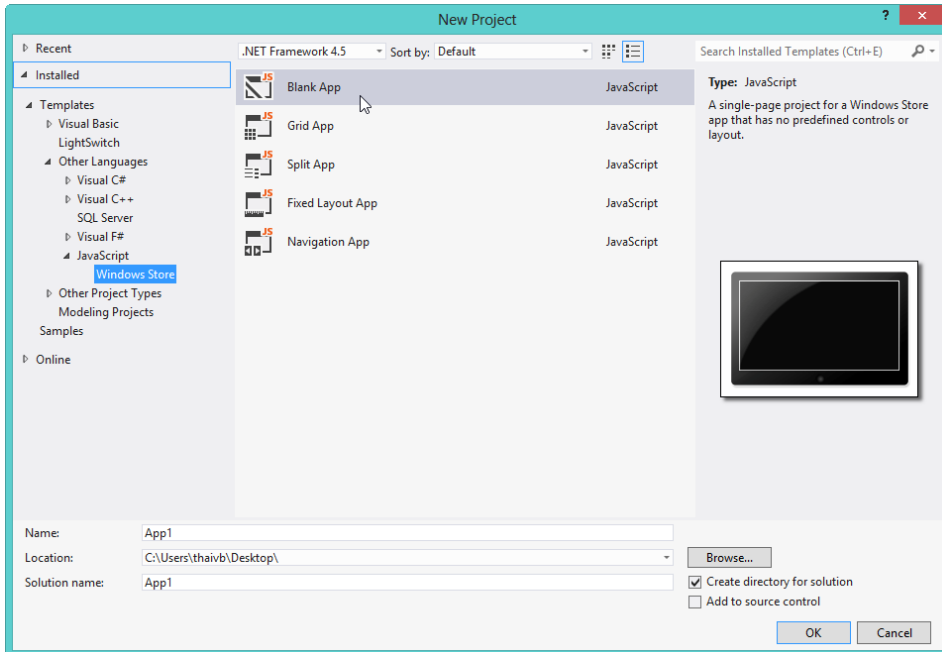
### 3.Windows 8 Apps คุณมีทางเลือก 2 ทาง คือ

- พัฒนาด้วยสคริปต์ XAML ร่วมกับภาษา Visual Basic 2012 หรือภาษา Visual C# 2012



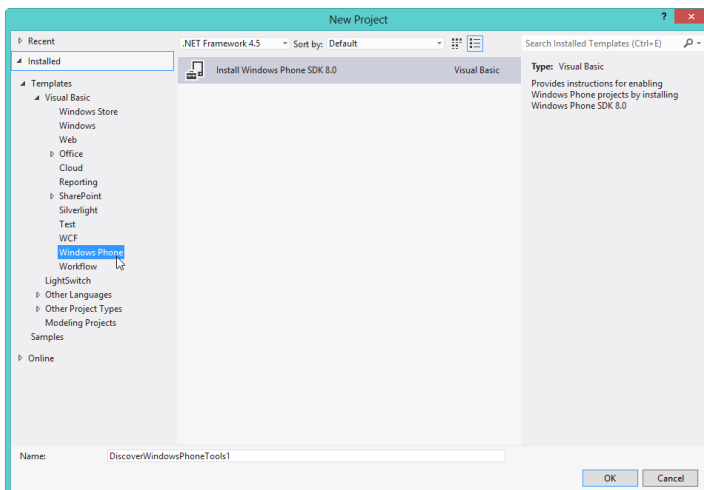
รูปที่ 1-12 แสดงโปรเจกต์แบบ Windows 8 Apps แบบ XAML ใน Visual Studio 2012 บน Windows 8

- พัฒนาด้วยภาษา HTML5 ร่วมกับภาษา JavaScript และ CSS3



รูปที่ 1-13 แสดงโปรเจกต์แบบ Windows 8 Apps แบบ HTML 5 ใน Visual Studio 2012 บน Windows 8

ในส่วนของ Smart Phone ที่รันด้วยระบบปฏิบัติการ Windows Phone 8 ผู้เขียนไม่ได้จัดเข้ามาในกลุ่ม เพราะว่า มีแนวโน้มไม่น้อยเช่นกันว่า มีโอกาสรวมเข้ากับ Windows 8 Apps



รูปที่ 1-14 แสดงโปรเจกต์แบบ Windows Phone ใน Visual Studio 2012 บน Windows 8



การแข่งขันในตลาด Smart Devices (Tablet และ Smart Phone) ในยุคปัจจุบัน มีการเปลี่ยนแปลงอยู่ตลอดเวลา ผู้เขียนไม่สามารถคาดเดาได้ว่า สถานะของโปรเจกต์แบบ Windows Phone จะเป็นอย่างไรในอนาคต ขอให้คุณผู้อ่านติดตามข่าวสาร ความคืบหน้าอย่างใกล้ชิด โดยที่ผู้เขียนจะมีการอัปเดตข่าวสารอยู่ตลอดเวลาที่แฟนเพจของผู้เขียนที่ <https://www.facebook.com/thaivb.net>



รูปที่ 1-15 แสดงแฟนเพจของผู้เขียน

จากโครงสร้างข้างต้น พบว่า **สคริปต์ XAML มีบทบาททั้งใน WPF Application, SilverLight, Windows Phone 8 และ Windows 8 Apps** ดังนั้น ในปัจจุบัน สิ่งทีโปรแกรมเมอร์สาย .NET ควรจะศึกษามากที่สุด นั่นคือ **สคริปต์ภาษา XAML** นั่นเอง ผู้เขียนคิดว่าคำถามแรกที่คุณอยากรู้ คำตอบมากที่สุด ก็คือ โปรแกรมเมอร์สาย .NET จะเดินไปในทิศทางใดต่อไป

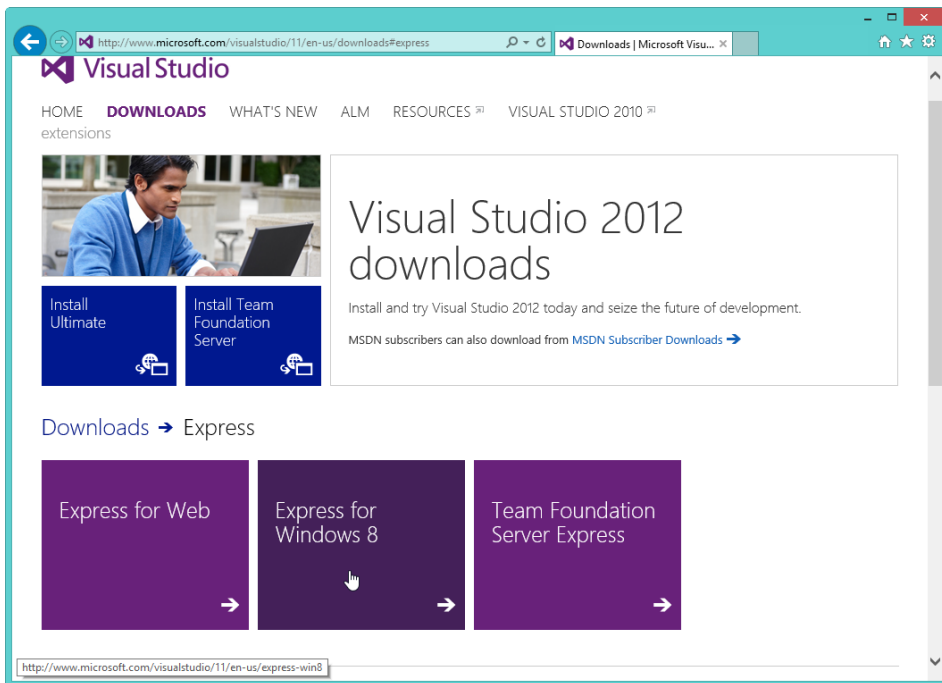
ในอดีตที่ผ่านมา สภาพแวดล้อมของเรามีแต่เพียงบน PC กับ Notebook เท่านั้น แต่ในปัจจุบันคุณไม่จำเป็นต้องเล่นอินเทอร์เน็ต เล่นแอปพลิเคชันบน PC หรือ Notebook อีกต่อไป คุณสามารถใช้งานอุปกรณ์จำพวก Tablet และ Smart Phone เข้ามาทำหน้าที่บางอย่างแทน PC หรือ Notebook ได้อีกด้วย แอปพลิเคชันที่คุณเล่นบนอุปกรณ์ Smart Devices เหล่านี้ ก็มาจากโปรแกรมเมอร์ในสายต่าง ๆ นั่นเอง

