

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

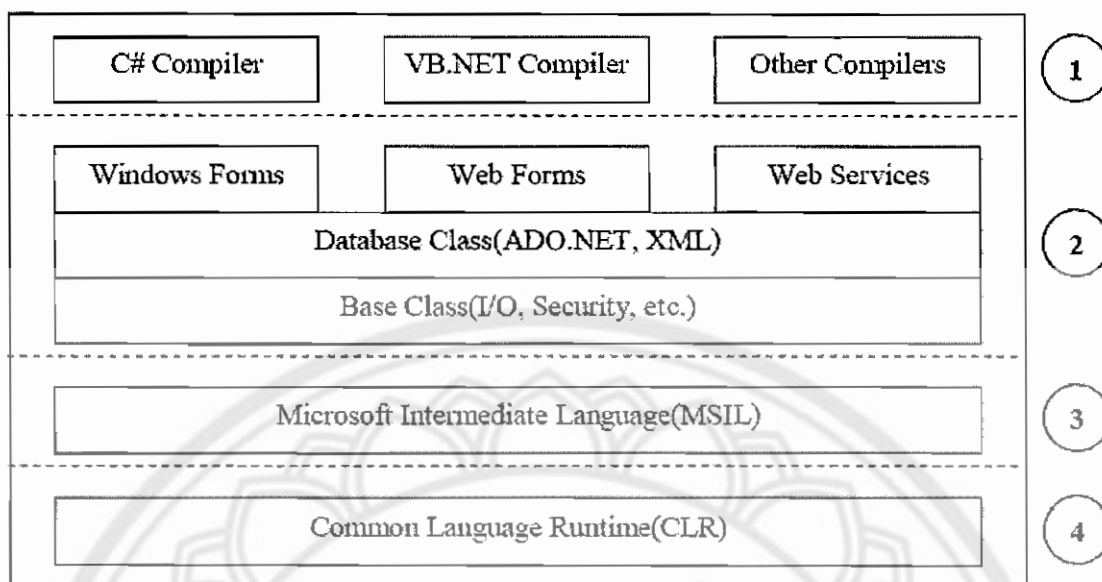
ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆที่เกี่ยวข้องกับการพัฒนาระบบทดสอบมาตรฐาน อุปกรณ์การวัดทางเคมีซึ่งเนื้อหาในบทนี้ จะกล่าวถึง .NET Framework ที่ใช้ในการพัฒนาโปรแกรมประยุกต์สำหรับทดสอบมาตรฐานอุปกรณ์การวัดทางเคมี การทำงานกับฐานข้อมูล รวมถึงเนื้อหาในส่วนวินโดว์ฟอร์มด้วย

2.1 .NET Framework (Microsoft.NET Framework)

2.1.1 .NET Framework คืออะไร

.NET Framework คือ กรอบการทำงานของการเขียนโปรแกรมที่ไม่โครซอฟท์คิดขึ้นมาเพื่อรองรับการติดต่อสื่อสาร เพื่อแลกเปลี่ยนข้อมูล(Exchange Data) ระหว่างกัน หรือแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์ม(Platform) ให้มีความสมบูรณ์ยิ่งขึ้น โดยอาศัยภาษา XML(Extensible Markup Language) ทำหน้าที่เป็นตัวกลางในการแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์มไฟล์ของฐานข้อมูล .NET Framework ถูกออกแบบมาเพื่อให้สามารถถูกใช้จากภาษาใดๆ ก็ได้ ไม่ว่าจะเป็น ภาษา C#, C++, Visual Basic, JScript, Delphi และอื่นๆ เพื่อให้สิ่งเหล่านี้เป็นไปได้ จึงเกิดภาษาเหล่านี้ ขึ้นมาในรูปของ Version เฉพาะ สำหรับ .Net อีกด้วย ได้แก่ภาษา Managed C++, Visual Basic.Net, Jscript .Net, Borland C#, Delphi8 เป็นต้น และมีภาษาอีกมากมายที่กำลังพัฒนา และปล่อยออกสู่ท้องตลาดอยู่ตลอดเวลา ไม่ใช่เพียงแค่ภาษาทั้งหมดเหล่านี้ จะมีการเข้าถึง .NET Framework เท่านั้น แต่มันยังสามารถสื่อสารกับภาษาอื่นๆได้อีกด้วย

2.1.2 โครงสร้างภายในของ .NET Framework



รูปที่ 2.1 แสดงโครงสร้างของ .NET Framework

จากรูป สามารถแบ่ง โครงสร้างภายในของ .NET Framework ได้เป็น 4 ชั้นหลักๆ ดังนี้

1. Programming Languages เนื่องจาก .NET Framework สามารถถูกใช้จากภาษาใดๆ ก็ได้ ไม่ว่าจะเป็นภาษา C#, C++, Visual Basic, JScript, Delphi และอื่นๆ ดังนั้นจะต้องเลือกตัวคอมไพเลอร์ที่เหมาะสมกับภาษานั้นๆ เช่นถ้าเขียนด้วย C# ก็ต้องใช้ C# Compiler เป็นต้น
2. Base Class เป็น class library พื้นฐาน ที่โปรแกรมต่าง ๆ ไม่ว่าจะเขียนด้วยภาษาใดบน .Net ก็สามารถใช้ร่วมกันได้ ซึ่งในชั้นนี้จะมีการรวบรวมเอาออบเจกต์ (Object) และคอนโทรล (Control) ที่โค้ดอ้างอิงถึงหรือเรียกใช้ มาตรวจสอบว่าเรียกใช้ถูกต้องตามคุณลักษณะของออบเจกต์ หรือคอนโทรลนั้นๆหรือไม่ ถ้าตรวจสอบแล้วถูกต้องก็จะส่งผ่านไปยังชั้นที่ 3 ซึ่งออบเจกต์และคอนโทรลที่ .NET Framework ได้เตรียมไว้นั้นแบ่งออกเป็น 5 กลุ่มหลักๆ คือ

- Web Service รวบรวมออบเจกต์และคอนโทรลต่างๆที่เกี่ยวข้องกับการสร้างฟังก์ชัน (Function) หรือซับรูทีน(Subroutine) เพื่อให้บริการบนอินเทอร์เน็ตหรือที่เรียกว่า เว็บเซอร์วิส(Web Service) นั่นเอง เว็บเซอร์วิสช่วยให้เราสามารถเรียกใช้ฟังก์ชัน หรือซับรูทีนในอินเทอร์เน็ตที่มีผู้เขียนไว้แล้วได้
- Web Forms รวบรวมออบเจกต์และคอนโทรลต่างๆที่เกี่ยวข้องกับการแสดงผลบนอินเทอร์เน็ต ซึ่งส่วนนี้จะเกี่ยวข้องกับเทคโนโลยี ASP.NET
- Windows Forms รวบรวมออบเจกต์และคอนโทรลต่างๆที่เกี่ยวข้องกับการแสดงผลบน Windows ซึ่งส่วนนี้จะเกี่ยวข้องกับการพัฒนาโปรแกรมทั่วไปบน Windows

