Professional Visual Basic 2015

- อ่านเข้าใจง่าย อธิบายอย่างเป็นขั้นตอน มีตัวอย่างประกอบทุกหัวข้อ
- รองรับการพัฒนาทั้ง PC, Mobile และ Web Site
- เหมาะสำหรับนักเรียน นักศึกษา และผู้ที่สนใจพัฒนาแอพพลิเคชัน

ดาวน์โหลดโค้ด
http://www.infopress.co.th/devbook/CodeVB2015.zip

ศุภชัย สมพานิช

IDC
PREMIER

มีเพียง “ความรู้” เท่านั้นที่มนุษย์ใช้พลิก “โลก” และเปลี่ยน “ชีวิต”
เราจึงสร้างสรรค์ และส่งมอบ “ความรู้” ในรูปแบบที่ดีกว่า
เพื่อให้คนไทย “เรียนรู้” ได้ตลอดชีวิต

Think Beyond

Read

Professional Visual Basic 2015

ผู้แต่ง ศุภชัย สมพานิช http://facebook.com/thaivb.net

บรรณาธิการ กิตินันท์ พลสวัสดิ์ kitinan_p@idcpremier.com

ออกแบบปก วสันต์ พึ่งพูลผล, ยุทธนา เกิดประดิษฐ์

ออกแบบและจัดรูปเล่ม วุฒิพันธ์ สมพระเมฆ, สิริลักษณ์ วาระเลิศ

พิสูจน์อักษร มนฤดี ศรีอุทโยภาส

ประสานงานการผลิต สุพัตรา อาจปรุ, ปฐมพล ธรรมศรีสกุล

ข้อมูลทางบรรณานุกรม

ศุภชัย สมพานิช

Professional Visual Basic 2015

นนทบุรี : ไอดีซีฯ, 2559

424 หน้า

1. วิชวลสตูดิโอ (โปรแกรมคอมพิวเตอร์)

2. การเขียนโปรแกรม

I ชื่อเรื่อง

005.133

ISBN 885-916-100-436-3

พิมพ์ครั้งที่ 1 พฤษภาคม 2559

จัดพิมพ์และจัดจำหน่ายโดย

บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 19 ห้อง 1901 อาคารจัสมินอินเตอร์เนชั่นแนลทาวเวอร์

ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120

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

ราคา 350 บาท

สมาชิกสัมพันธ์

โทรศัพท์ 0-2962-1081-3 ต่อ 121

โทรสาร 0-2962-1084

ร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114

โทรสาร 0-2962-1084

บรรณาธิการ

ปัจจุบันโปรแกรมหรือแอพพลิเคชันไม่ได้มีแต่เพียงแค่บนเครื่องคอมพิวเตอร์ตั้งโต๊ะเพียงอย่างเดียว แต่ได้มีการพัฒนาไปยังเครื่องมืออิเล็กทรอนิกส์ต่าง ๆ เป็นจำนวนมาก โดยเฉพาะบนโทรศัพท์เคลื่อนที่ หรือที่เรียกกันว่า สมาร์ทโฟน และแท็บเล็ต

จึงเป็นที่มาของ Visual Studio 2015 ที่ไมโครซอฟท์ตั้งใจพัฒนาขึ้นมาเพื่อรองรับการพัฒนาโปรแกรมเกือบจะทุก ๆ ช่องทาง ทั้งการพัฒนาโปรแกรมสำหรับ PC (หรือ Notebook), การพัฒนาโปรแกรมบนเว็บ (ไม่ขึ้นกับแพลตฟอร์ม), การพัฒนาโปรแกรมสำหรับมือถือและแท็บเล็ต เนื้อหาในหนังสือเล่มนี้จึงเรียบเรียงขึ้นบนแนวความคิดที่ว่า “เมื่อผู้อ่านอ่านหนังสือเล่มนี้ ต้องสามารถพัฒนาโปรแกรมบน Desktop, Web, มือถือ และแท็บเล็ตได้ด้วย Visual Basic 2015” ซึ่งทางสำนักพิมพ์เชื่อว่าผู้อ่านสามารถทำได้จริง เพราะผู้เขียนได้อธิบายทุกขั้นตอนอย่างละเอียด เป็นขั้นเป็นตอน สามารถทำตามได้จริง

ทั้งนี้ ทางสำนักพิมพ์หวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะสามารถช่วยให้ท่านผู้อ่านทุกท่านพัฒนาโปรแกรมได้จริง และหากหนังสือเล่มนี้มีขอผิดพลาดประการใดต้องขออภัยมา ณ โอกาสนี้ด้วย ทางสำนักพิมพ์พร้อมน้อมรับทุกคำติชม เพื่อนำไปใช้ในการปรับปรุงในครั้งต่อ ๆ ไป

กิตินันท์ พลสวัสดิ์
พฤษภาคม 2559

คำนำ

ในปัจจุบันโปรแกรม Visual Studio ถูกพัฒนามาถึงเวอร์ชัน 2015 ประกอบไปด้วยฟีเจอร์ต่างๆ มากมาย นับได้ว่าเป็นเวอร์ชันที่พัฒนาแบบก้าวกระโดดอีกครั้งหนึ่งในโลกของ .NET

“หนังสือ Professional Visual Basic 2015” เล่มนี้เป็นการนำเสนอเนื้อหาพัฒนาโปรแกรมทั้งสิ้น 4 ด้าน คือ Desktop Apps, Database Apps, Web Apps และ Mobile Apps เพื่อให้ตรงกับโลกของความเป็นจริงในช่วงเวลานี้

รูปแบบการนำเสนอใช้วิธีการนำเสนอตั้งแต่ระดับพื้นฐานที่สุด ไล่ระดับไปเรื่อยๆ โดยอาศัยความรู้ของแต่ละหัวข้อสะสมในทุกๆ บท

คุณผู้อ่านท่านใดที่ต้องการสอบถามปัญหา, ติชม หรือให้คำแนะนำ มี 2 ช่องทาง คือ

- แฟนเพจของผู้เขียนที่ https://www.facebook.com/thaivb.net
- ช่อง Youtube ค้นหาช่องที่ชื่อว่า Thaivb.NET

หากเนื้อหาในส่วนใดเกิดข้อผิดพลาด หรือมีการอัพเดตเพิ่มเติม สามารถติดตามได้จากแฟนเพจผู้เขียนข้างต้น

ศุภชัย สมพานิช
พฤษภาคม 2559

Professional Visual Basic 2015

CHAPTER

01

เตรียมความพร้อมก่อนเริ่มต้นเขียนโปรแกรมด้วยภาษา Visual Basic 2015

โลกของการเขียนโปรแกรมไม่ได้มีแต่เพียงแค่พัฒนาโปรแกรมบน PC เพียงอย่างเดียวเท่านั้น แต่ยังรวมไปถึงมือถือและแท็บเล็ตอีกด้วย

ไมโครซอฟต์พัฒนา Visual Studio 2015 (เรียกสั้นๆ ว่า VS 2015) ขึ้นมา เพื่อรองรับการพัฒนาโปรแกรม 3 ช่องทาง คือ

1. พัฒนาโปรแกรมสำหรับ PC (หรือ Notebook)
2. พัฒนาโปรแกรมบนเว็บ (ไม่ขึ้นกับแพลตฟอร์ม)
3. พัฒนาโปรแกรมสำหรับมือถือและแท็บเล็ต

ในปัจจุบันไมโครซอฟต์พัฒนา Windows 10 ขึ้นมา ทำให้โลกของโปรแกรมบน PC, มือถือและแท็บเล็ตกลายเป็นโลกเดียวกัน ส่งผลให้นักพัฒนาไม่ต้องเสียเวลาเรียนรู้ซ้ำซ้อนหลายขั้นตอนอีกต่อไป

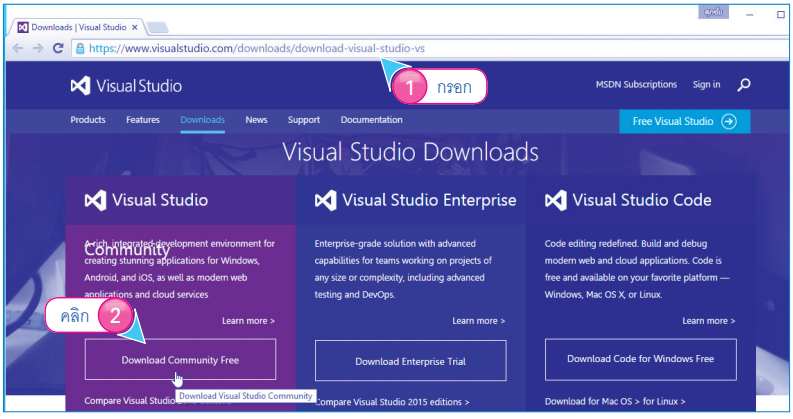
เนื้อหาที่จะนำเสนอในหนังสือเล่มนี้ ครอบคลุมการพัฒนาโปรแกรมบน Desktop, Web, มือถือ และแท็บเล็ต โดยอาศัยภาษา Visual Basic 2015 (เรียกสั้นๆ ว่า VB 2015) ที่มีเนื้อหาสาระเพียงพอที่จะต่อยอดเป็นโปรแกรมเมอร์มืออาชีพในลำดับต่อไปได้ไม่ยาก

การดาวน์โหลดและติดตั้ง Visual Studio 2015

ผู้อ่านสามารถเขียนโปรแกรมในตระกูล .NET ได้โดยการใช้โปรแกรมที่ชื่อว่า Visual Studio ซึ่งมีขั้นตอนดังนี้

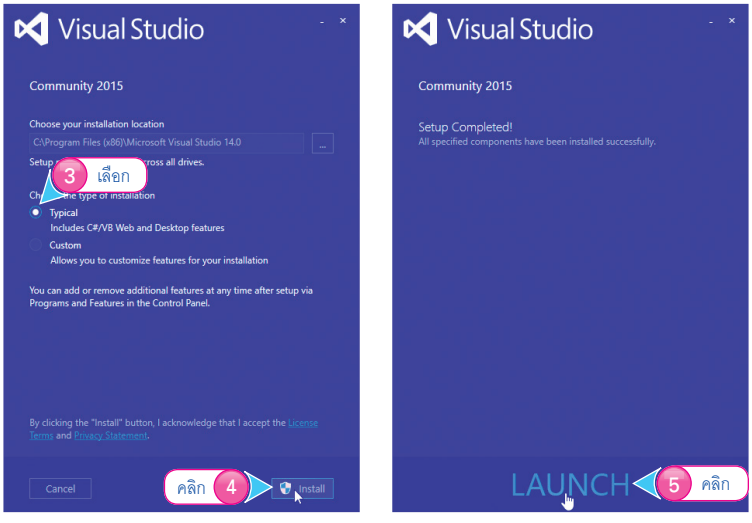
1. ให้ไปที่เว็บไซต์ https://www.visualstudio.com/downloads/download-visual-studio-vs เพื่อดาวน์โหลดโปรแกรม Visual Studio เวอร์ชันล่าสุดเท่าที่มีอยู่ในปัจจุบัน ดังรูปที่ 1-1
2. ให้ผู้อ่านคลิกดาวน์โหลดเวอร์ชัน Community Edition มาใช้งานได้ฟรี

รูปที่ 1-1 แสดงหน้าเว็บเพจที่ให้ดาวน์โหลด Visual Studio



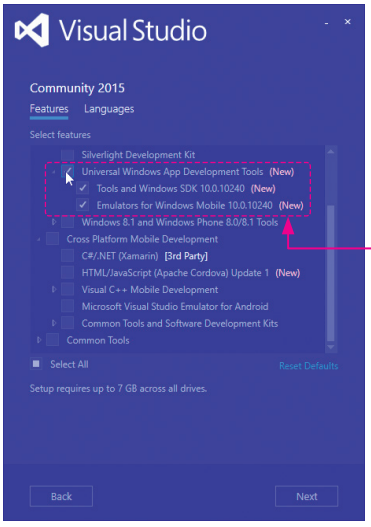
3. การติดตั้ง Visual Studio 2015 ให้เลือกแบบ Typical ซึ่งเป็นการติดตั้งแบบปกติ (ไม่รวมโปรเจ็กต์ประเภท Windows 10 Apps)
4. คลิกปุ่ม Install
5. คลิกปุ่ม LAUNCH