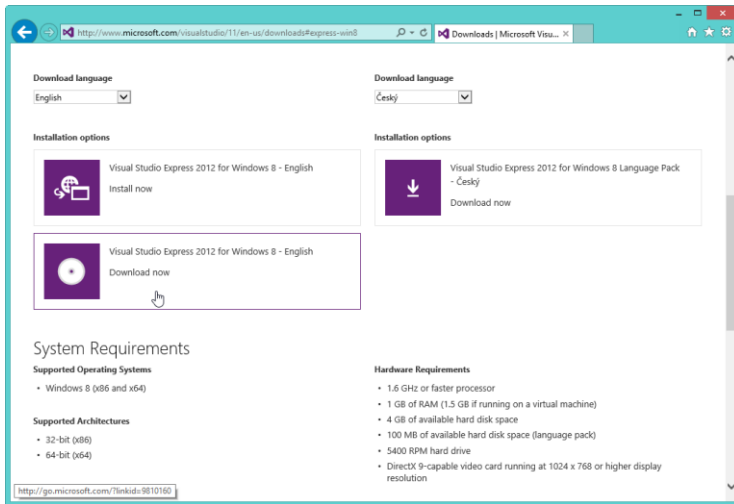
คำตอบที่ผู้เขียนมีให้กับคุณผู้อ่าน ก็คือ การเพิ่มทักษะให้กับตัวเอง เป็นผลดีกับคุณผู้อ่านอย่างแน่นอน การพัฒนาตัวเองในยุคปัจจุบัน ต้องเกิดทั้งทางกว้าง (เรียนรู้หลายเรื่อง) และทั้งทางลึก (ศึกษารายละเอียดให้มากที่สุด) ในเวลาเดียวกัน

ในวันที่คุณต้องลงนามจริง คุณไม่มีทางเลือกรู้ได้เลยว่า ในระหว่างการทำงานของคุณ คุณจะต้องประสบปัญหาอะไรบ้าง ต้องใช้เทคโนโลยีอะไรบ้าง สิ่งเดียวที่คุณรู้ตอนนี้ ก็คือ คุณต้องเตรียมตัวเองให้พร้อมกับสภาพแวดล้อมในปัจจุบันทุกทางเท่าที่จะทำได้

## การพัฒนา Windows 8 Apps ด้วย Visual Studio 2012 Express for Windows 8

คุณยังมีอีก 1 ทางเลือกในการพัฒนา Windows 8 Apps นั่นคือ อาศัย Visual Studio 2012 Express for Windows 8 ซึ่งเป็นเครื่องมือที่ไม่โครซอฟท์เปิดให้นักพัฒนาทั่วไป สามารถดาวน์โหลดมาใช้งานได้ฟรีแบบมีเงื่อนไขที่ <http://www.microsoft.com/visualstudio/11/en-us/downloads> มีขนาด 403 MB ดังรูปที่ 1-16





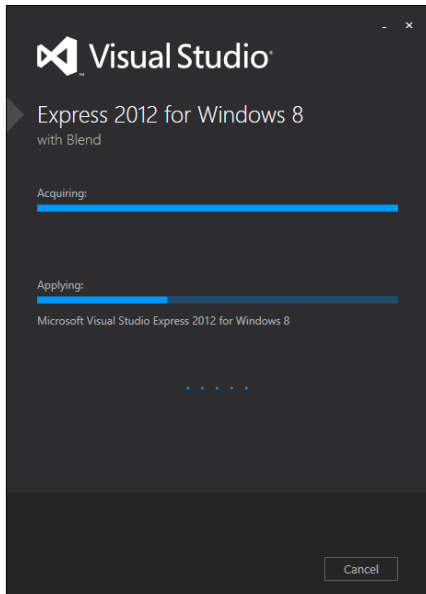
รูปที่ 1-16 แสดงเว็บไซต์ที่ให้ดาวน์โหลด Visual Studio 2012 Express for Windows 8

หลังจากที่คุณดาวน์โหลด Visual Studio 2012 Express for Windows 8 (ไฟล์ vs2012\_winexp\_enu.iso) มาแล้ว ขอให้คุณติดตั้งบน Windows 8 ตามขั้นตอนที่ปรากฏขึ้นมา มีขั้นตอนดังนี้

1. เริ่มต้น ให้คุณดับเบิลคลิกไฟล์ win8express\_full.exe เพื่อเริ่มต้นติดตั้ง Visual Studio 2012 Express for Windows 8

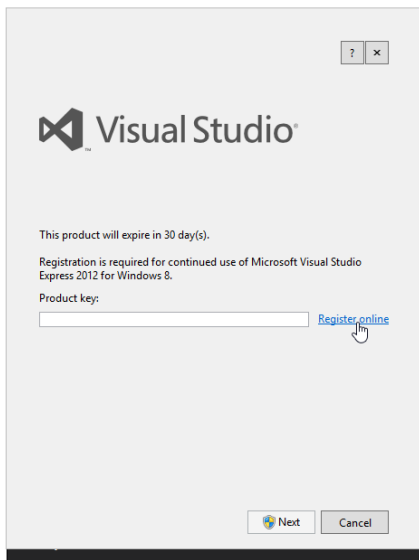


รูปที่ 1-17 แสดงหน้าจอเริ่มต้นติดตั้ง Visual Studio 2012 Express for Windows 8



รูปที่ 1-18 หลังจากติดตั้งเสร็จแล้ว

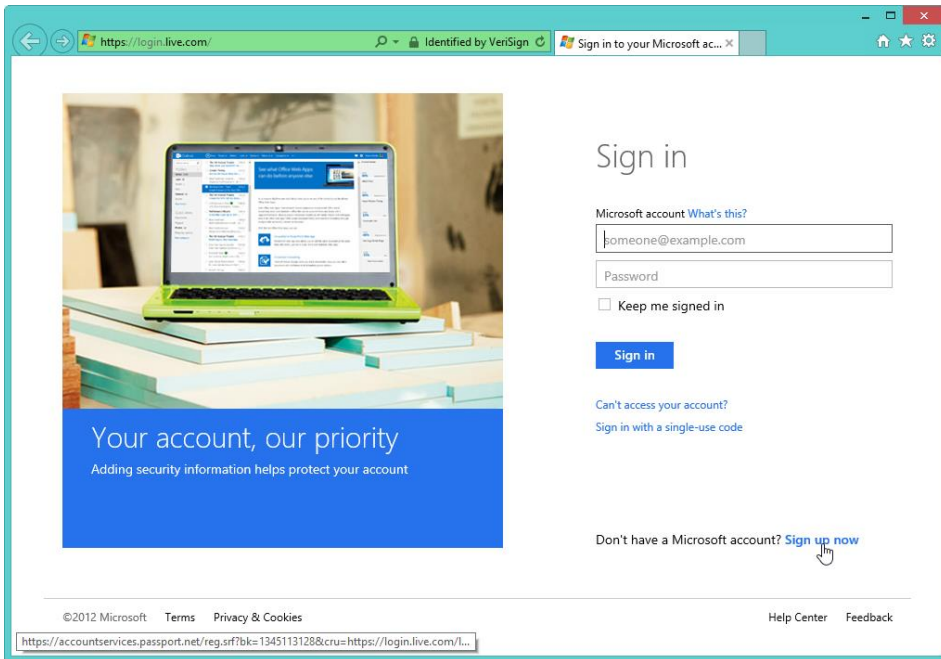
2. ในขั้นตอนสุดท้ายของการติดตั้ง จะมีการแสดงไดอะล็อกซ์ให้คุณป้อน Product key ดังรูปที่ 1-19