- Database Class รวบรวมออบเจกต์และคอนโทรลที่ใช้ติดต่อและจัดการกับฐานข้อมูล รวมถึงข้อมูลแบบ XML ด้วย
- Base Class ส่วนนี้เป็นส่วนที่เกี่ยวข้องกับการทำงานขั้นพื้นฐานต่างๆ เช่น เรื่องของอินพุต/เอาต์พุต เป็นต้น

3. Microsoft Intermediate Language (MSIL) หรือเรียกสั้นๆว่า IL เป็นภาษามาตรฐานของไมโครซอฟท์ เพราะฉะนั้นไม่ว่าโค้ดจะถูกเขียนด้วยภาษาใดก็ตาม ในขั้นนี้จะถูกแปลให้เป็นภาษาเดียวกันหมดนั่นคือ ภาษา IL ซึ่งภาษา IL จะถูกเก็บไว้ในไฟล์นามสกุล .exe ทำให้ .exe นี้ไม่สามารถทำงานได้บนเครื่องคอมพิวเตอร์ทั่วไปที่ปราศจาก .NET Framework ได้เพราะไม่มีระบบปฏิบัติการได้อ่านภาษา IL ได้

4. Common Language Runtime (CLR) คือผู้เดียวที่อ่านภาษา IL ได้ มันจึงเป็นตัวกลางระหว่างไฟล์ .exe (ที่เก็บภาษา IL ไว้) กับระบบปฏิบัติการ กล่าวคือ ไฟล์นามสกุล .exe ที่เป็น IL นั้นต้องอาศัย CLR ของ .NET Framework ช่วยในการแปลภาษา IL เป็นภาษาเครื่องที่ระบบปฏิบัติการเข้าใจและปฏิบัติตามได้

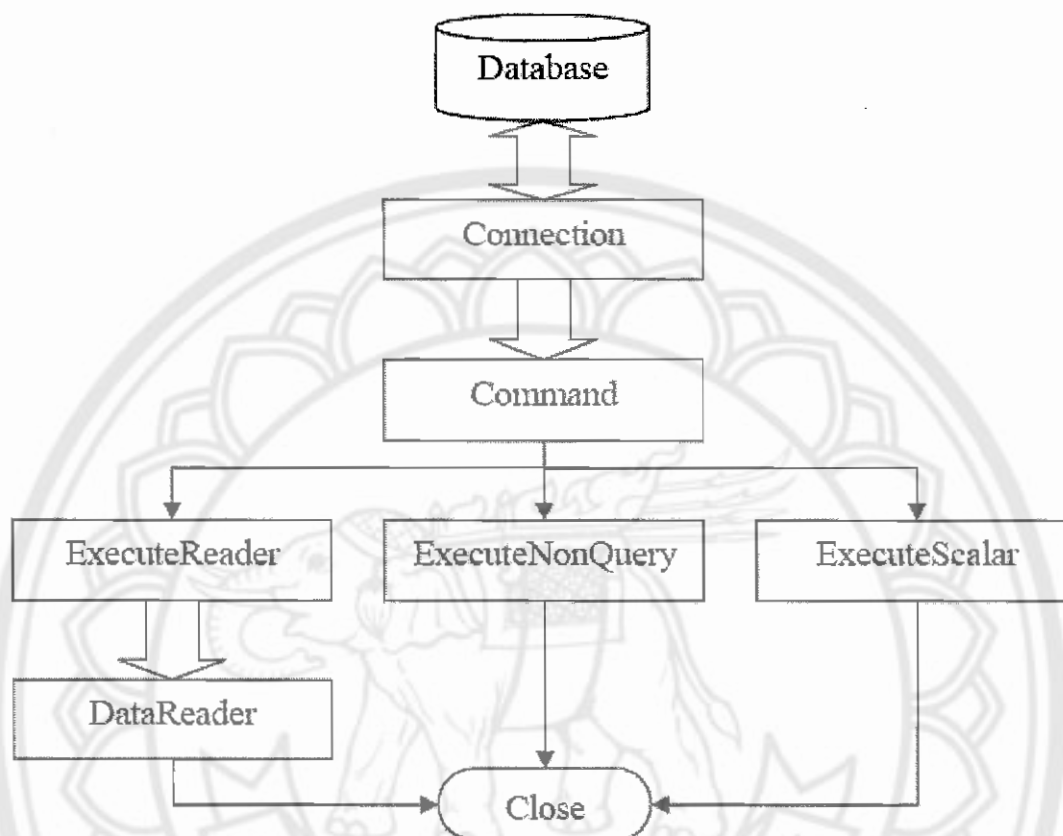
2.1.3 ข้อดีของ .NET Framework

1. ใช้ในการพัฒนาได้ทุกภาษา ทำให้เราไม่ต้องมาศึกษาภาษาใหม่ๆเมื่อต้องการสร้างโปรแกรมในแต่ละครั้ง นอกจากนั้นเรายังสามารถเลือกใช้ภาษาที่เราถนัดที่สุดในการพัฒนาโปรแกรมต่างๆได้ด้วย
2. เป็นระบบที่มีไลบรารีที่เป็นมาตรฐานเดียวกัน เนื่องจากมีไลบรารีที่เป็นมาตรฐานเดียวกันทั้งหมดทำให้เราไม่ต้องกังวลว่าภาษาที่ใช้เขียนนั้นมีไลบรารีตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องกังวลว่าถ้าใช้ไลบรารีของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มีไลบรารีตัวนั้น
3. สามารถนำเอาโค้ดภาษาหนึ่งผ่านการคอมไพล์เป็น IL แล้วไปใช้งานหรือพัฒนาต่อในอีกภาษาหนึ่งได้
4. CLR ทำงานได้เร็วกว่า JVM (Java Virtual Machine) เนื่องจาก CLR จะแปลภาษา IL ที่อยู่ในไฟล์ .exe ไปเป็นภาษาเครื่องรวดเดียวแล้วค่อยทำงาน (Compile) ในขณะที่ JVM จะแปล Byte code ในไฟล์ .class ไปเป็นภาษาเครื่องด้วยวิธีแปลไปทำงานไป (Interpreter) นอกจากนี้ CLR ยังฉลาดพอที่จะไม่คอมไพล์ทีเดียวทั้งหมด แต่จะใช้เทคนิคที่เรียกว่า Just In Time (JIT) เพื่อคอมไพล์ IL เฉพาะส่วนที่จำเป็นต้องใช้งาน หลังจากนั้นเมื่อต้องมีส่วนใดเพิ่มเติมก็ค่อยคอมไพล์เฉพาะส่วนนั้น และจะไม่คอมไพล์ส่วนที่เคยถูกคอมไพล์ไปแล้ว
5. มีภาษา C# ที่มีไวยากรณ์และโครงสร้างของภาษากลายกับภาษา Java มากทำให้ผู้ที่เขียน Java มาก่อนสามารถเปลี่ยนมาใช้สภาพแวดล้อมของ .NET Framework ได้ทันที