รูปที่ 1-2 แสดงการติดตั้งโปรแกรม Visual Studio 2015 Community Edition



ในกรณีที่ผู้อ่านต้องการพัฒนา Windows 10 Apps ด้วย ให้ติดตั้ง Visual Studio 2015 บน Windows 10 และเลือกติดตั้งแบบ Custom โดยเลือกตัวเลือกดังรูป

รูปที่ 1-3
กรณีติดตั้ง Visual Studio แบบ Custom



จากรูปที่ 1-3 ที่รายการ Universal Windows App Development Tools ให้ผู้อ่านเลือกติดตั้งเพิ่มเติม คือ ตัว Windows 10 SDK กับ Emulator สำหรับรันบนมือถือ Windows 10

รายการที่ต้องติดตั้งเพิ่มเติม

การตั้งค่า Visual Studio 2015 เบื้องต้น

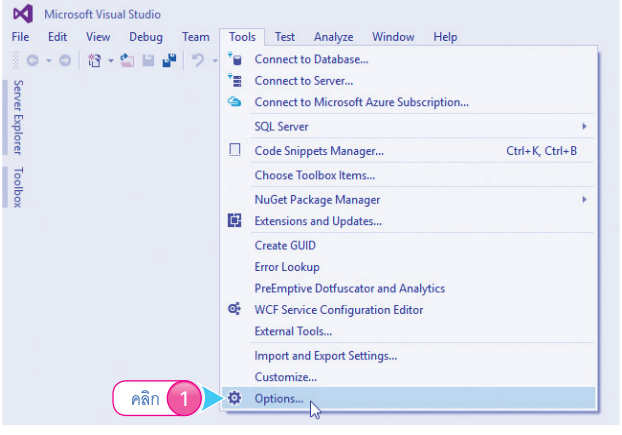
ก่อนเริ่มต้นเขียนเรียนรู้การพัฒนาแอพพลิเคชันด้วย Visual Studio 2015 ผู้เขียนอยากจะแนะนำให้ผู้อ่านรู้จักการตั้งค่าต่าง ๆ ที่ช่วยให้การพัฒนาแอพพลิเคชันต่าง ๆ ง่ายขึ้นดังนี้

การกำหนดให้แสดงไดอะล็อกเลือกโปรเจ็กต์ (New Project Dialog)

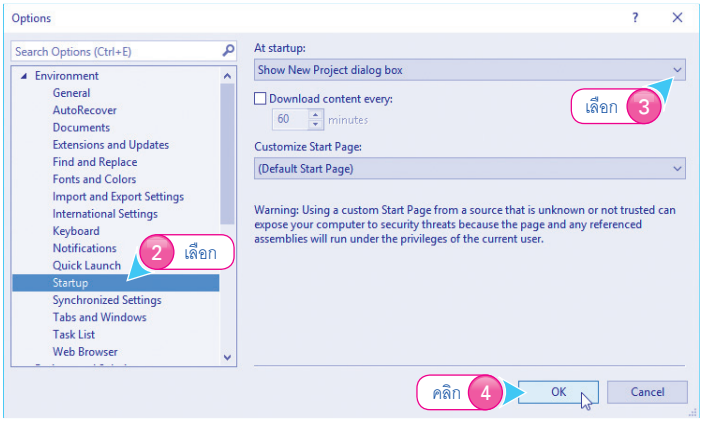
เพื่อเป็นการอำนวยความสะดวกในการสร้างโปรเจ็กต์ใหม่ทุกครั้ง ผู้อ่านสามารถกำหนดให้ VS แสดงไดอะล็อกเลือกสร้างโปรเจ็กต์ทุกครั้ง เมื่อมีการเปิด Visual Studio 2015 ขึ้นมา โดยมีขั้นตอนดังนี้

1. คลิกเมนู Tools > Options...
2. จะปรากฏไดอะล็อก Options ให้คลิกเลือก Environment > Startup
3. คลิกเลือก Show New Project dialog box
4. คลิกปุ่ม OK

รูปที่ 1-4
แสดงการเลือกรายการ Show New Project dialog box

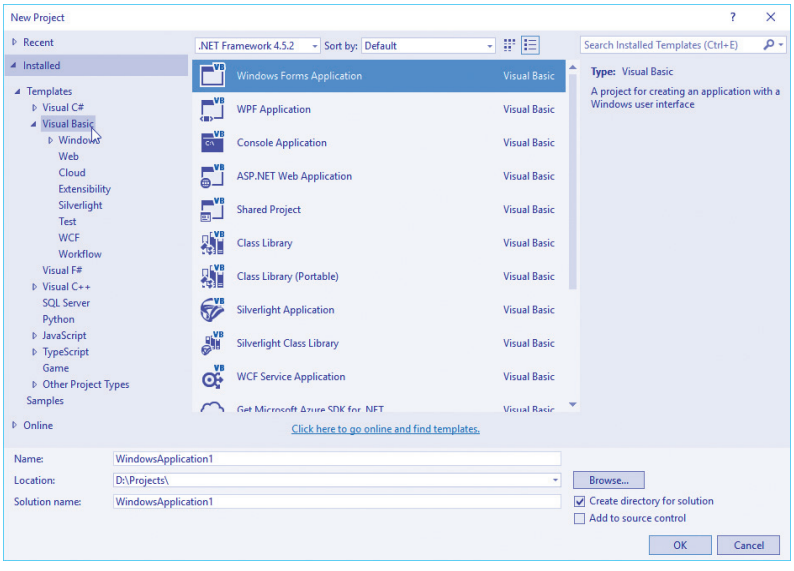


รูปที่ 1-4 (ต่อ)
แสดงการเลือกรายการ Show New Project dialog box



5. จะปรากฏไดอะล็อกสำหรับสร้างโปรเจ็กต์ใหม่ดังรูป

รูปที่ 1-5
แสดงไดอะล็อกเลือกสร้างโปรเจ็กต์ใหม่



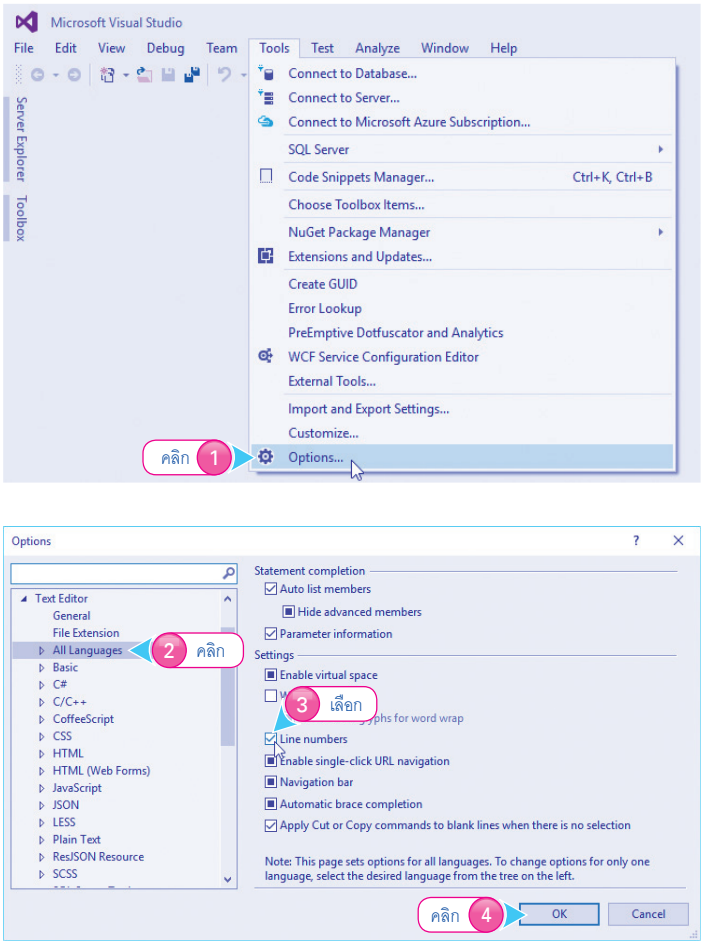
จากรูปที่ 1-5 ผู้อ่านสามารถเลือกสร้างโปรเจ็กต์ใน Visual Studio 2015 ได้ตามที่ต้องการจากไดอะล็อกนี้

การกำหนดให้แสดงหมายเลขกำกับโค้ดในแต่ละบรรทัด

การกำหนดให้ Visual Studio 2015 แสดงหมายเลขกำกับโค้ดในแต่ละบรรทัด ผู้อ่านสามารถทำได้โดยมีขั้นตอนดังนี้

1. คลิกเมนู Tools > Options...
2. จะปรากฏไดอะล็อก Options ให้คลิกเลือก Text Editor > All Languages
3. คลิกเลือก Line numbers
4. คลิกปุ่ม OK

รูปที่ 1-6
แสดงการกำหนดให้แสดงหมายเลขกำกับโค้ดแต่ละบรรทัด

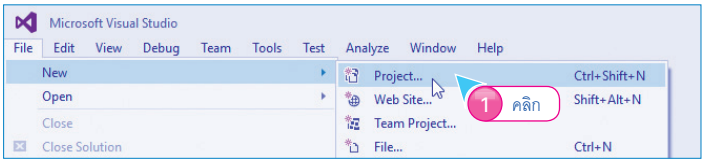


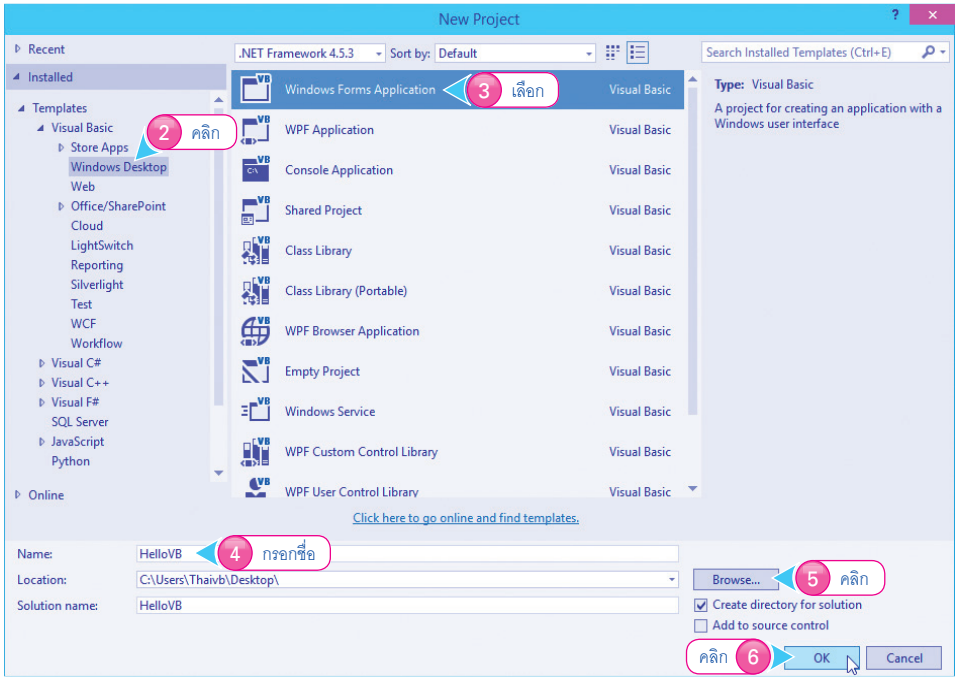
การสร้างโปรเจ็กต์ใหม่

การสร้างโปรเจ็กต์ใหม่ถือเป็นจุดเริ่มต้นของการพัฒนาแอพพลิเคชันต่าง ๆ ในที่นี้จะเริ่มต้นจาก Windows Forms Application ซึ่งเป็นโปรเจ็กต์หลักของ Visual Basic 2015 โดยมีขั้นตอนดังนี้

1. ให้ผู้อ่านคลิกเมนู File > New > Project...
2. คลิกเลือก Visual Basic > Store Apps > Windows Desktop
3. เลือกสร้างโปรเจ็กต์แบบ Windows Forms Application ซึ่งเป็นโปรเจ็กต์ที่ใช้สร้างโปรแกรมสำหรับทำงานบน Windows
4. กรอกชื่อโปรเจ็กต์
5. เลือกตำแหน่งที่อยู่ของโปรเจ็กต์
6. คลิกปุ่ม OK