รูปที่ 1-19 แสดงขั้นตอนสุดท้ายของการติดตั้ง

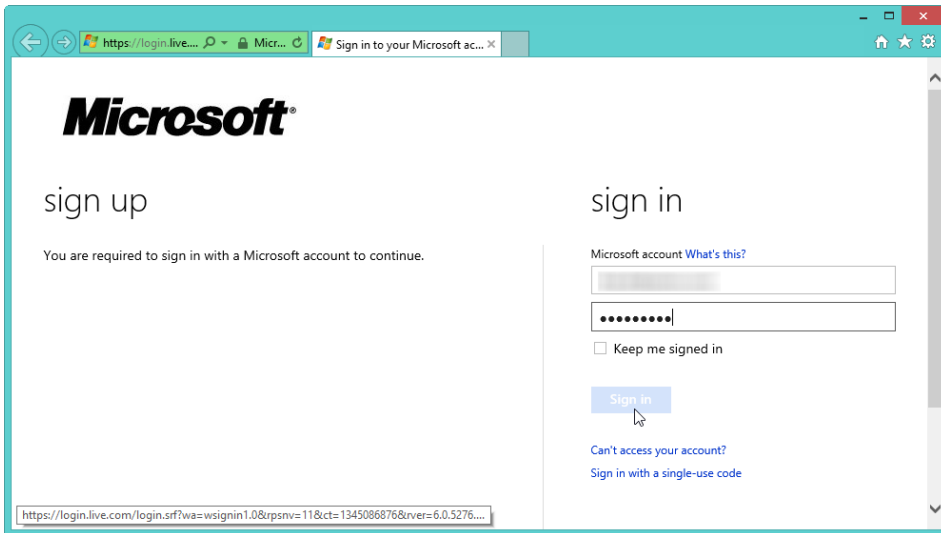
3.คุณสามารถขอ Product key ได้ฟรี โดยการคลิกที่ Register online ก็จะเข้าสู่ขั้นตอนการ sign in เข้าสู่ระบบของไมโครซอฟท์โดยใช้ e-mail ของคุณ แยกออกเป็น 2 กรณี

- กรณีใช้ e-mail อื่นๆ ที่ไม่อยู่ในเครือของไมโครซอฟท์ คุณต้องลงทะเบียน e-mail ของคุณ กับ Microsoft account ก่อน ได้ที่ <https://login.live.com>



รูปที่ 1-20 แสดงการลงทะเบียน e-mail อื่นกับ Microsoft account

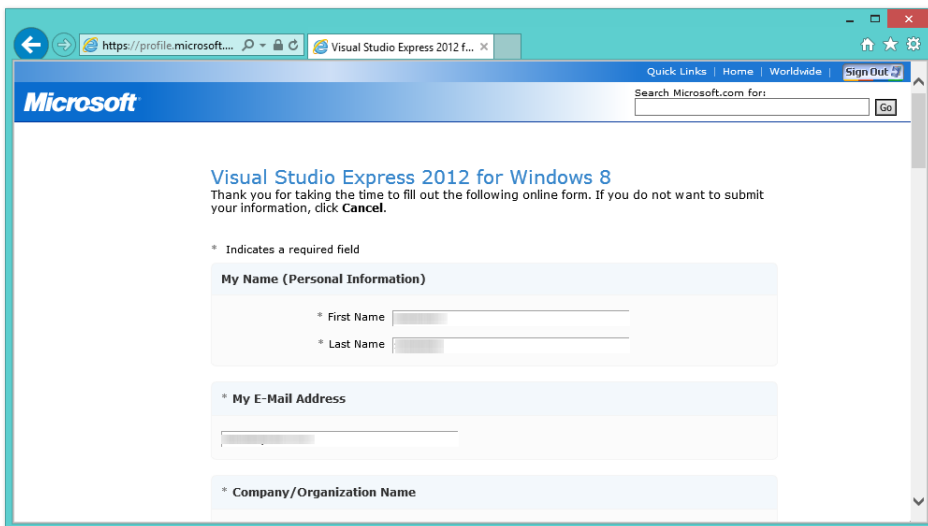
- กรณีใช้ e-mail ในเครือของไมโครซอฟท์ ([www.live.com/](http://www.live.com/) , [www.hotmail.com/](http://www.hotmail.com/) , [www.outlook.com/](http://www.outlook.com/) ) คุณสามารถใช้ e-mail ของคุณ sign in เข้าสู่ระบบ Microsoft account ดังรูปที่ 1-21



รูปที่ 1-21 แสดงการ *sign in* เพื่อขอ *Product key*

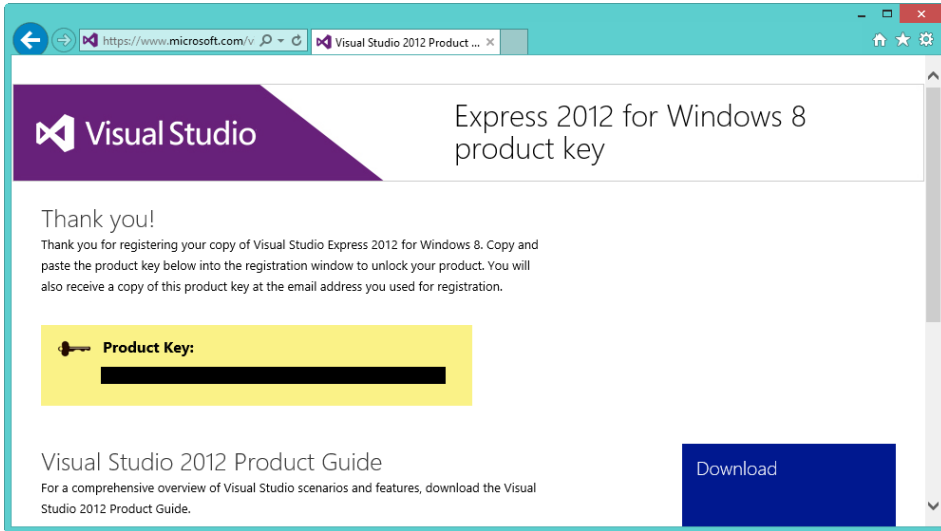
4.ต่อมา จะเข้าสู่ขั้นตอนการแก้ไขข้อมูลส่วนตัวของคุณใน Microsoft account จากนั้นคลิกที่ปุ่ม

ด้านล่างสุด เพื่อเข้าสู่ขั้นตอนต่อไป ดังรูปที่ 1-22



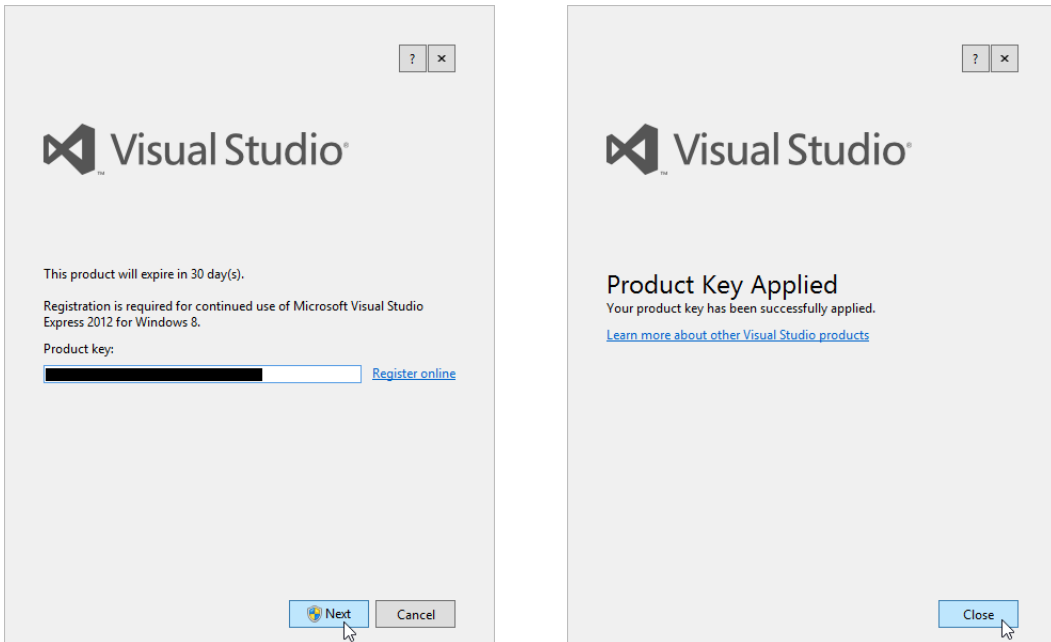
รูปที่ 1-22 แสดงขั้นตอนการป้อนข้อมูลส่วนตัวใน Microsoft account