2.1.4 การติดต่อและจัดการกับฐานข้อมูลด้วย ADO.NET

ADO.NET (ActiveX Data Object .NET) ทำหน้าที่รับผิดชอบในส่วนของการจัดการข้อมูลในฐานข้อมูลประเภทต่างๆ ของสถาปัตยกรรม .NET Framework แบ่งได้เป็น 2 รูปแบบดังนี้

1. การติดต่อฐานข้อมูลแบบ Connect นั้นหมายความว่าเมื่อเราสั่งให้ ADO.NET เปิดฐานข้อมูลเพื่อทำอะไรซักอย่าง มันจะรักษาสภาพการติดต่อนั้นไว้ตลอดจนกว่าเราจะบอกเลิกการติดต่อ หรือกล่าวได้ว่าโปรแกรมของเราและฐานข้อมูลจะมีการติดต่อกันตลอดเวลา



รูปที่ 2.2 แสดงโครงสร้างการติดต่อกับฐานข้อมูลแบบ Connect

จากรูป สามารถแบ่งขั้นตอนการติดต่อกับฐานข้อมูลแบบ Connect ได้เป็น 4 ขั้นตอน ดังนี้

- สร้าง Connection การเขียนโปรแกรมจะเริ่มจากการสร้าง Connection (ออบเจกต์ชนิด OdbcConnection) เพื่อทำหน้าที่เชื่อมต่อกับฐานข้อมูลที่ต้องการ

```

String strProvider;
strProvider = "ชื่อ Provider ของฐานข้อมูลประเภทนั้นๆ";
OdbcConnection conn = new OdbcConnection(strProvider);
Conn.Open();
  
```

- สร้าง Command หลังจากสร้าง Connection เพื่อเชื่อมต่อกับฐานข้อมูลแล้วต่อจากนั้นจะต้องสร้าง Command (ออบเจกต์ชนิด OdbcCommand) เพื่อสั่งดำเนินการใดๆกับฐานข้อมูล

```
OdbcCommand cmd = new OdbcCommand();
cmd.Connection.conn;
cmd.CommandText = "คำสั่ง SQL ที่ใช้ดำเนินการกับฐานข้อมูล";
```

- ดำเนินการกับฐานข้อมูล เมื่อสร้าง Connection และ Command แล้ว ต่อจากนั้นจะเห็นว่ามี 3 ทาง(ในเบื้องต้น) ซึ่งจะไปทางไหนนั้นขึ้นอยู่กับคำสั่ง SQL ที่ใช้ดังนี้

1) ExecuteReader หากใช้คำสั่ง SQL เพื่อดึงข้อมูลเพียงอย่างเดียว ซึ่งจะต้องประกาศตัวแปรชนิด OdbcDataReader เพื่อรับ DataReader(ออบเจกต์ชนิด OdbcDataReader) เพื่อส่งคืนมาจากเมธอด (Method) ExecuteReader()

```
cmd.CommandText = "SELECT * FROM Table1";
OdbcDataReader dr = cmd. ExecuteReader();
```

2) ExecuteNonQuery หากใช้คำสั่ง SQL เพื่อเพิ่ม, ลบ หรือ แก้ไขข้อมูล

```
cmd.CommandText = "DELETE FROM Table1";
cmd. ExecuteNonQuery();
```

3) ExecuteScalar หากใช้คำสั่ง SQL เพื่อดึงข้อมูลที่มีผลลัพธ์เพียงข้อมูลเดียว

```
cmd.CommandText = "SELECT COUNT (*) FROM Table1";
Int numRecord = (int) cmd. ExecuteScalar();
```

- ปิดการเชื่อมต่อกับฐานข้อมูล หลังจากใช้งานฐานข้อมูลเสร็จแล้ว ควรจะปิดการเชื่อมต่อกับฐานข้อมูลทันทีโดยเรียกเมธอด Close() ของ Connection

```
Conn.Close();
```

2. การติดต่อฐานข้อมูลแบบ Disconnect นั้นหมายความว่าเมื่อเราสั่งให้ ADO.NET เปิดฐานข้อมูลเพื่อทำอะไรสักอย่าง มันจะทำการติดต่อกับฐานข้อมูล จากนั้นจะดึงข้อมูลมาเก็บไว้ในหน่วยความจำ แล้วปิดฐานข้อมูลทันที หลังจากนั้นโปรแกรมของเราจะติดต่อกับข้อมูลที่อยู่ในหน่วยความจำแทน

2.2 Windows Form

ใน .NET Framework มีวิธีการแสดง และจัดการกับ User Interface 3 ทาง ได้แก่ Windows Form, Web Form และ Console Application โดยในบทนี้จะกล่าวถึง Windows Form และ Control ที่มีอยู่ใน Windows Form

2.2.1 ทำความรู้จักกับ Windows Form

Windows Form (หรือเรียกย่อๆว่า “WinForms”) เป็นการแสดงหน้าจอตามมาตรฐาน Win32 โดยใช้เทคโนโลยี Windows Foundation Classes (WFC) ซึ่งเป็นการพัฒนาเพื่อใช้กับ Visual J++ และสามารถนำมาใช้กับการพัฒนา Windows Form ได้

Windows Form เป็นส่วนหนึ่งของ Bases Class ใน .NET Framework ที่อยู่ใน Namespace “System.Windows.Forms” ซึ่งเป็นส่วนที่ใช้สร้างจอภาพของ Project ประเภท Windows Application