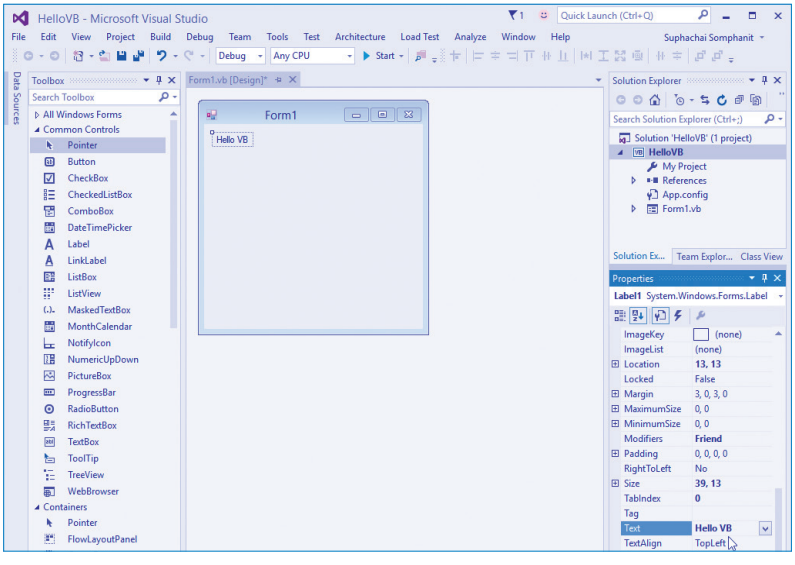
รูปที่ 1-7
แสดงการสร้างโปรเจ็กต์แบบ Windows Forms Application





7. จะปรากฏหน้า Windows Forms Application สำหรับสร้างโปรเจ็กต์ดังรูป

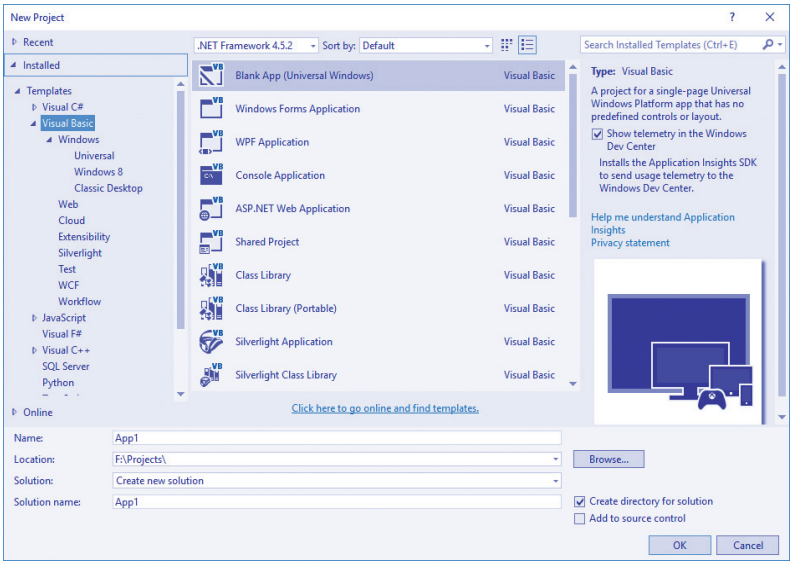
รูปที่ 1-8
แสดงสภาพแวดล้อมของ VS แบบ Windows Forms Application



ทำความรู้จักกับโปรเจ็กต์ประเภทอื่นๆ ของ Visual Studio 2015

นอกเหนือไปจากโปรเจ็กต์ประเภท Windows Forms Application แล้ว ตัวโปรแกรม Visual Studio 2015 ยังรองรับการพัฒนาแอพฯ ประเภทต่างๆ อีกมากมาย

รูปที่ 1-9 แสดงไดอะล็อกสร้างโปรเจ็กต์ประเภทต่างๆ ของ Visual Studio 2015



โปรเจ็กต์ประเภท Windows

โปรเจ็กต์ประเภท Windows ยังถูกแบ่งออกเป็น 3 ประเภท คือ

- Universal เป็นการพัฒนาแอพฯ ที่รันบนระบบปฏิบัติการ Windows 10 โดยสามารถสร้างโปรเจ็กต์ขึ้นมาเพียง 1 โปรเจ็กต์ แต่แอพฯ ที่ได้นำมารันได้ทั้ง PC แท็บเล็ต และมือถือ
- Windows 8 เป็นการพัฒนาแอพฯ ที่รันบนระบบปฏิบัติการ Windows 8/8.1 เท่านั้น
- Classic Desktop เป็นการพัฒนาแอพแบบที่เราคุ้นเคยกันเป็นอย่างดี แอพฯ ที่ได้มาก็คือ ไฟล์ exe สามารถติดตั้งในระบบปฏิบัติการ Windows ต่างๆ ที่มีการติดตั้ง .NET Framework ตามเวอร์ชันที่เราเป็นผู้สร้างโปรเจ็กต์ขึ้นมา

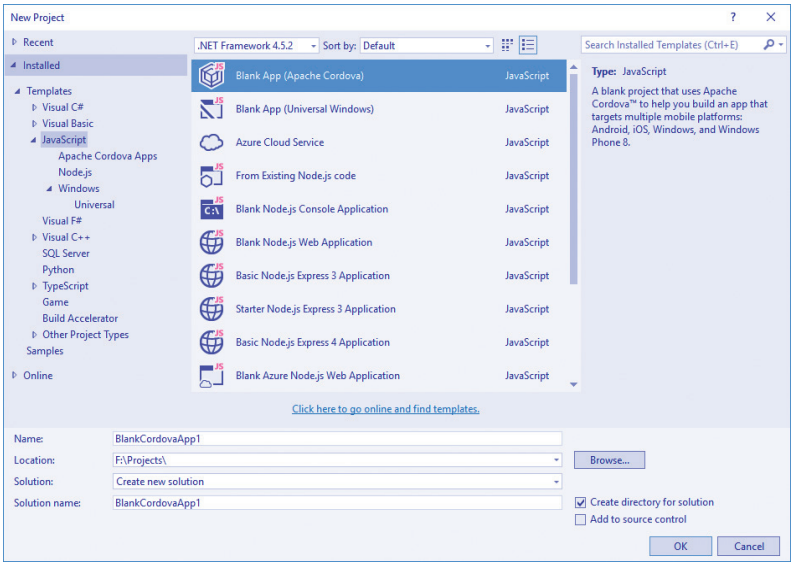
โปรเจ็กต์ประเภท Web

ถ้าต้องการสร้างเว็บไซต์ต่างๆ เช่น เว็บแสดงรูปภาพ, เว็บบอร์ด, เว็บข่าวสาร ฯลฯ แสดงผลในบราวเซอร์ เราถือว่าเป็นโปรเจ็กต์ประเภท Web อยู่ในความรับผิดชอบของภาษา ASP.NET

โปรเจ็กต์ประเภท JavaScript

เดิมทีเดียวภาษา JavaScript ถูกนำมาใช้พัฒนาด้าน Web Application เพียงอย่างเดียวเท่านั้น แต่ในปัจจุบันภาษา JavaScript ถูกนำมาพัฒนาแอพฯ ได้ทั้งแบบ Web Apps กับ Mobile Apps อีกด้วย

รูปที่ 1-10 แสดงโปรเจ็กต์ประเภท JavaScript



จากรูปที่ 1-10 ภาษา JavaScript ใน Visual Studio 2015 ในขั้นต้นสามารถสร้างโปรเจ็กต์ได้ 3 แบบ คือ

- **Apache Cordova Apps** เป็นการใช้งานภาษา JavaScript ร่วมกับภาษา HTML5 เพื่อสร้างแอพฯ ที่สามารถรันบนมือถือ โดยอาศัย **Cordova** (อาจจะเรียกว่า **PhoneGap** ก็ได้)
- **Node.js** เป็นการพัฒนา Web Apps ที่รันได้ทั้งฝั่ง Server/Client โดยอาศัยภาษา JavaScript เพียงภาษาเดียว
- **Windows – Universal** เป็นการพัฒนาแอพฯ ที่รันบน Windows 10 (PC, แท็บเล็ตและมือถือ) โดยอาศัยภาษา HTML5 ร่วมกับภาษา JavaScript

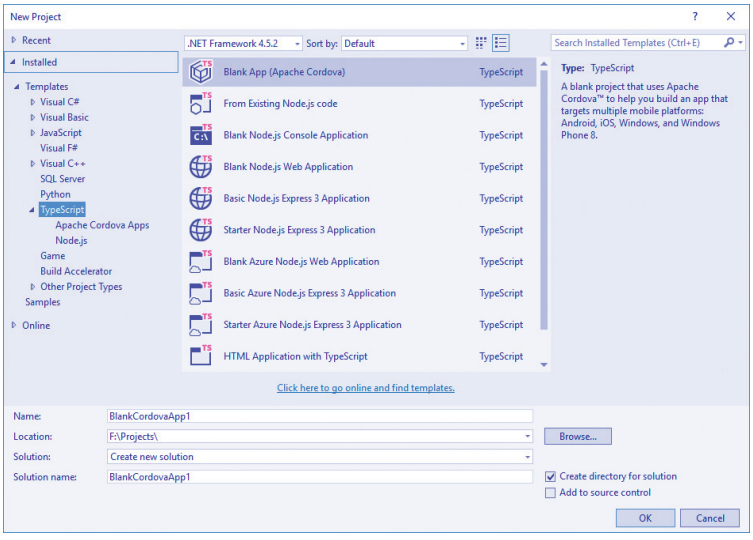
โปรเจ็กต์ประเภท TypeScript

ภาษา TypeScript เป็นภาษาใหม่ที่ไมโครซอฟท์สร้างขึ้นมา โดยมีหลักการที่ว่าทุกสิ่งทุกอย่างของภาษา JavaScript คือ ภาษา TypeScript และเพิ่มเติมความสามารถอื่นๆ ในตัวภาษา TypeScript อีกด้วย เช่น การกำหนดประเภทข้อมูล, การสร้างคลาสขึ้นมาใช้งานเองได้ เป็นต้น ปัจจุบันเป็นภาษาสคริปต์ที่กำลังได้รับความนิยมเป็นอย่างมาก

จากรูปที่ 1-11 ภาษา TypeScript ใน Visual Studio 2015 ในขั้นต้นสามารถสร้างโปรเจ็กต์ได้ 2 แบบ คือ

- **Apache Cordova Apps** เป็นการใช้งานภาษา TypeScript ร่วมกับภาษา HTML5 เพื่อสร้างแอพฯ ที่สามารถรันบนมือถือ โดยอาศัย Cordova (อาจจะเรียกว่า PhoneGap ก็ได้)
- **Node.js** เป็นการพัฒนา Web Apps ที่รันได้ทั้งฝั่ง Server/Client โดยอาศัยภาษา TypeScript เพียงภาษาเดียว

รูปที่ 1-11
แสดงโปรเจ็กต์ประเภท TypeScript



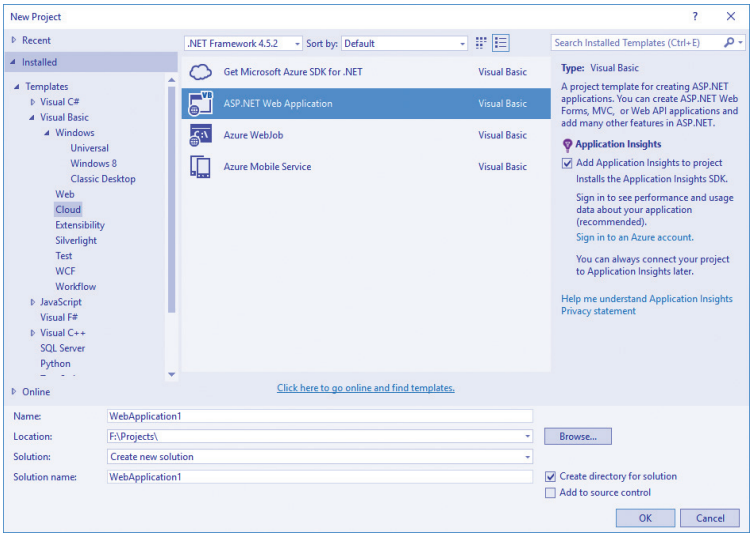
โปรเจ็กต์ประเภท Cloud

ในปัจจุบันการขอใช้บริการต่างๆ บนก้อนเมฆ (Cloud) ได้รับความนิยมเป็นอย่างสูง และบริการ Cloud ของไมโครซอฟท์ชื่อว่า Azure เราสามารถสร้างโปรเจ็กต์ได้ 2 รูปแบบใหญ่ๆ คือ