5.ท้ายที่สุด คุณก็จะได้ Product key ตามที่คุณต้องการ ดังรูปที่ 1-23



รูปที่ 1-23 แสดง Product Key ที่ได้

6.ท้ายที่สุด ให้คุณนำ Product key ที่ได้ มาใช้ใน Visual Studio 2012 Express for Windows 8 ก็จะทำให้คุณสามารถใช้งานโปรแกรมได้ตามที่คุณต้องการแล้ว ดังรูปที่ 1-24



รูปที่ 1-24 แสดงการนำ Product key ที่ได้ มาใช้ใน Visual Studio 2012 Express for Windows 8

## การขอสิทธิชั่วคราวในการพัฒนา Windows 8 Apps

การพัฒนา Windows 8 Apps ด้วยโปรเจกต์แบบ Windows Store ในครั้งแรก ต้องมีการขอสิทธิ์ที่เรียกว่า **Developer License** โดยที่ผู้เขียนขอเรียกว่า เป็นการขอสิทธิ์ชั่วคราว (เพราะว่ามีวันหมดอายุ) เพื่อให้คุณสามารถสร้าง Windows 8 Apps ด้วย VS 2012 ได้นั่นเอง



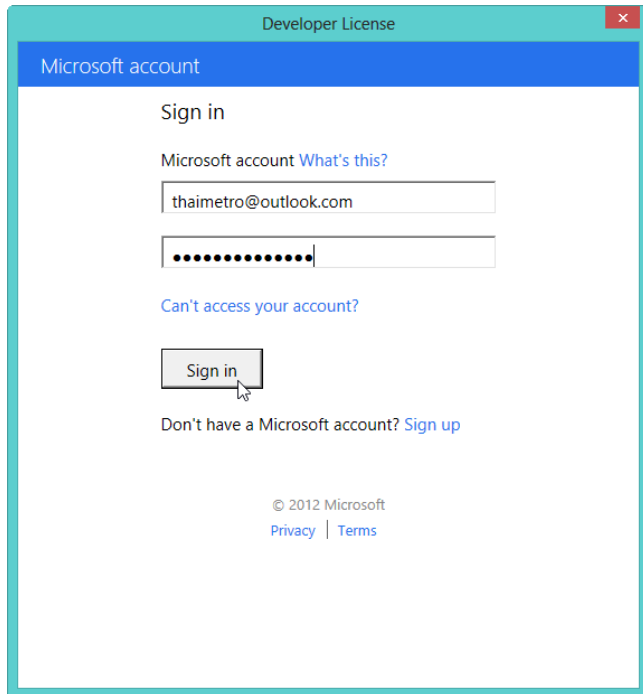
รูปที่ 1-25 แสดงไดอะล็อกยืนยันสิทธิ์ ที่ปรากฏขึ้นมาในครั้งแรก

จากรูปที่ 1-25 VS 2012 (ทุก Edition) จะแสดงไดอะล็อกนี้ขึ้นมาในครั้งแรก เพื่อให้คุณยืนยันสิทธิ์ในการพัฒนา Windows 8 Apps ด้วยโปรเจกต์แบบ Windows Store



คุณต้องขอสิทธิ์ชั่วคราวก่อน แยกออกเป็น 2 กรณี คือ

1.ให้คุณ Sign in ด้วย e-mail ที่คุณลงทะเบียนกับ Microsoft account ดังรูปที่ 1-26



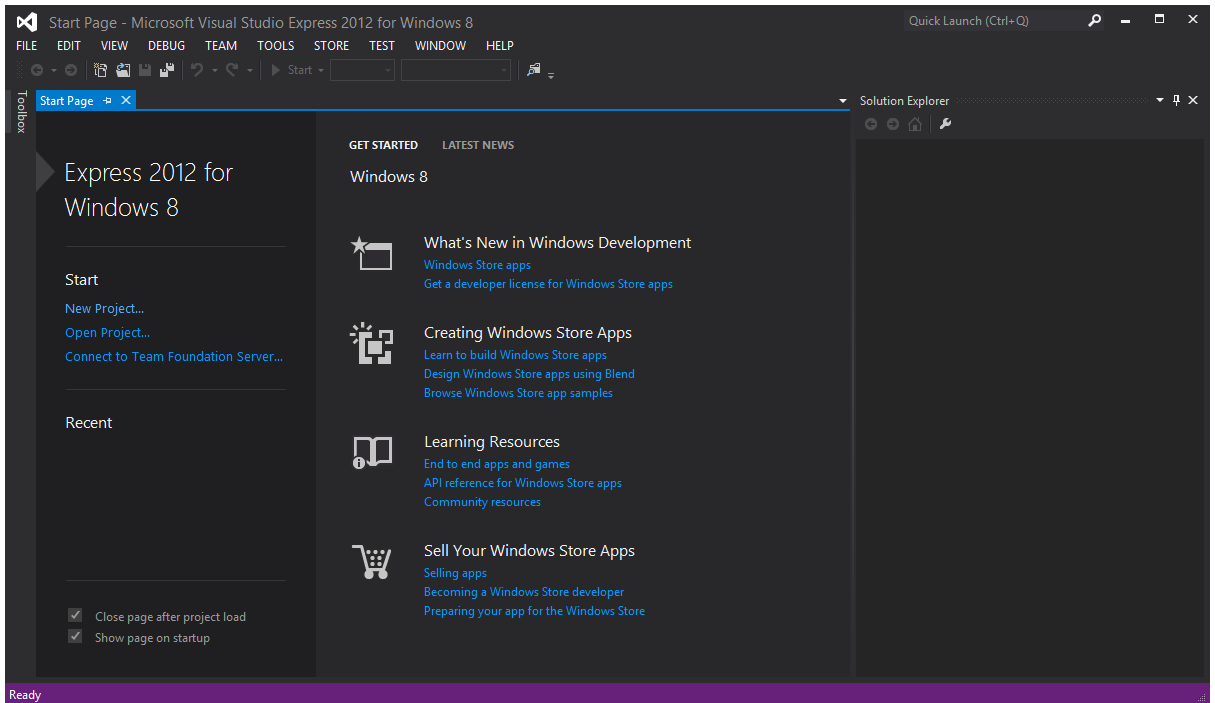
รูปที่ 1-26 แสดงการ Sign in เข้าสู่ระบบ Microsoft account

ต่อมา จะมีไดอะล็อก Developer License แจ้งให้คุณทราบว่า สิทธิชั่วคราวแบบ Developer license ที่คุณได้รับ จะหมดอายุเมื่อใด ดังรูปที่ 1-27