Windows Form ใน Visual Basic 2005 พัฒนามาจาก Form ของ Visual Basic สามารถใช้งานได้กับทุกภาษาที่อยู่บน .NET Framework โดยแบ่งการใช้งานของ Windows Form ออกเป็นการทำงานกับ Control และการใช้งานกับฟังก์ชัน Graphic ซึ่งการใช้งานเกี่ยวกับ Graphic นี้ใน Windows Form สามารถทำได้ดีเท่ากับการใช้ API (Application Programming Interface) ของ Windows

1. ความสำคัญของ Windows Form

เนื่องจากการใช้งานแอปพลิเคชันในปัจจุบัน ผู้ใช้งานต้องการใช้งานฟังก์ชันที่มากขึ้นและมีความยืดหยุ่นสูงขึ้น ซึ่ง .NET Framework สามารถตอบสนองความต้องการดังกล่าวได้ เช่น การสร้างแอปพลิเคชันให้ทำงานบน Browser ซึ่งต้องแบ่งเงินลงทุนในการพัฒนาออกเป็น 2 ส่วน โดยเงินทุนส่วนใหญ่จะใช้กับการพัฒนาการทำงานของโปรแกรมในแอปพลิเคชัน และส่วนที่เหลือจึงจะนำมาใช้พัฒนา User Interface บน Browser แต่หากสามารถใช้โค้ดเดียวกันกับการทำงานทั้งสองส่วนก็จะช่วยลดต้นทุนในการพัฒนาโปรแกรม ซึ่งปัจจุบัน Windows Form บนเทคโนโลยี .NET มีเทคนิคที่สามารถเชื่อมโยงการทำงานของแอปพลิเคชันให้ใช้ User Interface บน Browser ได้

ในเทคโนโลยี .NET มีการสร้างแบบจำลองของโปรแกรมขึ้นมาที่เรียกว่า “Window DNA” เพื่อให้แอปพลิเคชันสามารถใช้งานได้ 2 Interface ได้แก่ Win32 Interface และ Web Interface หรือสามารถอธิบายโดยยกตัวอย่างการสร้าง Project ใน .NET Framework ซึ่งต้องการให้แอปพลิเคชันใน Project สามารถทำงานได้ 2 กรณี คือกรณีแรกให้ทำงานอยู่บน Windows Form และกรณีที่สองให้ทำงานบน Web Form หรือ Active Server Pages ของ ASP.NET โดยที่ทั้งสอง Interface ต้องทำงานอยู่

บน Component เดียวกัน ตัวอย่างเช่น การส่งซื้อสินค้าแบ่งการทำงานออกเป็น 2 Interface คือ Win32 Interface สำหรับให้เจ้าหน้าที่ปฏิบัติงานทางโทรศัพท์กรอกยอดการสั่งซื้อสินค้าโดยใช้ Windows Form เพื่อช่วยให้พนักงานสามารถทำงานได้รวดเร็วขึ้น และ Web Interface สำหรับให้ลูกค้าเข้าไปในระบบได้โดยตรงเพื่อกรอกยอดการสั่งซื้อสินค้าลงบน Web Form ด้วยตัวเอง แต่การทำงานของทั้งสอง Interface จะต้องอยู่บนระบบเดียวกัน

การสร้างแอปพลิเคชันที่มีทั้ง Win32 Interface และ Web Interface ทำได้โดยการรวมเอา Component ของแต่ละชั้นเข้าด้วยกัน และจัดการกับแบบจำลอง Interface ของ Component ให้สามารถใช้ได้กับทุกภาษาบนเทคโนโลยี .NET ซึ่งรวมทั้งภาษาสคริปต์ด้วย จึงไม่จำเป็นต้องสร้าง Component เวอร์ชันพิเศษ แต่ต้องสร้างตัวแปรขึ้นใน Interface ซึ่งปัจจุบันแนวคิดเกี่ยวกับการสร้าง Interface ดังกล่าวยังไม่สามารถใช้งานได้จริง

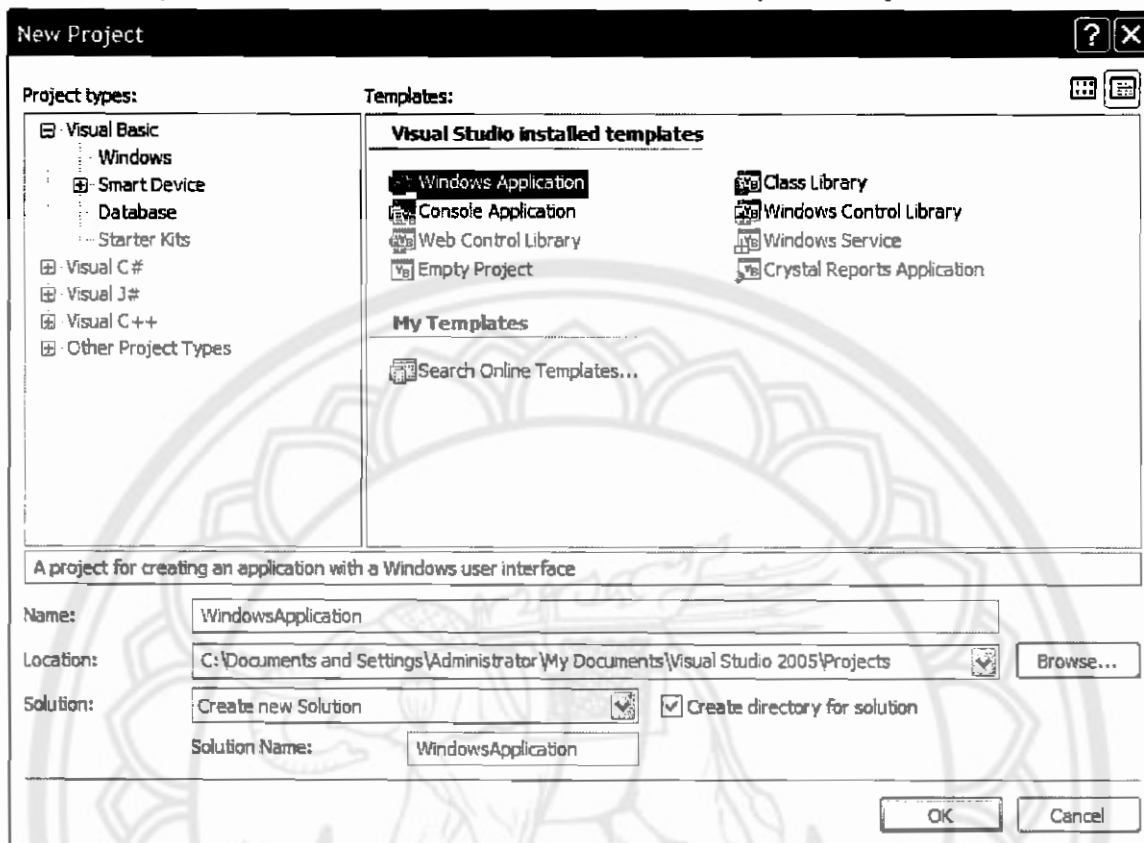
ดังนั้นวิธีที่ง่ายที่สุด คือ การรวม Windows Form เข้ากับ Web Service เช่นเดียวกับการรวม Web Form เข้ากับ Web Service ซึ่งทำให้ Windows Form สามารถใช้ Win32 Interface พร้อมทั้งจัดการกับข้อมูลบน Internet Server ผ่านทาง Web Service ได้เช่นเดียวกับการทำงานของ Web Form โดย Windows Form ใน Visual Basic 2005 จะสามารถเข้าถึงการทำงานของเว็บได้โดยใช้ Control พิเศษ ซึ่งได้รวมฟังก์ชันสำหรับใช้งานจำนวนมากไว้ให้ Windows Form