- **โปรเจ็กต์ประเภท ASP.NET Web Application** เป็นการพัฒนา Web Apps ด้วยภาษา ASP.NET เพื่อขอใช้บริการต่างๆ ใน Azure
- **โปรเจ็กต์ประเภท Azure Mobile Service** เป็นการสร้างบริการที่เชื่อมต่อกับ Azure ให้บริการด้านต่างๆ โดยอาศัยโครงสร้างของ Web API สำหรับอุปกรณ์ประเภท Mobile Devices ต่างๆ

ประเภทโปรเจ็กต์ที่ผู้เขียนนำเสนอรายละเอียดในขั้นต้นนี้ เป็นการติดตั้งเพิ่มเติมแยกต่างหาก เพื่อแสดงให้เห็นความสามารถของตัวโปรแกรม Visual Studio 2015 ที่มีอยู่มากมาย

รูปที่ 1-12
แสดงโปรเจ็กต์ประเภท Cloud

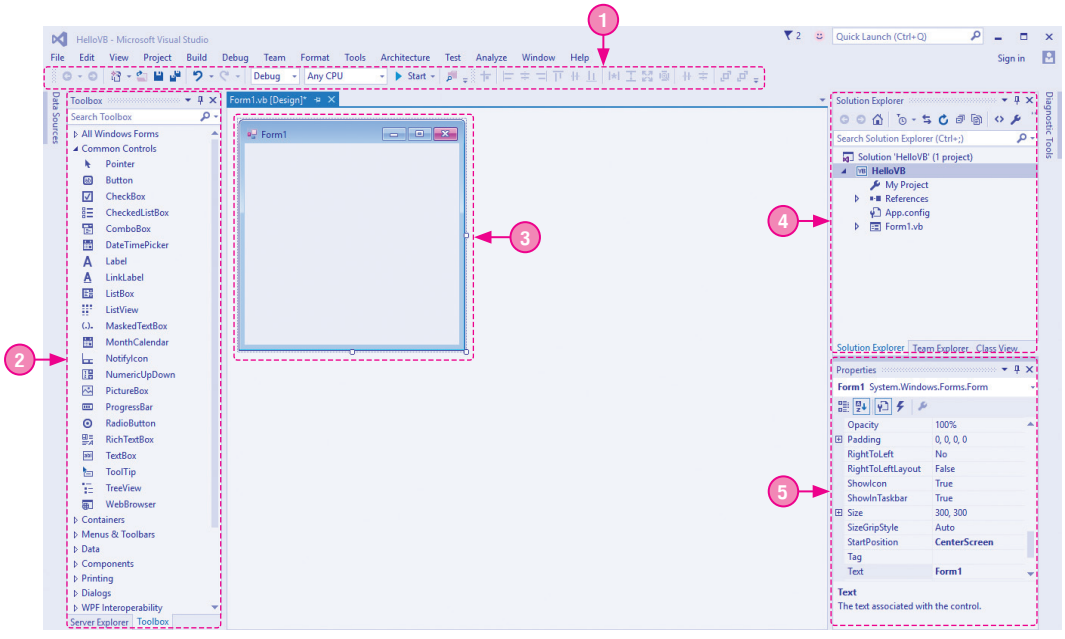


ทำความรู้จักกับสภาพแวดล้อม (IDE) ของ VS ในขั้นต้น

VS 2015 รองรับการพัฒนาแอพพลิเคชันหลายแพลตฟอร์ม (PC, มือถือ และแท็บเล็ต) ผู้อ่านสามารถเลือกสร้างโปรเจ็กต์ได้ตามต้องการ และสภาพแวดล้อมของโปรเจ็กต์แต่ละประเภทจะมีลักษณะคล้ายกัน เพื่อให้นักพัฒนาคุ้นเคยนั่นเอง

รูปที่ 1-13
แสดงสภาพแวดล้อม IDE ของ VS แบบ Windows Forms Application

ผู้เขียนขอยกตัวอย่างสภาพแวดล้อมของโปรเจ็กต์ประเภท Windows Forms Application มาอธิบาย ดังรูปที่ 1-13



- หมายเลข 1 แถบเครื่องมือสำหรับทดสอบโปรเจ็กต์ปัจจุบัน
- หมายเลข 2 แถบคอนโทรลสำเร็จรูปทำหน้าที่ออกแบบส่วนแสดงผล (User Interface เรียกย่อ ๆ ว่า UI) ภายในโปรเจ็กต์ของผู้อ่าน
- หมายเลข 3 ส่วนแสดงผลจำลอง ผู้อ่านสามารถออกแบบส่วนแสดงผลให้กับหน้าจอต่าง ๆ ได้ที่นี่ โดยการลากคอนโทรลต่าง ๆ มาวาดตามที่ต้องการ
- หมายเลข 4 เรียกว่า หน้าต่าง Solution Explorer ทำหน้าที่แสดงโครงสร้างโปรเจ็กต์ปัจจุบันของผู้อ่านว่า ประกอบไปด้วยไฟล์อะไรบ้าง
- หมายเลข 5 เรียกว่า หน้าต่างคุณสมบัติ (Properties) ทำหน้าที่แสดงคุณสมบัติต่าง ๆ ของคอนโทรลที่กำลังถูกโฟกัสในส่วนแสดงผลจำลองว่ามีสมบัติอะไรบ้าง สามารถแก้ไขคุณสมบัติต่าง ๆ ได้ตามที่ต้องการเช่นกัน เช่น ตั้งชื่อคอนโทรลได้ที่คุณสมบัติ Name, กำหนดสีพื้นหลังได้ที่คุณสมบัติ BackColor เป็นต้น

มือใหม่ต้องทราบก่อนเขียนโปรแกรมในโลกของ .NET

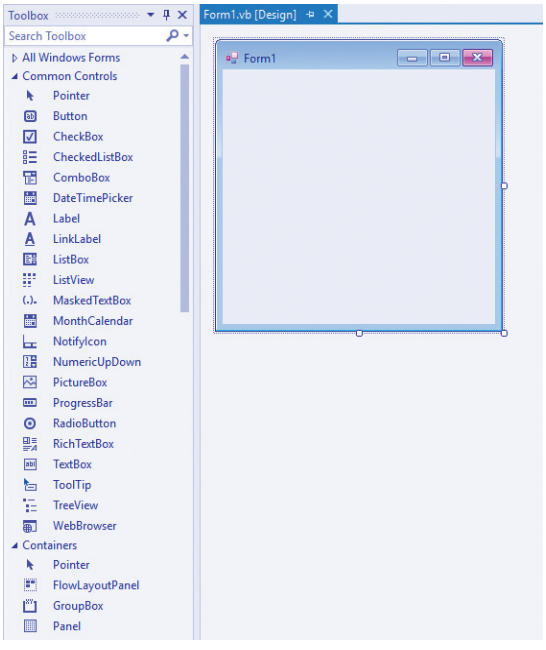
ก่อนที่จะเข้าสู่รายละเอียดของการเขียนโปรแกรมในโลกของ .NET Frawework ผู้เขียนจะขอนำเสนอหลักการพื้นฐานในภาพกว้างที่มือใหม่ต้องทราบในขั้นต้นเสียก่อน เพื่อปูพื้นฐานก่อนที่จะเข้าสู่รายละเอียดเนื้อหาส่วนต่าง ๆ ในแต่ละบทต่อไป

การออกแบบส่วนแสดงผลด้วยคอนโทรล (User Interface)

ไม่ว่าจะสร้างโปรเจ็กต์ประเภทใดก็ตามใน Visual Studio 2015 ไมโครซอฟท์กำหนดให้สภาพแวดล้อมของตัวโปรแกรม Visual Studio 2015 มีลักษณะใกล้เคียงกันมากที่สุดเท่าที่จะเป็นไปได้ เพื่อให้นักพัฒนาคุ้นเคยกับตัวโปรแกรมได้ไม่ยาก

แม้ว่าภาพประกอบผู้เขียนจะใช้โปรเจ็กต์ประเภท Windows Forms Application ก็จริงอยู่ แต่สามารถเทียบเคียงได้กับโปรเจ็กต์ประเภทอื่น ๆ ได้

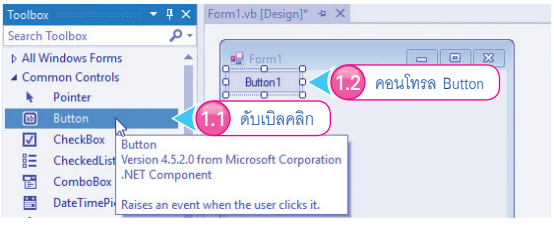
ก่อนอื่นขอให้ผู้อ่านสร้างโปรเจ็กต์แบบ Windows Forms Application ว่าง ๆ ขึ้นมา 1 โปรเจ็กต์ ดังรูปที่ 1-14



รูปที่ 1-14 แสดงแถบ Toolbox

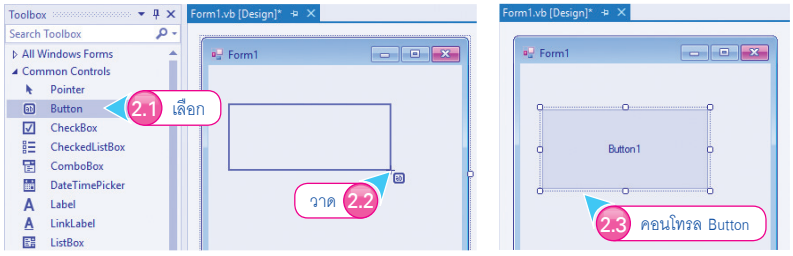
จากรูปที่ 1-14 แถบคอนโทรล (Control) อยู่ด้านซ้ายมือ ทำหน้าที่ออกแบบส่วนแสดงผล (User Interface เรียกย่อ ๆ ว่า UI) ในขั้นต้นนี้ผู้อ่านสามารถใช้งานคอนโทรลนี้ได้ 2 วิธี

1. ดับเบิลคลิกที่ตัวคอนโทรลเพื่อกำหนดให้ Visual Studio ใช้งานคอนโทรลดังกล่าว ในหน้าจอออกแบบโดยอัตโนมัติ เช่น ผู้เขียนต้องการใช้ปุ่มกด Button1 ดังรูปที่ 1-15



รูปที่ 1-15 แสดงการใช้ปุ่มกดด้วยวิธีการดับเบิลคลิกในแถบ Toolbox

2. คลิกเลือกคอนโทรลที่ต้องการใช้งานแล้วนำไปวาดในส่วนของฟอร์ม (Form) ดังรูปที่ 1-16

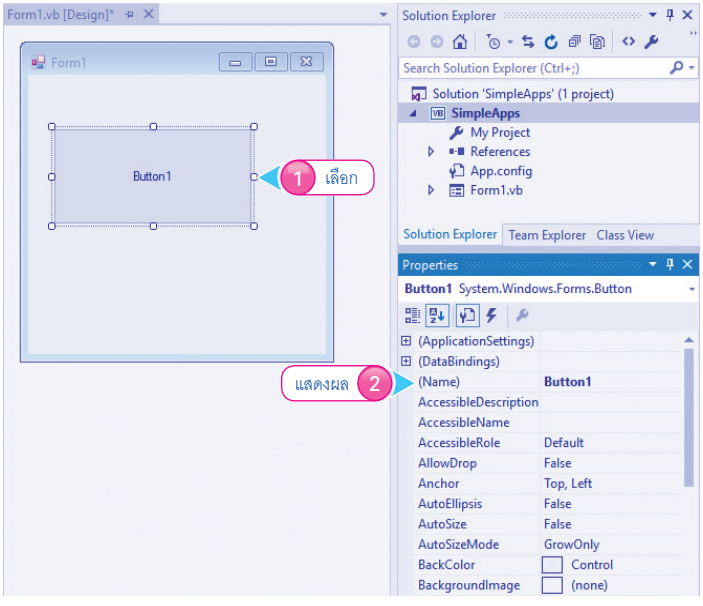


รูปที่ 1-16 แสดงการใช้ปุ่มกดด้วยวิธีวาด

จากรูปที่ 1-16 เราได้ส่วนแสดงผลง่าย ๆ มาแล้ว 1 หน้าจอ โดยมีปุ่มกด Button 1 ปุ่ม

การปรับแต่งคอนโทรลด้วยวิธีแก้ไขคุณสมบัติ (Property)