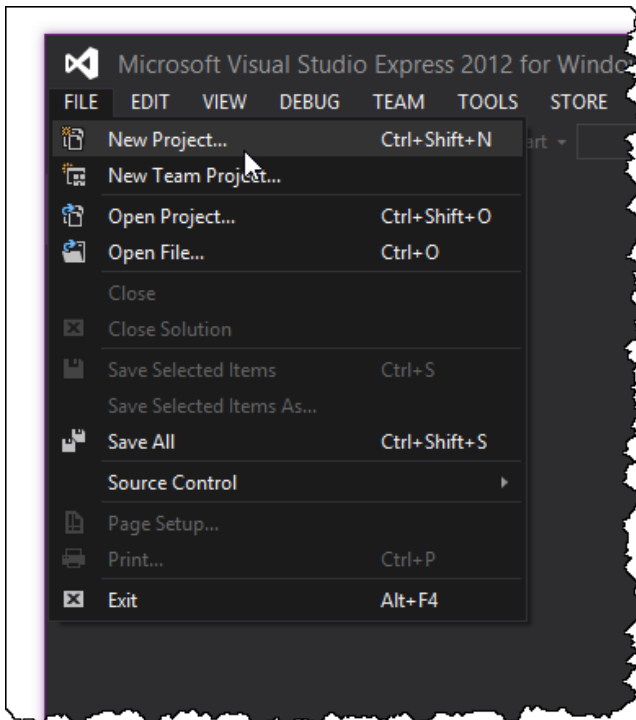


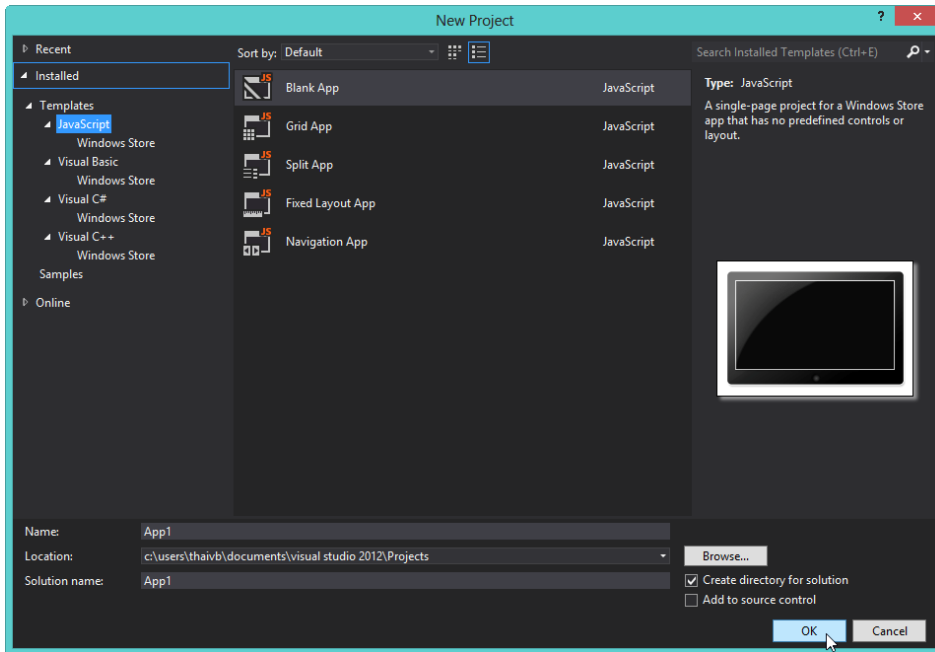
รูปที่ 1-27 แสดงสิทธิชั่วคราวแบบ Developer License ที่ผู้เขียนได้รับ

จากรูปที่ 1-27 สิทธิที่คุณได้รับ ส่งผลให้คุณสามารถพัฒนา Windows 8 Apps ได้แล้ว

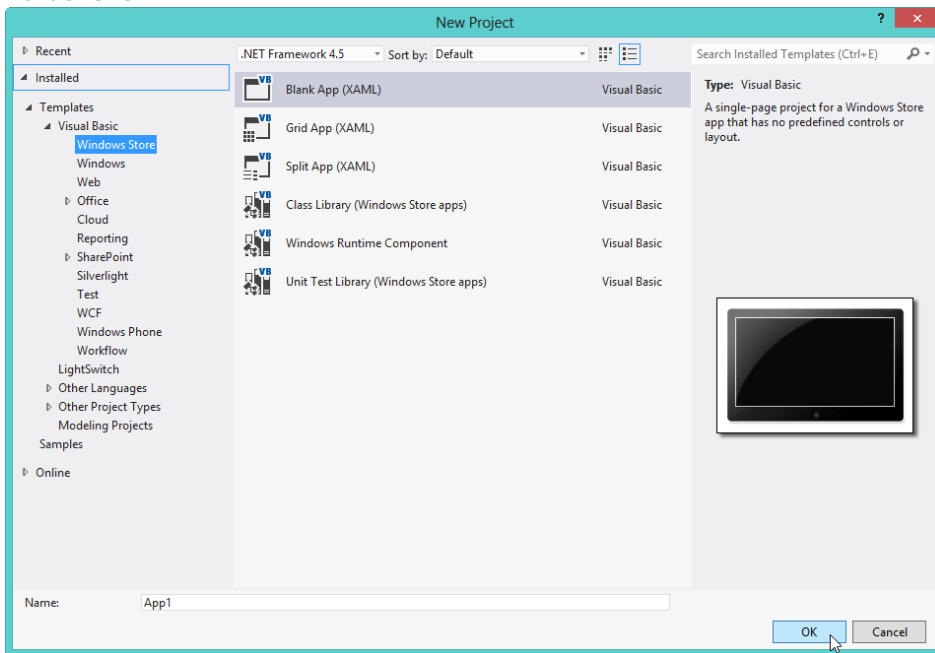


รูปที่ 1-28 แสดงสภาพแวดล้อมของ Visual Studio 2012 Express for Windows 8 ในครั้งแรก





รูปที่ 1-29 แสดงการสร้างโปรเจกต์แบบ Windows Store ใน Visual Studio 2012 Express for Windows 8

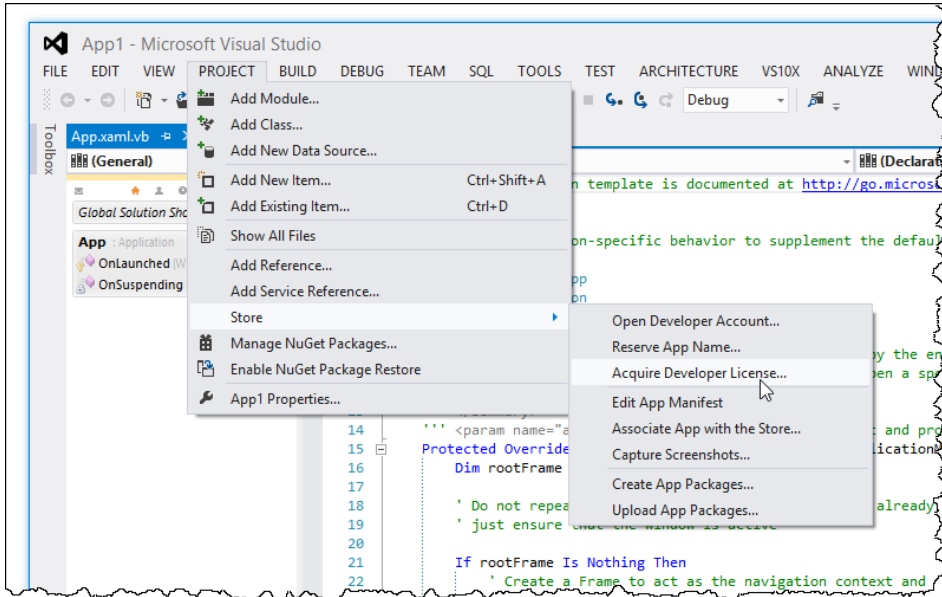


รูปที่ 1-30 แสดงโปรเจกต์แบบ Windows Store ของ VS 2012 ตั้งแต่เวอร์ชัน Professional ขึ้นไป

## การขอหรือต่ออายุ Developer license

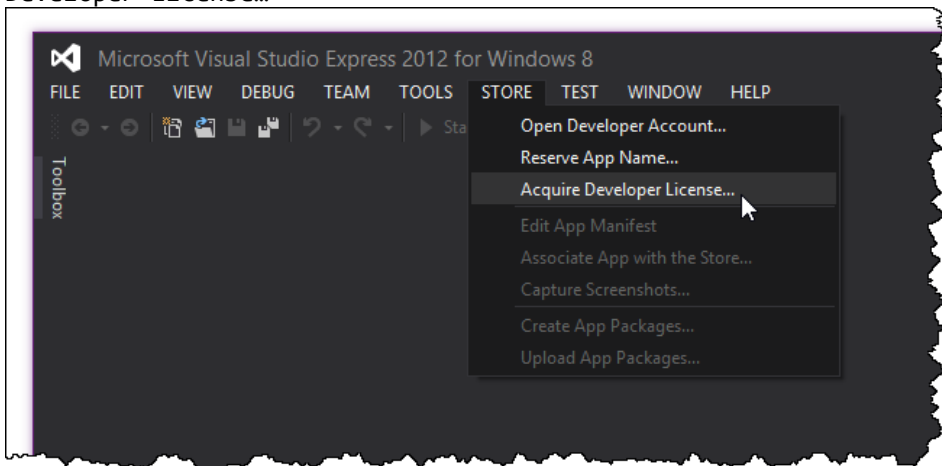
แยกออกเป็น 2 กรณี คือ

1.กรณี **Visual Studio 2012** ตั้งแต่เวอร์ชัน **Professional** ขึ้นไป คุณสามารถเลือกเมนู **PROJECT** > **Store** > **Acquire Developer License...** เพื่อขอหรือต่ออายุสิทธิ์ชั่วคราว