2. พื้นฐานของ Windows Form

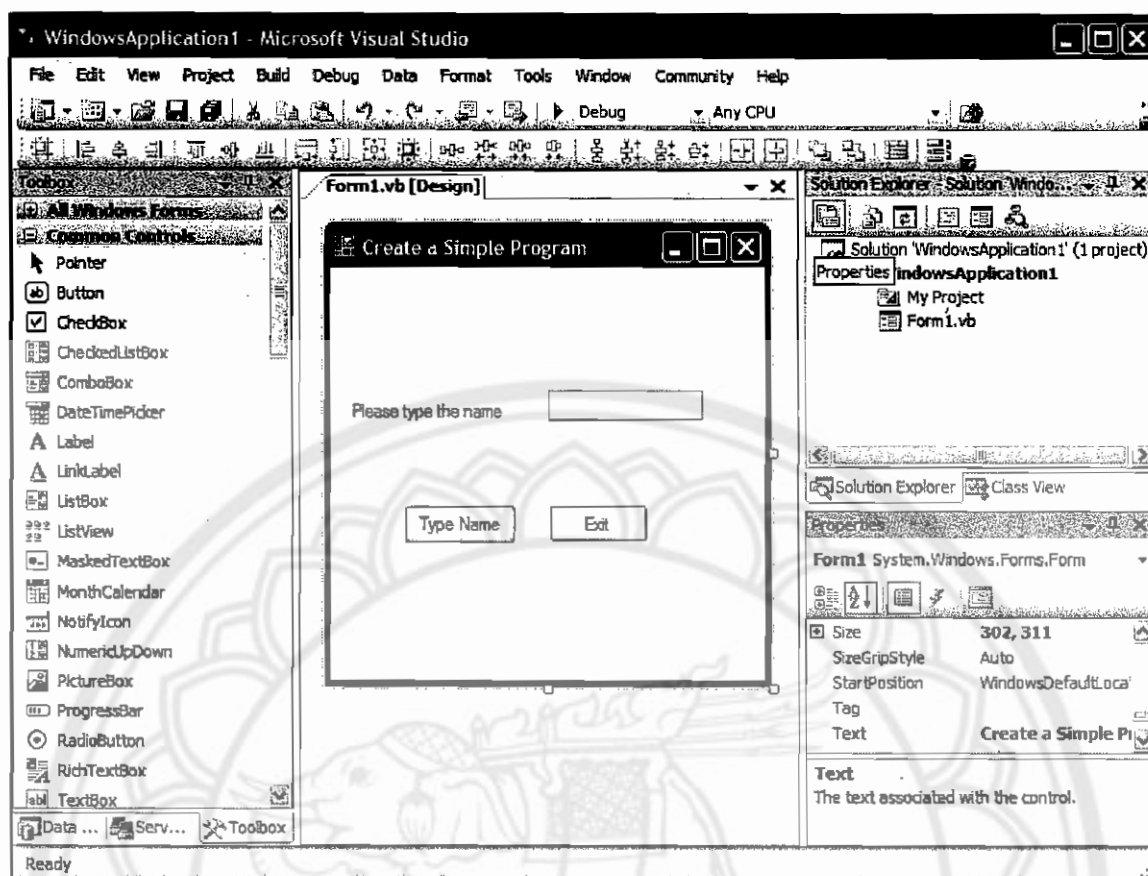
สามารถสรุปประเด็นสำคัญในการสร้าง Windows Form ได้ดังนี้

- โปรแกรมของ Windows Form มีลักษณะเป็น Class อยู่ในรูปของ Object Oriented Programming
- เนื่องจาก Form ของ Visual Basic 2005 อยู่ในรูปของ Class และไม่มีรูปแบบคำสั่งเพื่อใช้ Load Form จึงต้องกำหนด Instance ขึ้นแทน Form ที่ต้องการใช้งาน แล้วจึงเรียก Form ที่ต้องการใช้งานขึ้นมาแสดง
- Class ที่มีคุณสมบัติเช่นเดียวกับ Windows Form จะต้องเป็น Class ที่สืบทอดคุณสมบัติจาก Class ของ System.Windows.Forms
- Windows Form มีคุณสมบัติเหมือนกับทุก Class ที่อยู่ใน .NET Framework ซึ่งจะสามารถสร้าง Constructor โดยใช้คำสั่ง Sub New เพื่อ Load Form และ Destructor โดยใช้คำสั่ง Sub Dispose เพื่อ Unload Form
- การออกแบบ Form ด้วยการเขียนโค้ด ทำได้โดยเขียนลงในส่วนของ Class ที่ใช้จัดการกับ Form ซึ่งจะอยู่ในตำแหน่งเริ่มต้นของไฟล์
- การกำหนดการทำงานของ Form ใน Visual Basic 2005 จะใช้ Events ซึ่งมีข้อมูลของ Argument ให้ใช้งานจำนวนมาก

โปรแกรมที่ใช้ Windows Form เป็น Interface จะเริ่มต้นด้วยการสร้าง Windows Application โดยคลิกที่เมนู File -> New Project (ไอคอน New Project...) จะปรากฏหน้าต่างดังรูป



รูปที่ 2.3 แสดงการเปิดโปรเจกใหม่



รูปที่ 2.4 แสดงการสร้างแอปพลิเคชันใหม่

โครงสร้างของ Windows Form ทำหน้าที่เป็นพื้น (Background) ของหน้าจอโปรแกรม ดังรูปของ Form ข้างต้น ซึ่ง Control ที่อยู่ใน Toolbox สามารถนำมาสร้างบน Form ด้วยวิธีการต่างๆ