คอนโทรลต่าง ๆ ที่นำมาใช้งาน สามารถปรับแต่งหรือแก้ไขได้ในหน้าต่างคุณสมบัติ (Properties) กล่าวคือ เมื่อโฟกัสที่คอนโทรลตัวใดก็ตาม หน้าต่างคุณสมบัติจะเปลี่ยนไปโดยอัตโนมัติ



รูปที่ 1-17 แสดงรายการคุณสมบัติของปุ่มกด Button

จากรูปที่ 1-17 ส่วนแสดงผลปัจจุบันของผู้เขียนมีเพียง 1 ปุ่ม เมื่อถูกโฟกัสแล้วหน้าต่างคุณสมบัติก็จะแสดงรายการคุณสมบัติต่าง ๆ ของคอนโทรล Button ให้เราโดยอัตโนมัติ

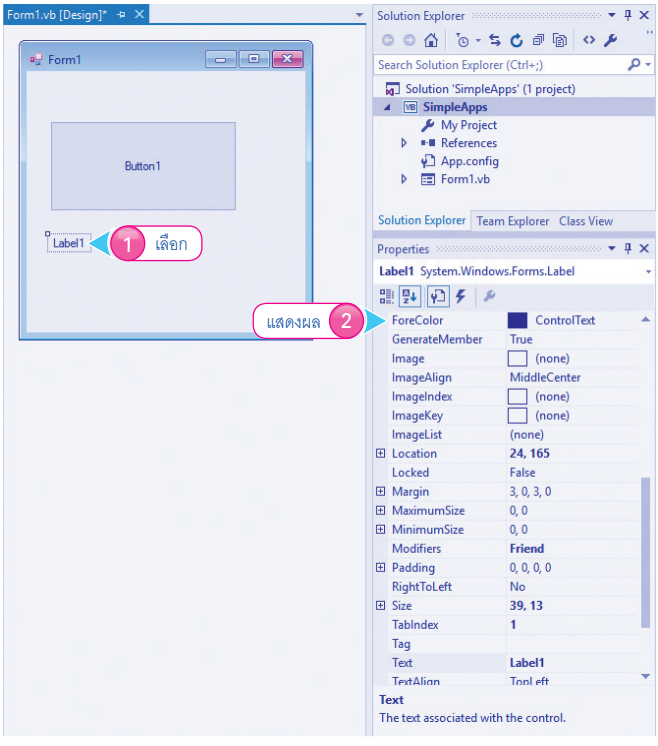
รายการคุณสมบัติของคอนโทรล Button ที่น่าสนใจ แสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
Name	ทำหน้าที่ตั้งชื่อให้กับตัวคอนโทรล ใช้สำหรับอ้างอิงตอนเขียนโค้ด
Text	กำหนดข้อความที่จะแสดงในปุ่มกด
BackColor	กำหนดสีพื้นหลังให้กับปุ่มกด
ForeColor	กำหนดสีตัวอักษรที่อยู่ในปุ่มกด

คุณสมบัติใดก็ตามที่ถูกแก้ไขจะแสดงด้วยตัวอักษรตัวหนา ส่วนตัวอักษรปกติ หมายถึง เป็นค่าเริ่มต้นที่โปรแกรม Visual Studio ตั้งไว้ ในการใช้งานคอนโทรลไม่จำเป็นต้องไปแก้ไขให้ครบทุกคุณสมบัติ โดยแก้ไขเฉพาะเท่าที่ต้องการเปลี่ยนแปลงเท่านั้น

ต่อมา ผู้เขียนลองใช้คอนโทรล Label ซึ่งทำหน้าที่แสดงข้อความ เมื่อคอนโทรล Label ได้รับโฟกัส พบว่าที่หน้าต่างคุณสมบัติก็จะแสดงรายการคุณสมบัติของคอนโทรล Label ให้เราโดยอัตโนมัติ

รูปที่ 1-18 แสดงรายการคุณสมบัติของโทรล Label



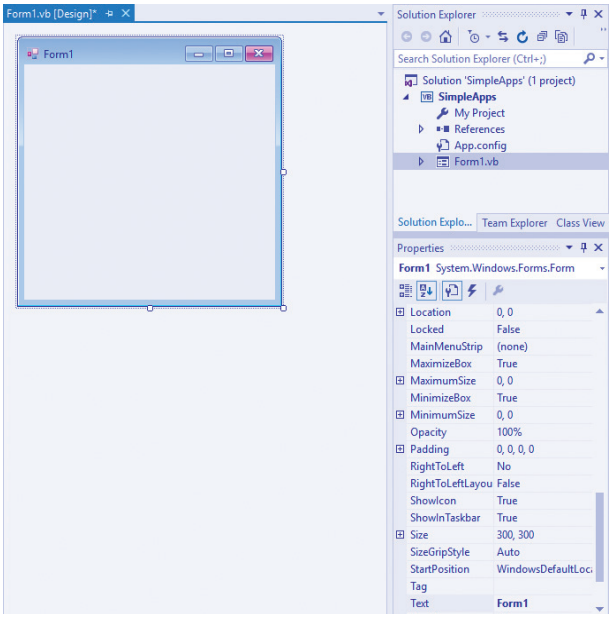
รายการคุณสมบัติที่น่าสนใจแสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
Name	ทำหน้าที่ตั้งชื่อให้กับตัวคอนโทรลใช้สำหรับอ้างอิงตอนเขียนโค้ด
Text	กำหนดข้อความที่จะแสดงใน Label
BackColor	กำหนดสีพื้นหลังให้กับคอนโทรล Label
ForeColor	กำหนดสีตัวอักษรที่อยู่ในคอนโทรล Label
AutoSize	กำหนดขนาดของคอนโทรล Label มี 2 แบบ คือ • True หมายถึง กำหนดให้ Label มีขนาดเท่าที่ข้อความแสดงอยู่ • False หมายถึง คุณสามารถกำหนดขนาดของ Label ได้อย่างอิสระ

คอนโทรลแต่ละตัวก็จะมีรายการคุณสมบัติทั้งที่เหมือนกันและต่างกันไปตามหน้าที่ของตัวมันเอง

ทำความรู้จักกับฟอร์ม (Form)

ไม่ว่าจะสร้างโปรเจ็กต์ประเภทใดก็ตาม โดยส่วนใหญ่ (โปรเจ็กต์บางประเภทไม่ต้องมีส่วนแสดงผลเริ่มต้น) จะมีส่วนแสดงผลเริ่มต้นมาให้ใช้ออกแบบหน้าจอแรก เช่น ในกรณีนี้เราสร้างโปรเจ็กต์แบบ Windows Forms Application ก็จะได้ฟอร์ม (Form) ที่มีชื่อว่า Form1 เป็นหน้าจอเริ่มต้น ดังรูปที่ 1-19

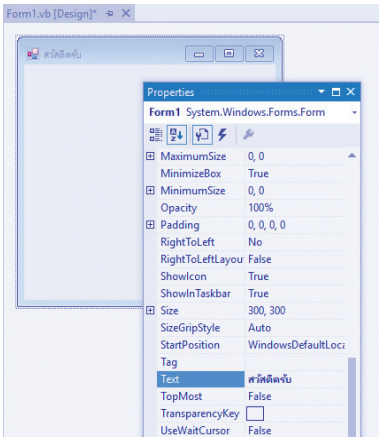


รูปที่ 1-19 แสดงรายการคุณสมบัติของฟอร์ม

จากรูปที่ 1-19 เมื่อโฟกัสที่ Form1 หน้าต่างคุณสมบัติก็จะแสดงรายการคุณสมบัติของ Form1 เช่นกัน โดยคุณสมบัติที่น่าสนใจก็คือ

คุณสมบัติ	หน้าที่
Name	ทำหน้าที่ตั้งชื่อให้กับฟอร์มใช้สำหรับอ้างอิงตอนเขียนโค้ด
Text	กำหนดข้อความที่จะแสดงในแถบ Title Bar
BackColor	กำหนดสีพื้นหลังให้กับฟอร์ม
ForeColor	กำหนดสีตัวอักษรที่อยู่ในฟอร์ม

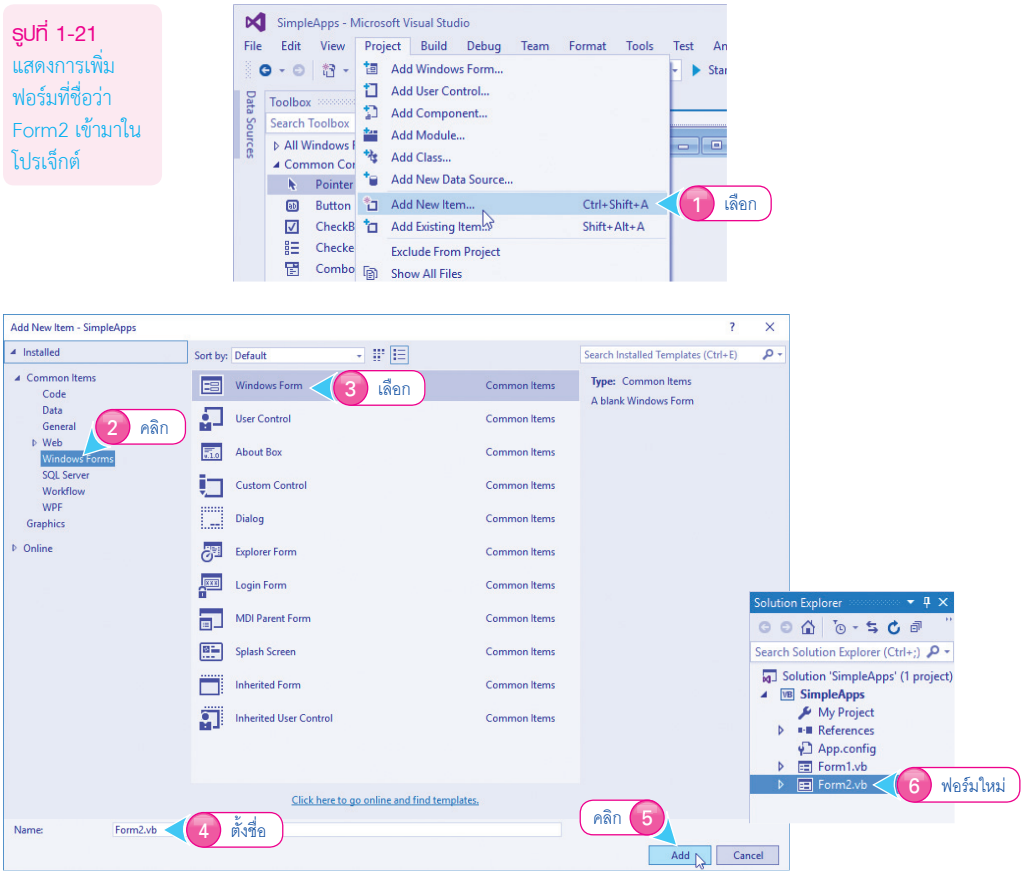
รูปที่ 1-20 แสดงการแก้ไขคุณสมบัติ Text ของ Form1



การเพิ่มฟอร์มใหม่เข้ามาในโปรเจ็กต์

ในการพัฒนาแอพฯ ขึ้นมาใช้งาน โดยส่วนใหญ่จะประกอบไปด้วยส่วนแสดงผลตั้งแต่ 2 หน้าจอขึ้นไป วิธีการเพิ่มหน้าจอใหม่เข้ามา ให้ผู้อ่านคลิกเมนู Project > Add New Item... ในกรณีนี้เราต้องการเพิ่มหน้าจอใหม่ประเภท Windows Forms Application โดยตั้งชื่อว่า Form2 ดังรูปที่ 1-21