รูปที่ 1-31 กรณี Visual Studio 2012 ตั้งแต่เวอร์ชัน Professional ขึ้นไป

2.กรณี **Visual Studio 2012 Express for Windows 8** ให้คุณเลือกเมนู **STORE** > **Acquire Developer License...**



รูปที่ 1-32 กรณี Visual Studio 2012 Express for Windows 8

## โครงสร้างของ Windows 8 Apps

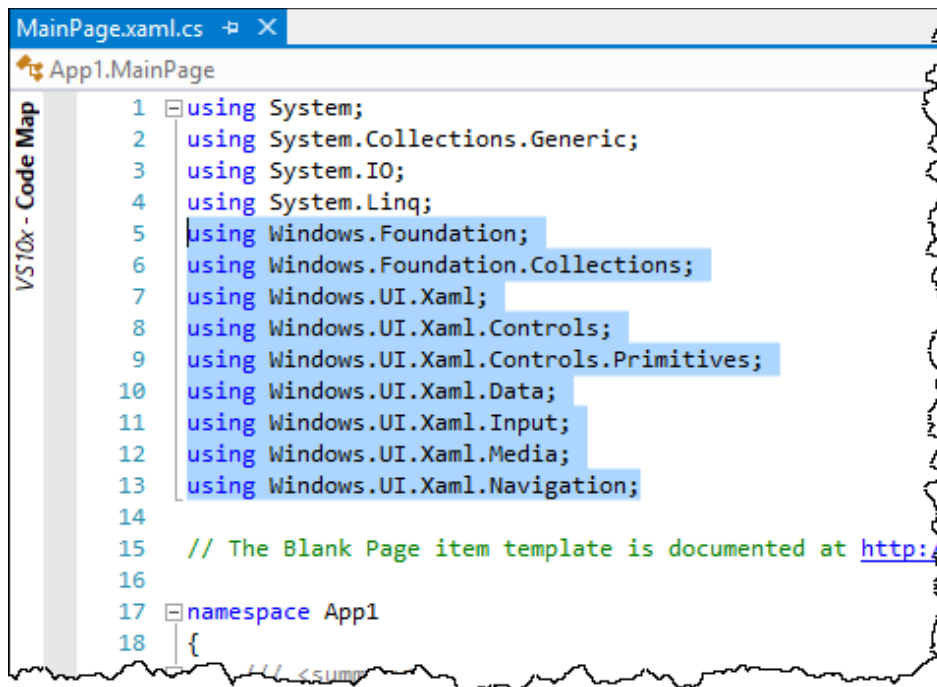
คุณสามารถพัฒนา Windows 8 Apps ได้ 2 แบบ คือ

1. อาศัยสคริปต์ XAML ร่วมกับภาษา VB 2012, VC# 2012 และ C++ 2012
2. อาศัยสคริปต์ HTML5 ร่วมกับ CSS3 & JavaScript

1. อาศัยสคริปต์ XAML ร่วมกับภาษา VB 2012, VC# 2012 และ C++ แต่เนื้อหาที่ผู้เขียนนำเสนอครอบคลุม 2 ภาษาเท่านั้น คือ VB 2012 & VC# 2012

สำหรับโค้ด VB 2012 & VC# 2012 ที่นำมาใช้ในการพัฒนา Windows 8 Apps เกิดจาก 2 ส่วนด้วยกัน คือ

- ความรู้เดิมจาก .NET เวอร์ชันเก่า โดยส่วนใหญ่ ยังคงใช้ใน Windows 8 Apps ได้เช่นเดิม อยู่ในฐานะเป็นกลุ่มคลาสที่อยู่ภายใต้เนมสเปซ System.\* อาจจะมีข้อแตกต่าง หรือข้อจำกัดไปจากเดิมบ้าง ขึ้นอยู่กับรายละเอียด ซึ่งจะเป็นเนื้อหาในลำดับถัดไป
- สิ่งที่ต้องศึกษาเพิ่มเติม อยู่ภายใต้เนมสเปซที่ขึ้นต้นด้วย Windows.\* รั้นภายใต้ WinRT นี้คือ สิ่งที่เราต้องเรียนรู้ว่า ในการพัฒนา Windows 8 Apps ขึ้นมา ประกอบด้วยอะไรบ้าง



```
MainPage.xaml.cs
App1.MainPage
VS10x - Code Map
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using Windows.Foundation;
6 using Windows.Foundation.Collections;
7 using Windows.UI.Xaml;
8 using Windows.UI.Xaml.Controls;
9 using Windows.UI.Xaml.Controls.Primitives;
10 using Windows.UI.Xaml.Data;
11 using Windows.UI.Xaml.Input;
12 using Windows.UI.Xaml.Media;
13 using Windows.UI.Xaml.Navigation;
14
15 // The Blank Page item template is documented at http://
16
17 namespace App1
18 {
```

รูปที่ 1-33 แสดงเนมสเปซในโปรเจกต์ Windows 8 Apps ด้วย VC# 2012

จากรูปที่ 1-33 มาจากโปรเจกต์ Windows 8 Apps ด้วย VC# 2012 แสดงให้คุณเห็นชัดเจนที่สุด กล่าวคือ เมื่อคุณสร้างโปรเจกต์ Windows 8 Apps ว่างๆ ขึ้นมา มีการอ้างอิงเนมสเปซพื้นฐานเท่าที่จำเป็น ในการพัฒนา Windows 8 Apps ขั้นต้นให้คุณโดยอัตโนมัติ ประกอบด้วยเนมสเปซ 2 กลุ่ม คือ

- กลุ่ม .NET เดิม (System.\*)
- กลุ่ม WinRT (Windows.\*)

ส่วนคุณผู้อ่านที่เคยพัฒนา WPF Application มาก่อน การย้ายมาพัฒนา Windows 8 Apps ถือว่า คุณอยู่ในฐานะครอบครัวเดียวกัน คิดแบบง่ายๆ ว่า บ้านของคุณใหญ่ขึ้น ถ้าคุณผู้อ่านเลือกสายนี้ สิ่งที่คุณได้ คือ คุณสามารถพัฒนาได้ทั้ง WPF Application และ Windows 8 Apps

ส่วนแสดงผลของ Windows 8 Apps ในขั้นต้น เก็บอยู่ในไฟล์ที่ชื่อว่า MainPage.xaml ดังสคริปต์ XAML ต่อไปนี้

```
XAML
<Page
  x:Class="App1.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:App1"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">

  <Grid Background="{StaticResource ApplicationPageBackgroundThemeBrush}">

  </Grid>
</Page>
```

ทุกๆ ครั้งที่คุณสร้าง Windows 8 Apps ด้วยโปรเจกต์ชนิด Blank Application ส่งผลให้ VS 2012 จะสร้างสคริปต์ XAML ข้างต้นให้คุณโดยอัตโนมัติ เป็นส่วนแสดงผลแบบว่างๆ มีสิ่งที่น่าสนใจ ในขั้นต้น คือ

1. เราถือว่า อีลีเมนต์ <Page>...</Page> ทำหน้าที่เป็น **ตัวบรรจุ (Container Control)** คอนโทรลอื่นๆ ในระดับบนสุด และยังทำหน้าที่เป็นสภาพแวดล้อมใน Windows 8 Apps อีกด้วย
2. VS 2012 กำหนดให้อีลีเมนต์ <Grid>...</Grid> ทำหน้าที่เป็นตัวบรรจุคอนโทรลอื่นๆ ในส่วนแสดงผล อยู่ในระดับรองลงมาจากอีลีเมนต์ <Page>...</Page> มีส่วนแสดงผลเป็นแบบเป็นตาราง คุณสามารถเปลี่ยนเป็นคอนโทรลบรรจุตัวอื่นได้เช่นกัน ผู้เขียนจะกล่าวถึงในเนื้อหาลำดับต่อไป