2.2.2 สถาปัตยกรรมของ Windows Form

Windows Form เป็น Class หนึ่งที่อยู่ใน Namespace “System” ภายใน Windows Form ประกอบด้วยกลุ่มของ Class จำนวนมากที่อยู่ในรูปของ Object โดยมีโครงสร้างแบบลำดับชั้น และจะสืบทอดคุณสมบัติจาก Class ต้นแบบ รวมทั้งเพิ่มเติมฟังก์ชันต่างๆ เข้าไปใน Class แต่ละลำดับชั้นสามารถแสดงโครงสร้าง Class แบบลำดับชั้น ดังแผนผังต่อไปนี้

จากรูปแสดง Class สุดท้ายของแต่ละลำดับชั้นที่สืบทอดคุณสมบัติจาก Class ต้นแบบ (Class “Control”) โดยจะแสดงเฉพาะ Class ที่สำคัญเพื่อให้เข้าใจการทำงานของ Windows Form

โครงสร้างแบบลำดับชั้นข้างต้นมี Class ที่สำคัญอยู่ 2 Class คือ Form และ UserControl ซึ่งคุณสมบัติทั้งหมดที่นำมาใช้ใน Windows Form สืบทอดจาก Class สุดท้ายที่เชื่อมต่อกับ Class “Form” ส่วน Control ต่างๆ ที่ใช้ใน Visual Basic 2005 จะสืบทอดคุณสมบัติจาก Class “UserControl” หรือจะสืบทอดจาก Class สุดท้ายที่เชื่อมต่อกับ UserControl

ในแต่ละ Class จะมีกลุ่มของสมาชิกที่สร้างขึ้น และสามารถนำคุณสมบัติที่สร้างขึ้นถ่ายทอดไปยัง Class ที่อยู่ต่ำกว่าในโครงสร้างแบบลำดับชั้น ตัวอย่างเช่น Class “Control” สร้าง Property “Left” ขึ้นมาคล้ายกับการทำงานของ Property “Left” ของ Form หรือ Control ใน Visual Basic 2005 แต่สามารถใช้กับทุก Class ที่อยู่ภายใต้โครงสร้างแบบลำดับชั้นเหมือนกัน

1. Class “Control”

Class “Control” อยู่ในลำดับชั้นบนสุด ประกอบด้วยฟังก์ชันพื้นฐานเพื่อกำหนดขอบเขตของหน้าจอ จัดการกับกระบวนการทำงานของ Message และการแสดงผลของหน้าจอ เช่น สามารถรับค่า Input ของ User ผ่านทางคีย์บอร์ด และเมาส์เพื่อสร้างส่วนแสดงผล โดย Class “Control” จะให้บริการเหมือนกับเป็น Base Class สำหรับบาง Component ที่ต้องการแสดงผลในรูปของ Win32 API โดยการพัฒนาแอปพลิเคชันทั่วไปไม่จำเป็นต้องสืบทอดสมบัติจาก Base Class โดยตรง แต่สามารถสืบทอดคุณสมบัติจาก Class ย่อยอื่นๆ ที่มีคุณสมบัติเหมาะสมกว่า

ในส่วนนี้จะแสดง Member ของ Class “Control” เฉพาะส่วนที่สำคัญสำหรับการใช้งานใน Visual Basic 2005 ดังนี้

ตารางที่ 2.1 แสดง Property ของ Class “Control”

ชื่อ Property	การใช้งาน
CanFocus	เป็น Property ที่อ่านอย่างเดียว (ReadOnly) ซึ่งเป็นตัวชี้ หรือเป็น Focus ของ Control
CauseValidation	เป็น Property ของ Form และ Control ใช้สำหรับตรวจสอบค่าที่ป้อนเข้ามาใน Control
Controls	เป็นที่เก็บ Child Control ซึ่งมี Control ต่างๆอยู่ข้างใน
Enabled	เป็น Property สำหรับเปิดใช้งาน Control
Handle	ใช้จัดการตัวแปรชนิด HWND (เป็น Parameter ใช้เก็บค่าสำหรับจัดการ Object ต่างๆ ใช้ Windows)
Top, Left, Height , Width, Bottom, Right, Location, Size	ระบุค่าที่เกี่ยวข้องกับขนาด และตำแหน่งของ Control
Visible	เป็นค่าที่กำหนดให้แสดง Control บนหน้าจอ

ตารางที่ 2.2 แสดง Method ของ Class “Control”

ชื่อ Method	การใช้งาน
BringToFront	นำ Control มาไว้ข้างหน้า Zorder
Focus	ใช้ตั้งค่า Focus ของ Control
Hide	ใช้ซ่อน Control โดยกำหนดค่าให้ Property “Visible” มีค่าเป็น False
Refresh	ปรับปรุงการแสดงผลของ Control และ Child Control
Show	แสดงให้เป็น Control โดยกำหนดค่าให้ Property “Visible” มีค่าเป็น True
Update	ปรับปรุงการแสดงผลให้เป็นปัจจุบันในส่วนที่ไม่ถูกต้อง
WndProc	เป็น Method ที่ใช้กับ Windows Message เมื่อมีการกระทำใน Class “Control” และทุก Class ของ Windows Form จะสามารถใช้ Windows Message ได้

ตารางที่ 2.3 แสดง Event ของ Class “Control”

ชื่อ Event	การใช้งาน
MouseEnter	ทำงานเมื่อเลื่อนเมาส์เข้าไปในบริเวณ Control 1 โดยไม่ต้องคลิกที่ Control
MouseLeave	ทำงานเมื่อเลื่อนเมาส์ออกจากบริเวณของ Control โดยไม่ต้องคลิกที่ Control ซึ่งจะทำงานตรงข้ามกับ MouseEnter
MouseMove	ทำงานเมื่อเลื่อนเมาส์ไปมาบน Control
MouseDown	ทำงานเมื่อคลิกที่ปุ่มด้านซ้ายของเมาส์
MouseUp	ทำงานตรงข้ามกับ MouseDown เมื่อปล่อยปุ่มที่คลิกด้านซ้ายของเมาส์
MouseHover	ทำงานเมื่อเลื่อนเมาส์ไปวางอยู่บน Control นั้นๆ (นานชั่วกระพริบหนึ่ง) เพียงครั้งเดียว และจะทำงานอีกครั้งเมื่อเลื่อนเมาส์ออกจาก Control นั้น แล้วเข้าไปใน Control นั้นใหม่ (ถ้านานชั่วกระพริบหนึ่ง)
Click	ทำงานเมื่อคลิกและปล่อยปุ่มด้านซ้ายของเมาส์บน Control เดียวกัน เวลาเดียวกัน
DoubleClick	ทำงานเมื่อคลิกเมาส์ด้านซ้ายติดต่อกัน 2 ครั้ง
GotFocus	ทำงานเมื่อ Control ได้รับการ Focus
KeyDown	ทำงานเมื่อกดแป้นพิมพ์
KeyPress	เกิดขึ้นเมื่อกดแป้นพิมพ์ แล้วจะแปลรหัสแป้นพิมพ์เป็นรหัส ASCII