รูปที่ 1-21 แสดงการเพิ่มฟอร์มที่ชื่อว่า Form2 เข้ามาในโปรเจ็กต์



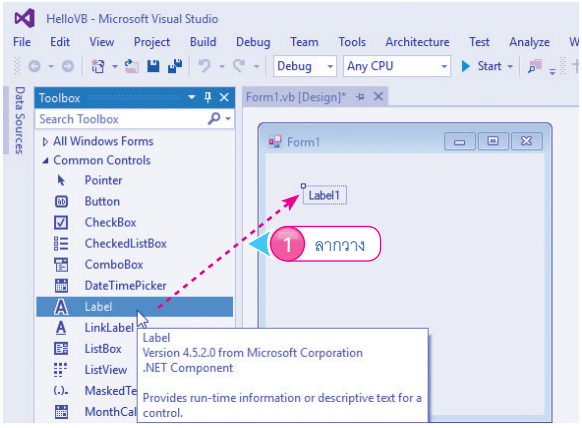
จากรูปที่ 1-21 ณ จุดนี้ โปรเจ็กต์ของเรามี 2 ฟอร์มแล้ว

เริ่มต้นสร้างแอพพลิเคชันแรก

ในหัวข้อนี้ผู้อ่านจะได้ใช้งาน Visual Studio 2015 เขียนโปรแกรมด้วยภาษา Visual Basic 2015 เป็นครั้งแรก โดยเป้าหมายของแอพพลิเคชันนี้ก็คือ แสดงข้อความ “Hello VB” โดยมีขั้นตอนดังนี้

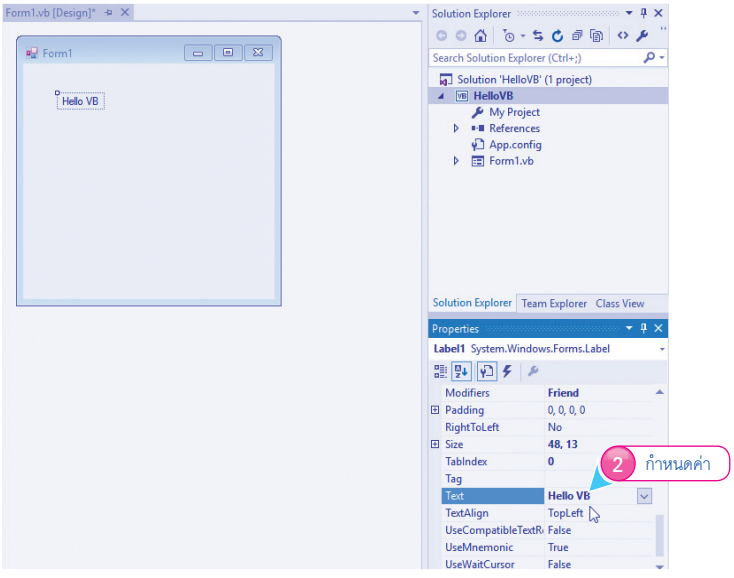
1. ให้ผู้อ่านเลือกใช้คอนโทรล Label เข้ามาทำหน้าที่แสดงข้อความ “Hello VB” โดยการลาก & วางใน Form1 ซึ่งทำหน้าที่เป็นส่วนแสดงผล ดังรูปที่ 1-22

รูปที่ 1-22 แสดงการใช้งานคอนโทรล Label ใน Form1



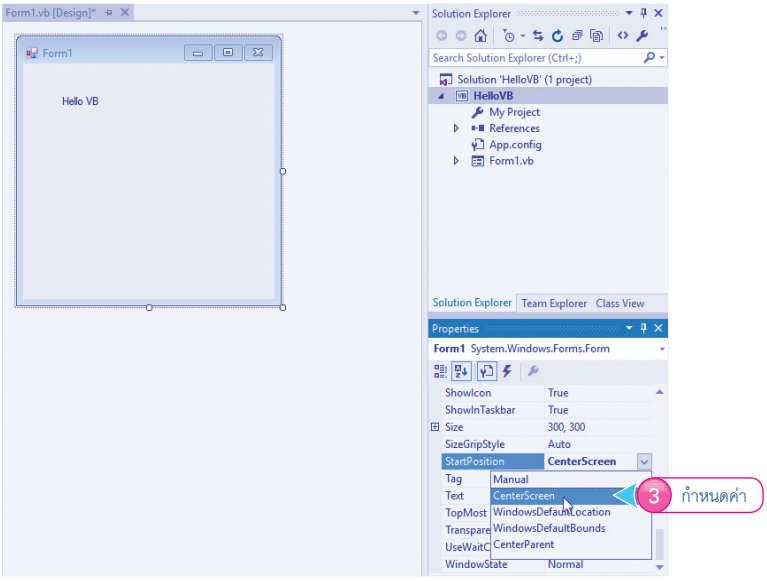
2. กำหนดข้อความใน Label ตัวนี้ ให้ผู้อ่านกำหนดที่คุณสมบัติ Text ในหน้าต่าง Properties เป็นข้อความตามที่ต้องการ

รูปที่ 1-23 แสดงการแก้ไขคุณสมบัติ Text ของคอนโทรล Label ตัวนี้



3. กำหนดให้ Form1 ปรากฏอยู่กึ่งกลางหน้าจอ โดยโฟกัสที่ Form1 ก่อน (คลิกเลือกที่ Form1) ในหน้าต่าง Properties ที่คุณสมบัติ StartPosition ซึ่งทำหน้าที่กำหนดตำแหน่งของ Form1 ให้ผู้เขียนเลือกเป็นแบบ CenterScreen คือ เมื่อโปรแกรมทำงานกำหนดให้ Form1 ปรากฏขึ้นมากึ่งกลางหน้าจอ ดังรูปที่ 1-24

รูปที่ 1-24
แสดงการแก้ไขตำแหน่งของฟอร์ม ให้อยู่กึ่งกลางหน้าจอ

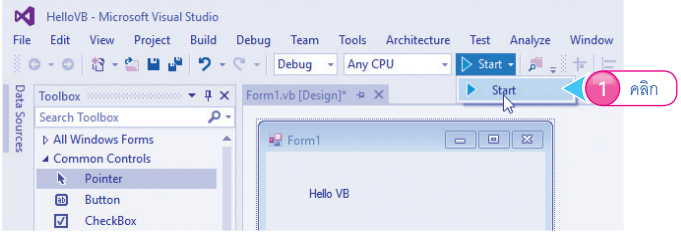


จากรูปที่ 1-24 ณ จุดนี้ เราได้โปรเจ็กต์ตามที่เราต้องการแล้ว

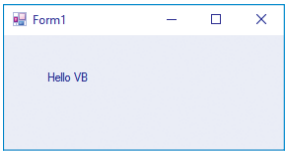
การทดสอบโปรเจ็กต์

หลังจากที่ผู้อ่านออกแบบและเขียนโค้ดแล้ว ก็จะเข้าสู่ขั้นตอนการทดสอบโปรเจ็กต์เพื่อดูแอพพลิเคชันที่สร้างขึ้นมีผลการทำงานเป็นไปตามที่เราต้องการหรือไม่ โดยการคลิกปุ่ม Start เพื่อเริ่มต้นทดสอบรันโปรเจ็กต์

รูปที่ 1-25
แสดงการเริ่มทดสอบโปรเจ็กต์



รูปที่ 1-26
ผลการทำงานของโปรเจ็กต์ปัจจุบัน



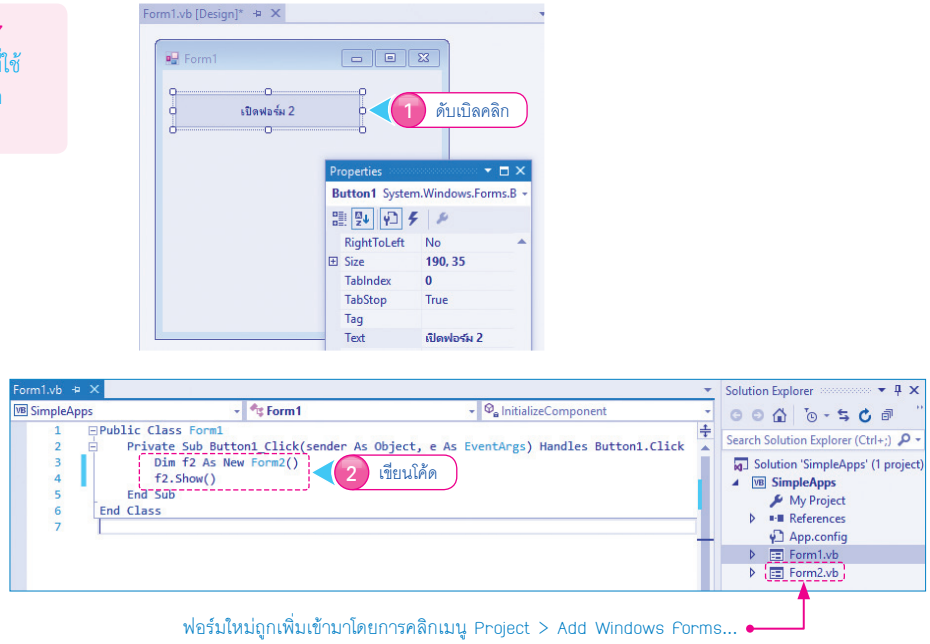
จากรูปที่ 1-26 เห็นได้ว่า เราได้ข้อความ “Hello VB” ปรากฏขึ้นมาใน Form1 ตามที่เราต้องการแล้ว

วิธีการสั่งให้เปิดฟอร์มใหม่โดยการเขียนโค้ด

สำหรับวิธีการสั่งให้เปิดฟอร์มใหม่ที่เราเพิ่มเข้ามามีขั้นตอนดังนี้

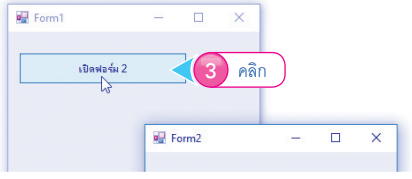
1. ดับเบิลคลิกปุ่ม Button1 เพื่อสร้างเหตุการณ์ Click()
2. เขียนโค้ด ดังรูปที่ 1-27

รูปที่ 1-27 แสดงโค้ดที่ใช้สำหรับเปิด Form2



3. ทดสอบรันโปรเจ็กต์ พบว่า เราสามารถเปิด Form2 ได้แล้ว ดังรูปที่ 1-28

รูปที่ 1-28 แสดงโค้ดที่ใช้สำหรับเปิด Form2



หลักการทำงานของภาษา VB

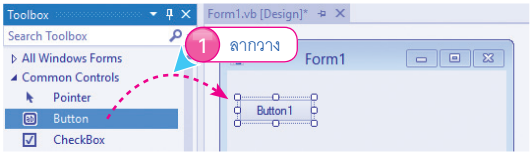
หลักการทำงานของภาษา VB เรียกว่า **การเขียนโปรแกรมตอบสนองเหตุการณ์ที่เกิดขึ้น (Event Driven Programming)** กล่าวคือ ผู้อ่านเป็นผู้กำหนดเองว่า เมื่อเกิดเหตุการณ์ Click() ให้ทำอะไร, เมื่อเกิดเหตุการณ์ Load() ให้ทำอะไร เป็นต้น

หมายความว่า แท้ที่จริงแล้วการกระทำต่าง ๆ ที่ผู้ใช้งานกระทำต่อโปรแกรมของผู้อ่าน ผู้อ่านเป็นผู้กำหนดขึ้นมาเองทั้งสิ้นว่า “ให้ทำอะไร” ประเด็นที่ตามมา ก็คือ วิธีการการสร้างเหตุการณ์ให้กับคอนโทรลแต่ละตัวทำอย่างไร

สมมติว่า ผู้เขียนต้องการสร้างเหตุการณ์ Click() ให้กับปุ่มกด Button จะมีวิธีการดังนี้