2.อาคัยสคริปต์ HTML5 ร่วมกับ CSS3 & JavaScript สิ่งที่สำคัญที่สุด ก็คือ HTML5 ที่นำมาใช้ในการพัฒนา Windows 8 Apps มีทั้งส่วนที่เป็นมาตรฐานเดิมของ HTML5 และส่วนที่เพิ่มเติมขึ้นมาโดยไมโครซอฟท์ เพื่อเพิ่มประสิทธิภาพในการทำงานและสร้างส่วนแสดงผลใน Windows 8 Apps นั้นเอง เช่น กลุ่มแอ็ททริบิวต์ที่ขึ้นต้นด้วยคำว่า data- ฯลฯ เป็นต้น ถ้าคุณผู้อ่านเลือกสายนี้ สิ่งที่คุณได้ คือ การสร้างเว็บไซต์ด้วยมาตรฐาน HTML5, การจัดส่วนแสดงผลด้วย CSS3 และ Windows 8 Apps อยู่ในไฟล์ที่ชื่อว่า default.html

```
HTML5
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>App1</title>

  <!-- WinJS references -->
  <link href="//Microsoft.WinJS.1.0/css/ui-dark.css" rel="stylesheet" />
  <script src="//Microsoft.WinJS.1.0/js/base.js"></script>
  <script src="//Microsoft.WinJS.1.0/js/ui.js"></script>

  <!-- App1 references -->
  <link href="/css/default.css" rel="stylesheet" />
  <script src="/js/default.js"></script>
</head>
<body>
  <p>Content goes here</p>
</body>
</html>
```

สำหรับคุณผู้อ่านที่เป็นมือใหม่จริงๆ การเริ่มต้นศึกษา เข้ามาสู่โลกของ Windows 8 Apps ไม่ใช่เรื่องยากเย็นอีกต่อไป เพราะว่า ไมโครซอฟท์เตรียมเครื่องมือ และจัดโครงสร้างของ Windows 8 Apps ให้ง่ายและสะดวกต่อการเรียนรู้เป็นอย่างมาก นี่คือ โจทย์ของผู้เขียนในการนำเสนอหนังสือในซีรีส์ของ Windows 8 Apps

## รูปแบบการนำเสนอตัวอย่างในหนังสือเล่มนี้

รูปแบบการนำเสนอสคริปต์ XAML และโค้ดการทำงาน (VB 2012 และ VC# 2012) ผู้เขียนแยกออกเป็น 2 กรณี

**1.กรณีสคริปต์ XAML** ผู้เขียนลงเฉพาะสคริปต์ที่เป็นเนื้อหาเท่านั้น เพื่อให้สะดวกต่อการนำเสนอ ส่วนสคริปต์ XAML ที่เกิดจาก VS 2012 สร้างให้คุณอัตโนมัติ ผู้เขียนขอละไว้ในฐานที่เข้าใจว่า เป็นสคริปต์ XAML ที่ไม่มีความเกี่ยวข้องใดๆ ต่อเนื้อหาในหัวข้ออื่นๆ เพื่อให้คุณผู้อ่านทราบว่า นี่คือ สคริปต์ XAML ที่คุณต้องเขียนเอง หรือเป็นเนื้อหาที่กำลังนำเสนออยู่

แต่ถ้าเนื้อหามีความเกี่ยวข้อง หรือเชื่อมโยงกัน ผู้เขียนจะลงสคริปต์ XAML ดังกล่าวทั้งหมด เพื่อให้คุณผู้อ่านเห็นความสัมพันธ์ระหว่างสคริปต์ที่ VS 2012 สร้างให้คุณโดยอัตโนมัติ กับสคริปต์ที่คุณต้องเข้าไปแก้ไข หรือเขียนเอง

**2.กรณีโค้ดทำงาน (VB 2012 และ VC# 2012)** สำหรับโค้ดการทำงานทุกตัวอย่าง ทุกหัวข้อที่นำเสนอในหนังสือเล่มนี้ ผู้เขียนลงโค้ดครบถ้วนทั้งหมด

สำหรับตัวอย่างโค้ดและสคริปต์ของหนังสือเล่มนี้ จะใช้ตารางแต่ละสีทำหน้าที่กำกับโค้ดไว้ สำหรับกรอบสีแดง หมายถึง โค้ดหรือสคริปต์ที่มีข้อผิดพลาด ผู้เขียนยกตัวอย่างขึ้นมาเพื่อเปรียบเทียบเนื้อหาในหัวข้ออื่นๆ

XAML

VB 2012

VC# 2012

Error

XML

XSLT



## รูปแบบการลงโค้ด (VB 2012 & VC# 2012) และสคริปต์ XAML ในหนังสือเล่มนี้

เพื่อให้คุณผู้อ่านและผู้เขียนเข้าใจตรงกัน รูปแบบการลงโค้ด (VB 2012 & VC# 2012) และสคริปต์ XAML ของหนังสือเล่มนี้ มีรายละเอียดดังนี้

1. **สคริปต์ XAML** สำหรับสคริปต์ XAML ที่เกิดจาก VS 2012 สร้างให้คุณโดยอัตโนมัติ ผู้เขียนไม่ลงในหนังสือเล่มนี้ ดังสคริปต์ XAML ต่อไปนี้

```
XAML
<Page
  x:Class="App1.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:App1"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">

  <Grid Background="{StaticResource ApplicationPageBackgroundThemeBrush}">

  </Grid>
</Page>
```

สคริปต์ XAML ข้างต้น เป็นโครงสร้างของส่วนแสดงผลที่ VS 2012 สร้างให้คุณโดยอัตโนมัติ เป็นส่วนที่ผู้เขียนไม่ลงในหนังสือเล่มนี้ **ผู้เขียนลงเฉพาะสคริปต์ XAML ที่คุณต้องเขียนเอง** ซึ่งเป็นเนื้อหาของหัวข้ออื่นๆ

แต่ถ้าเนื้อหาดังกล่าว เกี่ยวข้องกับสคริปต์ XAML ที่ VS 2012 สร้างให้คุณโดยอัตโนมัติ ผู้เขียนจะลงสคริปต์ XAML เต็มรูปแบบ เพื่อให้คุณเห็นรายละเอียดอย่างชัดเจน โดยที่จะระบุชื่อไฟล์ \*.xaml กำกับสคริปต์ไว้ด้วย เพื่อให้คุณทราบว่า เป็นสคริปต์ XAML ของไฟล์ใด

โดยที่ผู้เขียนจะแยกสคริปต์ออกเป็น 2 ส่วน คือ

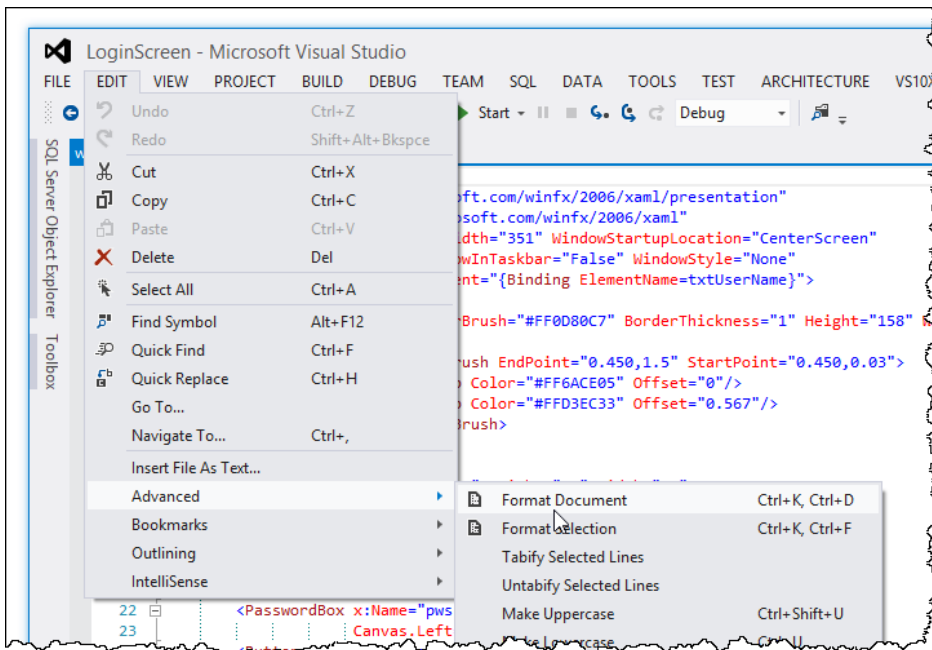
- สคริปต์ที่เกิดจาก VS 2012 สร้างให้คุณโดยอัตโนมัติ จะแสดงสคริปต์เป็นตัวสีเทา
- สคริปต์ที่คุณต้องเขียนเอง จะลงด้วยสีตามปกติที่ได้จาก VS 2012