ชื่อ Event	การใช้งาน
KeyUp	ทำงานเมื่อปล่อยแป้นพิมพ์
Leave	ทำงานเมื่อเลิก Focus ที่ Control
Resize	เกิดขึ้นเมื่อปรับขนาด Form
PropertyChanged	ทำงานเมื่อมีการเปลี่ยนแปลงค่า Property ของ Control

หมายเหตุ: Win32 API คือ Function ควบคุมการทำงานของระบบปฏิบัติการ Win32 อยู่ในรูปของไฟล์ “.dll” ซึ่งถูกเรียกใช้ในขณะที่ระบบปฏิบัติการกำลังทำงานอยู่ สามารถนำมาใช้พัฒนาโปรแกรมให้มีความสามารถเหมือนกันระบบปฏิบัติการได้

2. Class “RichControl”

Class “RichControl” สืบทอดคุณสมบัติจาก Class “Control” โดยมีการใช้งานเหมือนกับ Control ทั่วไป ใน RichControl มี Member ที่น่าสนใจคือ Property ที่เกี่ยวกับ Font, ForeColor, BackColor และ BackGroundImage รวมถึง Property เพื่อใช้ในการจัด Layout เช่น Dock และ Anchor โดยมี Member ที่สำคัญใน Class “RichControl” ดังนี้

ตารางที่ 2.4 แสดง Property ของ Class “RichControl”

ชื่อ Property	การใช้งาน
AllowDrop	กำหนดให้สามารถใช้ Event ที่เกี่ยวกับการ Drag และ Drop ได้เมื่อกำหนดค่าเป็น True
Anchor	ใช้กำหนดขอบเขตของ Control ซึ่งทำหน้าที่ปรับความห่างของขอบหน้าต่างกับตัวคอนโทรลให้ได้ระยะเดิมเสมอ
Dock	ใช้จัด Layout และตำแหน่งของ Control บน Form ซึ่งสามารถขยายหรือลดขนาด Form ได้ โดยที่ Control ยังคงรักษาคำแหน่งเดิมไว้
BackColor	ใช้กำหนดสีฉากหลัง
Font	ใช้กำหนดรูปแบบตัวอักษร
ForeColor	ใช้กำหนดสีตัวอักษร

ตารางที่ 2.5 แสดง Method ของ Class “RichControl”

ชื่อ Method	การใช้งาน
DoDragDrop	ทำงานเมื่อเริ่ม Drag เมาส์

ตารางที่ 2.6 แสดง Event ของ Class “RichControl”

ชื่อ Event	การใช้งาน
DragDrop, DragEnter, DragLeave, DragOver	ทำงานเมื่อเกิดเหตุการณ์ที่เกี่ยวกับการ DragDrop
Paint	เกิดขึ้นเมื่อมีการวาด Control เดิมซ้ำบนหน้าจอ

Control ส่วนใหญ่ที่อยู่ใน NameSpace “WinForms” จะสืบทอดคุณสมบัติมาจาก Class “RichControl”

3. Class “ScrollableControl”

เพิ่มคุณสมบัติการเลื่อนข้อความ (Scrolling) ที่อยู่ในพื้นที่ของหน้าต่าง ปกติแล้วจะไม่สามารถใช้งาน Class นี้ได้โดยตรง แต่จะกระทำโดยผ่านทาง Member ของ Class ที่สืบทอดสมบัติต่อจาก Class นี้ โดยมี Member ที่สำคัญใน Class “ScrollableControl” ดังนี้

ตารางที่ 2.7 แสดง Property ของ Class “ScrollableControl”

ชื่อ Property	การใช้งาน
AutoScroll	ใช้กำหนดเพื่อควบคุมการเปิด/ปิด Scrolling และควบคุมการแสดงผลของ Scroll Bar เมื่อกำหนดค่าเป็น True
AutoScrollMinSize	ใช้กำหนดค่าต่ำสุดของ AutoScroll เพื่อให้ Scroll Bar ทำงาน เมื่อขนาดของ Control มีขนาดเล็กกว่าค่าที่ระบุไว้ หรือ Control อยู่นอกพื้นที่กรอบของหน้าต่าง
AutoScrollPosition	ใช้กำหนดตำแหน่งในแกน X และแกน Y ของ AutoScroll
Vscroll	ใช้กำหนดให้แสดง Scroll Bar ในแนวตั้ง
Hscroll	ใช้กำหนดให้แสดง Scroll Bar ในแนวนอน

4. Class “ContainerControl”

เป็น Class สืบทอดมาจาก Class “ScrollableControl” แต่เพิ่มความสามารถช่วยจัดการกับ Control ในส่วนของการ Focus และการใช้ปุ่ม Tab จาก Control หนึ่งไปยัง Control หนึ่งโดยใน Class “ContainerControl” มี Member ที่สำคัญดังนี้

ตารางที่ 2.8 แสดง Property ของ Class “ContainerControl”

ชื่อ Property	การใช้งาน
ActiveControl	ใช้ Focus ไปที่ Control ปัจจุบัน
ParentForm	ใช้กำหนดให้เป็น Form หลัก ซึ่งมี Form หรือ Control ต่างๆ บรรจุอยู่ภายใน Form นี้
ActiveControl	แสดง Control โดยระบุค่า Parameter เพื่อให้ Control ทำงาน
Validate	ตรวจสอบเหตุการณ์สุดท้ายของ Control (โดยส่วนมากแล้วการเปลี่ยนแปลงของ Control จะไม่ถูกตรวจสอบในทันที) จนถึงเหตุการณ์แรก แต่จะไม่รวม Control ปัจจุบัน