1. ให้ผู้อ่านลากปุ่มกด Button มาวางใน Form1 ดังรูปที่ 1-29

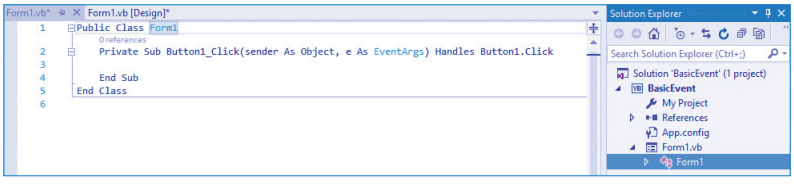
รูปที่ 1-29 แสดงการใช้งานคอนโทรล Button ใน Form1



โดยการสร้างเหตุการณ์ให้กับปุ่มกด Button1 นี้ มี 2 วิธี ดังข้อที่ 2 กับ 3

2. ดับเบิลคลิกที่คอนโทรล Button1 เพื่อสั่งให้ VS สร้างเหตุการณ์ประจำตัวของคอนโทรล Button ขึ้นมา พบว่า เหตุการณ์ประจำตัวของคอนโทรล Button ก็คือ เหตุการณ์ Click() เพราะคอนโทรล Button ถูกสร้างขึ้นมาเพื่อทำหน้าที่รองรับการคลิกจากผู้ใช้งาน

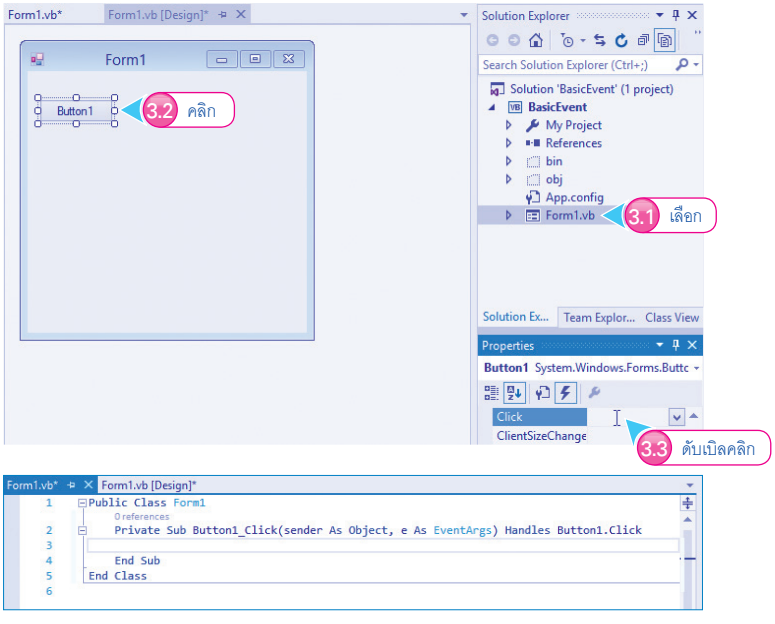
รูปที่ 1-30 แสดงการสร้างเหตุการณ์ Click() ด้วยวิธีดับเบิลคลิก



แต่ละคอนโทรลมีเหตุการณ์ประจำตัวแตกต่างกัน ขึ้นอยู่กับว่าคอนโทรลตัวนั้นๆ ถูกสร้างขึ้นมาเพื่อ “ทำหน้าที่อะไร”

3. อีกวิธีหนึ่ง ก็คือ ให้ผู้อ่านโฟกัสที่คอนโทรล Button1 ก่อน จากนั้นคลิกปุ่ม ⚡ ในหน้าต่าง Properties เพื่อเลือกสร้างเหตุการณ์ที่เราต้องการ จะเห็นได้ว่า ปุ่มกด Button สามารถรองรับได้หลายเหตุการณ์ ให้ผู้อ่านดับเบิลคลิกที่ Click เพื่อสร้างเหตุการณ์ Click() ขึ้นมาสำหรับปุ่มกด Button1 ปุ่มนี้

รูปที่ 1-31 แสดงเหตุการณ์ Click() ที่ได้จากวิธีเลือกสร้างเหตุการณ์



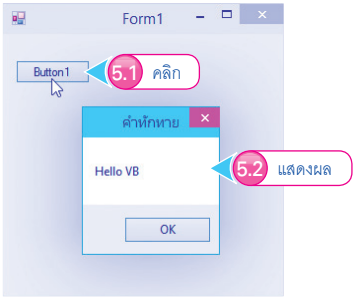
4. ผู้เขียนลองตั้งโจทย์ง่ายๆ ว่า ต้องการแสดงข้อความ “Hello VB” เมื่อผู้ใช้งานคลิกปุ่ม Button1 ซึ่งสามารถทำได้โดยการเขียนโค้ดดังต่อไปนี้

Form1.vb

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        MessageBox.Show("Hello VB", "คำทักทาย")
    End Sub
End Class
```

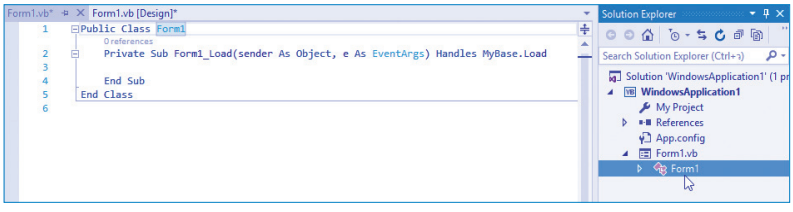
5. จากโค้ดข้างต้นเป็นการสั่งให้แสดงข้อความ “Hello VB” โดยอาศัยคลาส MessageBox เข้ามาทำหน้าที่สร้างไดอะล็อกแสดงข้อความ เมื่อรันโปรแกรมและมีการคลิกปุ่ม Button1 ดังรูปที่ 1-32

รูปที่ 1-32
แสดงข้อความที่ได้



6. ในกรณีที่ผู้อ่านดับเบิลคลิกบริเวณพื้นที่ว่างๆ ของ Form1 ส่งผลให้ VB สร้างเหตุการณ์ Load() ขึ้นมาเพื่อให้กำหนดว่า เมื่อ Form1 ถูกโหลดขึ้นมาแล้วให้ทำอะไร เพราะเหตุการณ์ประจำตัวของ Form ก็คือ เหตุการณ์ Load()

รูปที่ 1-33
แสดงที่จัดเก็บโค้ดภาษา VB



รูปที่ 1-33 ที่หน้าต่าง Solution Explorer ทำหน้าที่แสดงโครงสร้างของโปรเจ็กต์ปัจจุบัน ให้ผู้อ่านคลิกที่รายการ Form1 (.vb) เพื่อแสดงโค้ดทั้งหมดที่อยู่ใน Form1

สรุปท้ายบท

เนื้อหาในบทนี้เป็นเพียงขั้นตอนการเตรียมและทดสอบสภาพแวดล้อมให้เครื่องของผู้อ่านพร้อมใช้งานเท่านั้น เนื้อหาในบทต่อไปจะเข้าสู่โลกของการเขียนโปรแกรมอย่างแท้จริง

CHAPTER

02

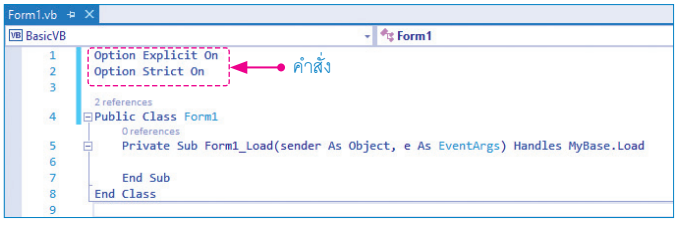
พื้นฐานการเขียนโปรแกรมด้วยภาษา Visual Basic

เนื้อหาในบทนี้จะเข้าสู่โลกของการเขียนโปรแกรมกับภาษา VB 2015 ด้วย Visual Studio 2015 ผู้อ่านจะได้ศึกษาพื้นฐานและไวยากรณ์ทางภาษาที่ต้องทราบเป็นลำดับแรกก่อนที่จะเข้าสู่หัวข้อต่อ ๆ ไป

คำสั่ง Option Explicit On กับ Option Strict On

ทุกครั้งที่ผู้อ่านเริ่มเขียนโค้ดของภาษา VB ควรกำหนดให้ตัว Editor ของ VS ใช้คำสั่ง Option Explicit On และคำสั่ง Option Strict On บริเวณตำแหน่งบนสุดเสมอ ดังรูปที่ 2-1

รูปที่ 2-1 แสดงตำแหน่งการเขียนคำสั่งทั้ง 2



โดยความหมายของคำสั่งทั้ง 2 ข้างต้น มีดังนี้

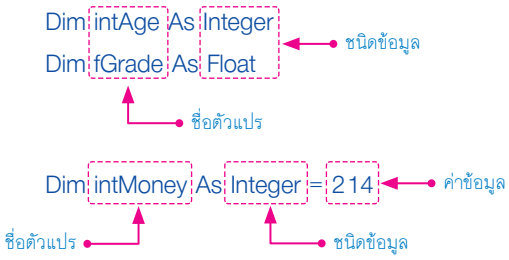
- **คำสั่ง Option Explicit On** หมายถึง ตัวแปรทุกตัวที่ใช้งานต้องมีการประกาศใช้งานด้วยคำสั่ง Dim ก่อนเสมอ
- **คำสั่ง Option Strict On** หมายถึง กำหนดให้ตัวแปลภาษาของ VB มีความเข้มงวดและเคร่งครัดไวยากรณ์ในภาษา VB เทียบเท่ากับภาษา VC# เพื่อให้ได้โค้ดที่มีคุณภาพนั่นเอง

การประกาศตัวแปรสำหรับใช้เก็บข้อมูล

ไม่ว่าผู้อ่านจะเริ่มต้นศึกษาภาษาใดก็ตาม ไวยากรณ์ที่ต้องทำความเข้าใจเป็นลำดับแรก คือ การประกาศตัวแปร (Variable) สำหรับทำหน้าที่เก็บข้อมูล โดยการประกาศตัวแปรในภาษา VB จะใช้คำสั่ง Dim ซึ่งมีไวยากรณ์ดังต่อไปนี้

VB 2015
Dim ชื่อตัวแปร As ชนิดข้อมูล [= ค่าข้อมูล]

ตัวอย่างเช่น



กฎการตั้งชื่อตัวแปร

ในภาษา VB 2015 การตั้งชื่อตัวแปรต่าง ๆ ขึ้นใช้งาน มีกฎข้อบังคับดังนี้

- ต้องประกอบด้วยอักขระ A–Z, a–z, 0–9 หรือ _ เท่านั้น
- อักขระตัวแรกต้องเป็นตัว A–Z, a–z หรือ _ อย่างใดอย่างหนึ่งเท่านั้น (เป็นตัวเลขไม่ได้)
- มีความยาวไม่เกิน 1,023 ตัว
- ต้องไม่ซ้ำกับ Reserved Words ของ VB

Reserved Words

ในภาษา VB 2015 มี Reserved Words ขั้นต้นที่กำหนดให้ห้ามใช้เป็นชื่อตัวแปรต่าง ๆ ดังนี้

AddHandler	AddressOf	Alias	And
AndAlso	As	Boolean	ByRef
Byte	ByVal	Call	Case
Catch	CBool	CByte	CChar
CDate	CDbl	CDec	Char
CInt	Class		CLng
CObj	Const	Continue	CSByte
CShort	CSng	CStr	CType
CUInt	CULng	CUShort	Date
Decimal	Declare	Default	Delegate

Dim	DirectCast	Do	Double
Each	Else	ElseIf	End
	EndIf	Enum	Erase
Error	Event	Exit	
Finally	For		Friend
Function	Get	GetType	
Global	GoSub	GoTo	Handles
If		Implements	
Imports		In	
Inherits	Integer	Interface	Is
IsNot	Let	Lib	Like
Long	Loop	Me	Mod
Module		MustInherit	MustOverride
MyBase		Namespace	Narrowing
New		Next	
Not	Nothing	NotInheritable	NotOverridable
Object	Of	On	Operator
Option	Optional	Or	OrElse
Out	Overloads	Overridable	Overrides
ParamArray	Partial	Private	Property
Protected	Public	RaiseEvent	ReadOnly
ReDim	REM	RemoveHandler	Resume
Return	SByte	Select	Set
Shadows	Shared	Short	Single
Static	Step	Stop	String
Structure		Sub	SyncLock
Then	Throw	To	TRUE
Try	TryCast		UInteger
ULong	UShort	Using	Variant
Wend	When	While	Widening
With	WithEvents	WriteOnly	Xor
#Const	#Else	#ElseIf	#End

ขอบเขตของตัวแปร (Variable Scope)

หลังจากที่ผู้อ่านประกาศตัวแปรที่ต้องการใช้งานแล้ว เนื้อหาลำดับถัดมาที่ควรรับทราบก็คือ ตัวแปรดังกล่าวมีขอบเขตที่สามารถเรียกใช้งานได้แค่ไหน ซึ่งมีขอบเขตอยู่ 3 ระดับ คือ

1. ระดับฟอร์ม ถ้าผู้อ่านประกาศตัวแปรชนิดนี้ไว้ในฟอร์มจะมีขอบเขตกว้างมากที่สุด กล่าวคือ ทุกเหตุการณ์ที่อยู่ในฟอร์มนั้นๆ จะสามารถเรียกใช้งานได้
2. ระดับ Local หรือเรียกอีกอย่างว่า ระดับ Procedure มีขอบเขตขนาดกลาง เป็นระดับที่เหมาะสมกับการใช้งานมากที่สุด คือ มีขอบเขตอยู่ในเหตุการณ์ (Event) แต่ละเหตุการณ์
3. ระดับ Block มีขอบเขตขนาดเล็กที่สุด มักที่จะใช้เก็บค่าข้อมูลแค่ชั่วคราว หรือใช้ในวนลูป (คำสั่ง For, While) เสียเป็นส่วนใหญ่ เช่น ตัวแปรที่อยู่ในบล็อกของคำสั่งต่างๆ เป็นต้น

สำหรับการเลือกใช้ตัวแปรระดับใดก็ตาม ไม่มีข้อกำหนดหรือข้อบังคับใดๆ ทั้งสิ้น ขึ้นอยู่กับลักษณะของโปรเจ็กต์และความเหมาะสม

NOTE

คำว่า เหมาะสม หมายถึง ในการเรียกใช้งานตัวแปรทั้งหมดที่อยู่ในโปรเจ็กต์ของผู้อ่าน สิ่งหนึ่งที่ต้องคำนึงถึงก็คือ หน่วยความจำ (RAM) การเขียนโปรแกรมที่ดีจะต้องมีการใช้งานทรัพยากรของระบบได้เหมาะสมกับความสามารถของตัวมันเอง

ขอบเขตของตัวแปรจะเป็นตัวบ่งบอกว่า ตัวแปรนั้นๆ จะมีอายุในการทำงานนานเท่าไร เพราะเมื่อสิ้นสุดการทำงานของตัวแปรนั้นๆ แอพพลิเคชันจะคืนทรัพยากรที่จองมาใช้งานกลับสู่ระบบ ถ้าผู้อ่านเลือกใช้ขอบเขตและประเภทของตัวแปรได้อย่างเหมาะสมแล้ว โปรเจ็กต์ที่ได้ก็จะมีทั้งประสิทธิภาพในด้านการทำงานและการใช้ทรัพยากรระบบ ตัวอย่างเช่น

ผู้เขียนประกาศตัวแปรขึ้นมา 2 ตัว คือ ตัวแปร num กำหนดให้มีชนิดข้อมูลเป็นตัวเลขจำนวนเต็ม Integer และกำหนดค่าเริ่มต้นให้เก็บค่า 1000 ส่วนตัวแปร str กำหนดให้มีชนิดข้อมูลเป็นข้อความ String และกำหนดค่าเริ่มต้นเป็นชื่อ-สกุลผู้เขียน

VB 2015