2. **โค้ด VB 2012 & VC# 2012** ผู้เขียนถือว่าเป็นโค้ดการทำงานที่คุณต้องเขียนเองทั้งหมด ดังนั้น โค้ดส่วนนี้ **ผู้เขียนลงครบทั้ง 2 ภาษา ทุกตัวอย่าง** มีชื่อไฟล์ \*.vb หรือ \*.cs กำกับไว้ชัดเจนว่า โค้ดดังกล่าว อยู่ในไฟล์ใดเช่นกัน

3.กรณีเขียนสคริปต์หรือโค้ดใหม่เพิ่มเข้าไปในสคริปต์หรือโค้ดที่ถูกสร้างขึ้นมาโดย VS 2012 สำหรับกรณีนี้ เกิดจากการทำงานร่วมกันระหว่างสคริปต์หรือโค้ด ที่เราต้องเขียนขึ้นมาเอง กับสคริปต์หรือโค้ดที่มาจาก VS 2012 โดยอัตโนมัติ ผู้เขียนแยกแยะโดยการกำหนดให้สคริปต์หรือโค้ดที่คุณต้องเขียนขึ้นมาเองเป็นสี ส่วนสคริปต์หรือโค้ดที่มาจาก VS 2012 โดยอัตโนมัติ ผู้เขียนเลือกใช้สีเทา เพื่อให้เกิดความแตกต่าง

**การจัดระเบียบโค้ดหรือสคริปต์ใน VS 2012**

มีคำสั่งใน VS 2012 ที่น่าสนใจ ทำหน้าที่จัดระเบียบโค้ดหรือสคริปต์ให้ดูง่ายยิ่งขึ้น โดยการเลือกเมนู EDIT > Advanced > Format Document



รูปที่ 1-34 แสดงการเลือกคำสั่ง Format Document

```
GroupDetailPage.xaml
37
38 <!-- Horizontal scrolling grid used in most view states -->
39 <GridView
40     x:Name="itemGridView"
41     AutomationProperties.AutomationId="ItemGridView"
42     AutomationProperties.Name="Items In Group"
43     TabIndex="1"
44     Grid.RowSpan="2"
45     Padding="120,126,120,50"
46     ItemsSource="{Binding Source={StaticResource itemsViewSource}}"
47     ItemTemplate="{StaticResource Standard500x130ItemTemplate}"
48     SelectionMode="None"
49     IsSwipeEnabled="false"
50     IsItemClickEnabled="True"
51     ItemClick="ItemView_ItemClick">
52
53     <GridView.Header>
54         <StackPanel Width="480" Margin="0,4,14,0">
55             <TextBlock Text="{Binding Subtitle}" Margin="0,0,18,20" Style="{StaticResource SubtitleTextBlock}">
56                 <Image Source="{Binding Image}" Height="400" Margin="0,0,18,20" Stretch="UniformToFill" Style="{StaticResource Image}">
57                 <TextBlock Text="{Binding Description}" Margin="0,0,18,0" Style="{StaticResource DescriptionTextBlock}">
58             </StackPanel>
59     </GridView.Header>
60     <GridView.ItemContainerStyle>
61         <Style TargetType="FrameworkElement">
62             <Setter Property="Margin" Value="52,0,0,10"/>
63         </Style>
64     </GridView.ItemContainerStyle>
65 </GridView>
66
67 <!-- Vertical scrolling list only used when snapped -->
68 <ListView
```

รูปที่ 1-35 ผลการจัดระเบียบโค้ดหรือสคริปต์ด้วย VS 2012

### สรุปท้ายบท

เนื้อหาในบทนี้ เป็นการกล่าวถึงในภาพรวมของความสัมพันธ์ระหว่าง Windows 8 กับ Visual Studio 2012 รวมไปถึงทิศทางในอนาคตของโปรแกรมสาย .NET

ศุภชัย สมพานิช

Thaivb.NET

พัฒนาแอปพลิเคชันด้านฐานข้อมูลด้วย

# Entity Framework 5

ใน VB 2012 และ VC# 2012 สำหรับมือใหม่

Visual Studio 2012

การใช้งาน LINQ to Entities ตั้งแต่เริ่มต้นจนใช้งานได้จริง  
ทำงานกับระบบฐานข้อมูลแบบไวยากรณ์ SQL  
เนื้อหาพร้อมประยุกต์ใช้งานจริงในการพัฒนาระบบงาน  
ตัวอย่าง Workshop

- เหมาะกับผู้เริ่มต้นจนใช้งานจริง
- ใช้กับ SQL Server 2008/2012
- เนื้อหาละเอียด ชัดเจน
- เนื้อหาครบถ้วนทั้ง VB และ VC#

 Visual Studio

สอบถามปัญหาได้ที่

[www.facebook.com/thaivb.net](http://www.facebook.com/thaivb.net)

ศุภชัย สมพานิช

Thaivb.NET

คู่มือเขียน

# Windows 8 Apps

ด้วย HTML5 และ JavaScript สำหรับผู้เริ่มต้น



ปูพื้นฐานออกแบบส่วนแสดงผลเป็นขั้นตอน  
เนื้อหาละเอียด ชัดเจน และเข้าใจอย่างแท้จริง  
ตัวอย่าง Workshop พร้อมใช้งาน  
ศึกษาและใช้งาน HTML5, CSS3, JavaScript และ jQuery ตั้งแต่เริ่มต้น

สอบถามปัญหาได้ที่ [www.facebook.com/thaivb.net](http://www.facebook.com/thaivb.net)

ศุภชัย สมพานิช

Thaivb.NET

พัฒนาแอปพลิเคชันด้วย  
**WPF 4.5**

ใน VB 2012 และ VC# 2012 สำหรับมือใหม่

**Visual Studio 2012**

ทำความรู้จักกับการพัฒนาแอปพลิเคชันด้วย WPF 4.5  
ศึกษาการปรับแต่งส่วนแสดงผลให้ตรงกับความต้องการ  
การพัฒนาแอปพลิเคชันกับฐานข้อมูล SQL 2008/2012  
ตัวอย่าง Workshop พร้อมใช้งานได้จริง

■ เหมาะกับผู้เริ่มต้น

■ เนื้อหาละเอียด ชัดเจน

■ ใช้กับ SQL Server 2008/2012

■ โค้ดครบถ้วนทั้ง VB และ VC#

 **Visual Studio**

[www.facebook.com/thaivb.net](http://www.facebook.com/thaivb.net)

ศุภชัย สมพานิช

Thaivb.NET

คู่มือเขียน

# Windows 8 Apps

ด้วย VB 2012 & VC# 2012 สำหรับผู้เริ่มต้น



Visual Studio

เนื้อหาละเอียด ชัดเจน และเข้าใจอย่างแท้จริง

ตัวอย่าง Workshop พร้อมใช้งาน

ศึกษาและใช้งานภาษาสคริปต์ XAML ร่วมกับ VB 2012 & VC# 2012

สอบถามปัญหาได้ที่ [www.facebook.com/thaivb.net](http://www.facebook.com/thaivb.net)