5. Class “Form”

เป็น Class ที่ใช้งานมากที่สุด โดยสืบทอดคุณสมบัติ (Property, Method และ Event) จาก Class ที่กล่าวมาทั้งหมดข้างต้นและเพิ่มความสามารถในการแสดงแถบข้อความ รวมทั้งระบบของเมนู ดังนั้น Form ที่อยู่บน Class จะคล้ายกับ Windows Form ทั่วไป แต่ Class “Form” จะสามารถสร้างหน้าต่างที่ไม่มีขอบ หรือ Form ลอยหรือ Modal Form เช่น Dialog Box นอกจากนี้ใน Class “Form” ยังสามารถกำหนดค่า Default ที่ทำงานอยู่บน Form โดยการกดปุ่ม Enter เพื่อให้ทำงาน และกดปุ่ม Esc เพื่อยกเลิก

ใน Class “Form” มี Member ที่สำคัญดังนี้

ตารางที่ 2.9 แสดง Property ของ Class “Form”

ชื่อ Property	การใช้งาน
AcceptButton	เป็น Object ของ Button Control ทำงานเมื่อกดปุ่ม Enter โดยสามารถใช้ได้กับ Control ประเภท Button เท่านั้น
ActiveMDIChild	ใช้กำหนดให้แสดงหน้าต่าง MDI Child ใน Form ที่เป็น MDI (Multiple Document Interface) Form
CancelButton	เป็น Object ของ Button Control ทำงานเมื่อกดปุ่ม Esc โดยสามารถใช้ได้กับ Control ประเภท Button เท่านั้น
ClientButton	กำหนดขนาดพื้นที่การใช้งานบน Form (ไม่รวมพื้นที่ของ Border, Title Bar และอื่นๆ)
DesktopBounds, DesktopLocation	เป็น Property ที่กำหนดขนาด และ Location ของ Form บนจอภาพ
DialogResult	แสดงค่าผลลัพธ์ที่ได้จากการคลิกปุ่มใน Dialog Box

ชื่อ Property	การใช้งาน
HelpButton	กำหนดให้แสดงปุ่ม Help บน Title Bar ของ Form
IsMDIChild	ตั้งค่าให้ Form เป็น MDIChild
IsMDIContainer	ตั้งค่าให้ Form เป็น MDIParent ซึ่งใช้เก็บ Form ประเภท MDIForm
Menu	กำหนดให้เป็นเมนูหลักของ Form ซึ่งจะปรากฏอยู่ด้านล่างของ Title Bar
MDIChildren	กำหนดค่าให้อ้างถึง Array ของ MDIChild Form ที่แสดงอยู่ใน MDI Container
MDIParent	กำหนดค่า MDIParent ให้กับ MDIChild Form
MaximizeBox	เป็นการกำหนดให้แสดงปุ่มขยายขนาดบน Title Bar ของ Form
MinimizeBox	เป็นการกำหนดให้แสดงปุ่มย่อขนาดบน Title Bar
Modal	เป็น Property สำหรับอ่านค่าอย่างเดียว ซึ่งจะแสดง Form แบบ Modal
OwnedForm, Owner	เป็น Property ที่ใช้ตั้งค่าการขึ้นต่อกันระหว่าง Form เช่น เมื่อปิด Form แรก Form ที่ขึ้นอยู่กับการจะถูกลบปิดโดยอัตโนมัติ
BorderStyle	ใช้กำหนดเส้นขอบให้กับ Form
ControlBox	เป็นการกำหนดให้แสดงปุ่มปิดบน Title Bar ของ Form
Icon	ใช้กำหนดให้แสดง Icon บน Title Bar ของ Form
KeyPreview	ใช้กำหนดให้ Form รับค่าจากการกดแป้นพิมพ์
ShowInTaskBar	ใช้กำหนดให้ Form ปรากฏอยู่บน TaskBar
Size	ใช้กำหนดขนาดของ Form
StartPosition	ใช้กำหนดตำแหน่งแรกของ Form เมื่อเริ่มทำงาน เช่น กำหนดให้ Form ปรากฏขึ้นกลางหน้าจอ
WindowState	ใช้กำหนดลักษณะของฟอร์มเมื่อเปิดฟอร์มนั้น เช่น กำหนดให้ Form ใหญ่เต็มหน้าจอ

ตารางที่ 2.10 แสดง Method ของ Class “Form”

ชื่อ Method	การใช้งาน
Close	ทำหน้าที่ปิด Form
ShowDialog	เป็นการแสดง Form แบบ Modal Dialog Box ซึ่งหน้าต่างที่ถูกเรียกขึ้นมาจะอยู่ด้านบนเสมอ โดยไม่สามารถทำงานที่หน้าต่างอื่นได้ ต้องปิดหน้าต่างนี้ก่อนจึงจะไปใช้งานหน้าต่างอื่นได้
Activate	ใช้ Focus ไปที่ Form
Show	แสดงให้เห็น Form

ตารางที่ 2.11 แสดง Event ของ Class “Form”

ชื่อ Event	การใช้งาน
Closed, Closing	จะทำงานเมื่อ Form ถูกปิด โดย Event “Closing” จะเหมือนกับ Form QueryUnload ส่วน Event “Closed” จะเหมือนกับ Form Unload ซึ่งสามารถเทียบได้กับการใช้ Dispose ใน Constructor
MDIChildActive	ใช้สำหรับ Form ที่ถูกตั้งค่าเป็น MDI และจะทำงานเมื่อมีการเรียกใช้ MDIChildForm
Activated	เกิดขึ้นเมื่อ Form ทำงานอยู่ในส่วนโค้ดของโปรแกรม

Member ของ Class “Form” ถูกใช้งานมากที่สุด และทุก Form ที่สร้างขึ้นจะมาสืบทอดคุณสมบัติจาก Class นี้

6. Class “UserControl”

Class “UserControl” ของ Visual Basic 2005 เป็นการสร้างหน้าจอที่มีลักษณะคล้าย Form เพื่อวาง Control เรียกว่า “UserControl” และสามารถนำ UserControl ที่สร้างขึ้นใหม่ไปใช้งานใน Toolbox โดยการ Compile เป็น .dll สำหรับ Property ของ UserControl จะมีคุณสมบัติเหมือนกับ Class ที่ได้รับการถ่ายทอดมา

7. ความสอดคล้องกันของ Windows Form และ Web Form

ฟังก์ชันการใช้งาน และชื่อที่ใช้งานของ Windows Form เหมือนกับใน Web Form ซึ่งแม้ว่าใน Windows Form จะไม่มี Class “UI” แต่ชื่อ Class ที่อยู่ใน Namespace จะมีชื่อเหมือนกัน เช่น ใน Windows Form และ Web Form มี Class “Button” ซึ่งเป็นชื่อเดียวกัน และมี Property “Text” และ Event “OnClick” เหมือนกัน รวมทั้งใน Windows Form และ Web Form มี Class “UserControl” เหมือนกัน