```
Option Explicit On
Option Strict On

Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim num As Integer = 1000
        Dim str As String = "Suphachai Somphanit"
    End Sub
End Class
```

เหตุการณ์ ค่าข้อมูล ชนิดข้อมูล ชื่อตัวแปร

ตัวแปรทั้ง 2 ตัวถูกประกาศให้อยู่ภายในเหตุการณ์ Form1_Load() เราเรียกตัวแปรประเภทนี้ว่า **ตัวแปรที่มีขอบเขตแบบ Local** หมายถึง สามารถใช้งานได้เฉพาะ “ภายในเหตุการณ์ Form1_Load()” เท่านั้น

ต่อมาผู้เขียนลองประกาศตัวแปรอีกลักษณะหนึ่ง โดยตำแหน่งที่ประกาศตัวแปรอยู่ภายนอกเหตุการณ์ต่าง ๆ กล่าวคือ สร้างตัวแปรที่ชื่อว่า str ขึ้นมา และกำหนดให้มีชนิดข้อมูลเป็นข้อความ String

VB 2015

```
Option Explicit On
Option Strict On

Public Class Form1
    Dim str As String = ""

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        str = "Visual Basic"
    End Sub
                                    เหตุการณ์
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        str = "Visual Basic"
    End Sub
End Class
```

ในกรณีนี้ตัวแปร str ถูกประกาศอยู่ภายนอกเหตุการณ์ต่าง ๆ เรียกว่า ตัวแปรที่มีขอบเขตระดับฟอร์ม ผู้อ่านสามารถเรียกใช้ตัวแปร str นี้ ในเหตุการณ์ต่าง ๆ ได้ ในกรณีนี้ คือ เหตุการณ์ Form1_Load() กับเหตุการณ์ Button1_Click()

ตัวแปรระดับ Local และระดับฟอร์ม

เพื่อเพิ่มความเข้าใจในการใช้งานตัวแปรระดับ Local และระดับฟอร์ม ให้ผู้อ่านศึกษาการใช้งานตัวแปรจากตัวอย่างต่อไปนี้

1. ออกแบบฟอร์มดังรูป 2-2
2. เขียนโค้ดในเหตุการณ์ Form1_Load() ดังนี้

ตัวอย่างที่ 2-1 ตัวแปรระดับ Local และระดับฟอร์ม (Form1.vb)

```
Option Explicit On
Option Strict On
Public Class Form1
    Dim str As String = "ตัวแปรระดับฟอร์ม"
                                    เหตุการณ์ Form1_Load
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'Dim str As String = "ตัวแปรระดับ Local"
        Label1.Text = str
    End Sub
End Class
```

3. เขียนโค้ดในเหตุการณ์ Button1_Click() ดังนี้

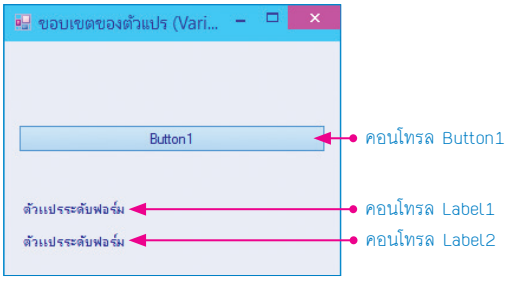
ตัวอย่างที่ 2-1 ตัวแปรระดับ Local และระดับฟอร์ม (Form1.vb)

```
Option Explicit On
Option Strict On
Public Class Form1
    Dim str As String = "ตัวแปรระดับฟอร์ม"

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'Dim str As String = "ตัวแปรระดับ Local"
        Label1.Text = str
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Label2.Text = str
    End Sub
End Class
```

1 เหตุการณ์ Button1_Click

4. เมื่อรันโปรแกรมจะได้ผลลัพธ์ดังนี้

รูปที่ 2-2 ผลการรันตัวอย่างที่ 2-1



จากรูปที่ 2-2 ผู้เขียนประกาศตัวแปรในระดับฟอร์มที่ชื่อว่า str เห็นได้ว่า คอนโทรล Label1 ถูกกำหนดค่าในเหตุการณ์ Form1_Load() ส่วนคอนโทรล Label2 ถูกกำหนดค่าในเหตุการณ์ Button1_Click() เห็นได้ว่าการประกาศตัวแปรในลักษณะนี้ ผู้อ่านสามารถใช้งานตัวแปรตัวนี้ได้หลายเหตุการณ์

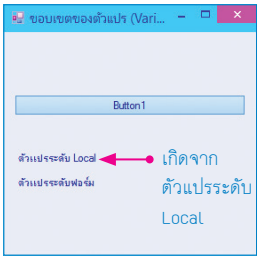
5. ให้ผู้อ่านนำคอมเมนต์โค้ดบรรทัดต่อไปนี้ออก แล้วลองรันตัวอย่างอีกครั้ง ดังรูปที่ 2-3

VB 2015

```
        Dim str As String = "ตัวแปรระดับ Local"
```

6. เมื่อรันโปรแกรมจะได้ผลลัพธ์ดังนี้

รูปที่ 2-3 หลังจากเอาหมายเหตุออก



จากรูปที่ 2-3 ผู้เขียนสร้างตัวแปรอีก 1 ตัว มีขอบเขตระดับ Local จงใจให้มีชื่อเดียวกับตัวแปรระดับฟอร์มข้างต้น พบว่า คอนโทรล Label1 แสดงข้อความของตัวแปรในระดับ Local

สรุปได้ว่า ถ้ามีตัวแปรชื่อเดียวกันทั้งในระดับฟอร์มกับระดับ Local ตัวแปรที่อยู่ในระดับ Local มีความสำคัญมากกว่าตัวแปรที่อยู่ในระดับฟอร์มนั้นเอง

ตัวแปรระดับ Block

เพื่อเพิ่มความเข้าใจในการใช้งานตัวแปรระดับ Block ซึ่งเป็นตัวแปรที่มีขอบเขตแคบที่สุด ให้ผู้อ่านศึกษาการใช้งานตัวแปรจากตัวอย่างต่อไปนี้

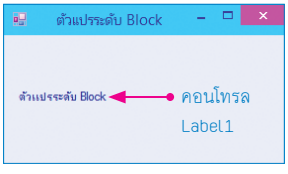
1. ออกแบบฟอร์มดังรูป 2-4
2. เขียนโค้ดในเหตุการณ์ Form1_Load() ดังนี้

ตัวอย่างที่ 2-2 ตัวแปรระดับ Block (Form1.vb)

```
Option Explicit On
Option Strict On

Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Dim i As Integer
        For i = 0 To 2
            Dim strVal As String = "ตัวแปรระดับ Block"
            Label1.Text = strVal
        Next
        "Label1.Text = strVal
    End Sub
End Class
```


3. เมื่อรันโปรแกรมจะได้ผลลัพธ์ดังนี้

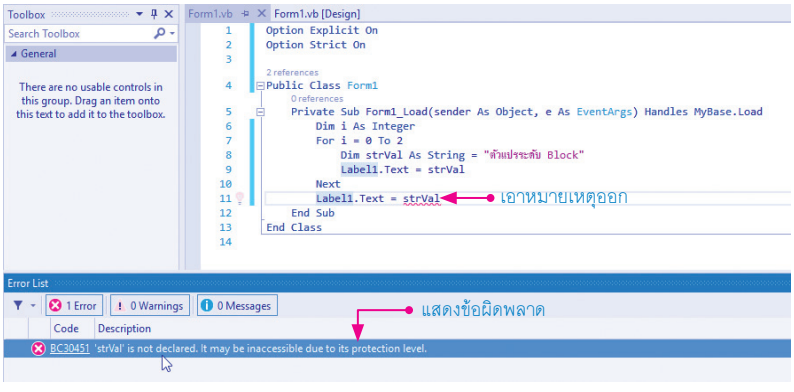
รูปที่ 2-4 ผลการรันตัวอย่างที่ 2-2



จากรูปที่ 2-4 ผู้เขียนประกาศตัวแปรที่ชื่อว่า strVal ในบล็อกคำสั่งของลูป For ส่งผลให้ตัวแปรตัวนี้มีขอบเขตการใช้งานเฉพาะภายในบล็อกของลูป For Next นี้เท่านั้น

4. ให้ผู้อ่านเอาคอมเมนต์ออก พบว่า ไม่สามารถอ่านค่าของตัวแปร strVal ตัวนี้นอกลูป For ได้เลย ดังรูปที่ 2-5

รูปที่ 2-5 แสดงการใช้งานตัวแปร strVal นอกบล็อกคำสั่ง



จากรูปที่ 2-5 ข้อผิดพลาดที่ปรากฏขึ้นมา เกิดจากตัวแปร strVal ยังไม่มีการประกาศใช้งาน เพราะว่าเราเรียกใช้งานตัวแปรตัวนี้เกินขอบเขตของมันนั่นเอง