

## บทที่ 2

# ความรู้เบื้องต้นเกี่ยวกับบอร์แลนด์เดลไฟ

### 2.1 ความนำ

เดลไฟเป็นเครื่องมือที่ใช้พัฒนาโปรแกรมบนระบบปฏิบัติการวินโดวส์ โดยใช้โครงสร้างภาษาปาสคาลเป็นหลักในการพัฒนา ซึ่งเป็นโครงสร้างภาษาที่เขียนง่าย สามารถเห็นผลลัพธ์ที่จะได้ตั้งแต่ในขณะที่กำลังสร้าง และถูกใช้พัฒนาแอปพลิเคชันได้ทุกระดับทุกประเภทครบถ้วนในตัวเดียว โดยไม่ต้องใช้ภาษาอื่นเพิ่มเติมอีก เนื่องจากเดลไฟมีเครื่องมือช่วยเหลือในการนำคำสั่งต่าง ๆ ใช้งานได้สะดวกและรวดเร็ว

เดลไฟเป็นผลิตภัณฑ์ของบริษัทบอร์แลนด์ซอฟต์แวร์ (Borland Software Corporation) ซึ่งก่อตั้งขึ้นในปี ค.ศ. 1980 ต่อมาได้เปลี่ยนชื่อบริษัทเป็นอินไพรส์ (Inprise) ในช่วงปี ค.ศ. 1996-2000 เนื่องจากประสพภาวะขาดทุนทางธุรกิจ แต่อย่างไรก็ตาม ปัจจุบันยังคงใช้ชื่อบริษัทผู้ผลิตเป็นบอร์แลนด์เช่นเดิมจนถึงเวอร์ชัน 7 ปัจจุบันบริษัทเอ็มบาร์คาเดโร เทคโนโลยี (Embarcadero Technologies) ได้ซื้อหุ้น เดลไฟและพัฒนาไปถึง Delphi XE4 เดลไฟมีคุณสมบัติเป็นโปรแกรมภาษาภาพ (Visual Programming) ซึ่งหมายถึง การพัฒนาโดยเห็นผลที่เกิดขึ้นเมื่อรันโปรแกรมได้ตั้งแต่ในขณะที่กำลังสร้าง (What you see is what you get : WYSIWYG) โดยการนำชิ้นส่วนต่าง ๆ ที่ต้องการ ซึ่งต่อมาจะถูกเรียกว่า วัตถุ (Object) อาทิ ปุ่ม (Button), ป้ายข้อความ (Label), รูปภาพ (Image) เป็นต้น โดยเรียกแหล่งรวมวัตถุเหล่านั้นว่า องค์ประกอบ (Component) นำมาวางบนพื้นที่ที่ต้องการนำเสนอบนวินโดวส์ เรียกว่า ฟอรัม (Form) นอกจากนี้ ยังสามารถรายละเอียดของวัตถุได้ในคุณสมบัติของวัตถุ (Property) และสามารถกำหนดการทำงานของวัตถุได้ในพฤติกรรมของวัตถุ (Method) ดังนั้นจึงสรุปได้ว่า Borland Delphi ในหลักการพัฒนาโปรแกรมในเชิงวัตถุที่เรียกว่า การโปรแกรมเชิงวัตถุ (Object Oriented Programming : OOP)

## 2.2 พื้นฐานเดลไฟ

เดลไฟเป็นโปรแกรมที่ใช้การคอมไพล์เลอร์แปลภาษาโปรแกรมเป็นภาษาเครื่อง ซึ่งภายหลังการคอมไพล์โปรแกรมเราจะได้ไฟล์ที่สามารถใช้งานได้ที่ มีนามสกุล .EXE ในไฟล์นามสกุลดังกล่าวจะเก็บภาษาเครื่องไว้เมื่อมีการใช้งานจึงไม่ต้องผ่านขั้นตอนการแปลงภาษาอีก ทำให้สามารถทำงานได้เร็ว นอกจากนั้นความสามารถของเดลไฟที่ใช้ในการพัฒนาแอปพลิเคชันยังง่ายด้วยการมีคอมโพเนนท์ให้เลือกใช้เป็นจำนวนมาก โดยเดลไฟจัดเก็บคอมโพเนนท์เหล่านั้นอยู่ในห้องสมุด หรือ คอมโพเนนท์ไลบรารี (component library) ซึ่งจะกล่าวต่อไปในบทที่ 4 นอกจากนั้นเดลไฟยังมีวิซาร์ด (wizard) ที่ใช้ในการสร้างฟอร์มและออบเจ็คที่ต้องใช้ บ่อย ๆ หรือสร้างได้ยาก เช่น วิซาร์ดในการสร้างรายงานแบบต่าง ๆ หรือฟอร์มที่มีวัตถุประสงค์การใช้ต่างกัน เช่น About Box และ Dialog Box เป็นต้น ซึ่งเรียกอีกอย่างหนึ่งว่า เทมเพลต (template) สิ่งเหล่านี้ช่วยให้การสร้างแอปพลิเคชันง่ายขึ้น และรวดเร็วขึ้น

สำหรับการติดต่อกับฐานข้อมูลนั้น เดลไฟมีคอมโพเนนท์ที่สามารถเชื่อมต่อ เพื่อให้จัดการกับฐานข้อมูลทุกประเภท เช่น dBase, Paradox, Microsoft Access, SQL Server หรือ InterBase เป็นต้น นอกจากนี้ ตั้งแต่เดลไฟเวอร์ชัน 5 ขึ้นไป ยังเพิ่มคุณสมบัติในการติดต่อกับฐานข้อมูลด้วย ADO ซึ่งบริษัทไมโครซอฟต์ได้พัฒนาขึ้นมาใหม่เพื่อให้สามารถติดต่อกับแหล่งข้อมูลได้หลายประเภทโดยไม่ต้องใช้เครื่องของเดลไฟ Borland Database Engine (BDE)

เมื่อติดตั้งเดลไฟ มักได้กลุ่มแอปพลิเคชันที่ออบอร์แลนด์เดลไฟ อยู่ในเมนูโปรแกรมของระบบปฏิบัติการ ภายในกลุ่มจะประกอบด้วยเครื่องมือที่อำนวยความสะดวกในการบริหารจัดการ ดังนี้

Delphi X	โปรแกรมเดลไฟที่ใช้พัฒนาแอปพลิเคชัน
BDE Administrator	เครื่องมือที่ใช้กำหนดคุณสมบัติต่าง ๆ ในการติดต่อกับฐานข้อมูล ซึ่งการเชื่อมต่อแอปพลิเคชันเข้ากับฐานข้อมูลด้วย BDE ซึ่งต้องอาศัย BDE Administrator เป็นตัวช่วยควบคุมการ

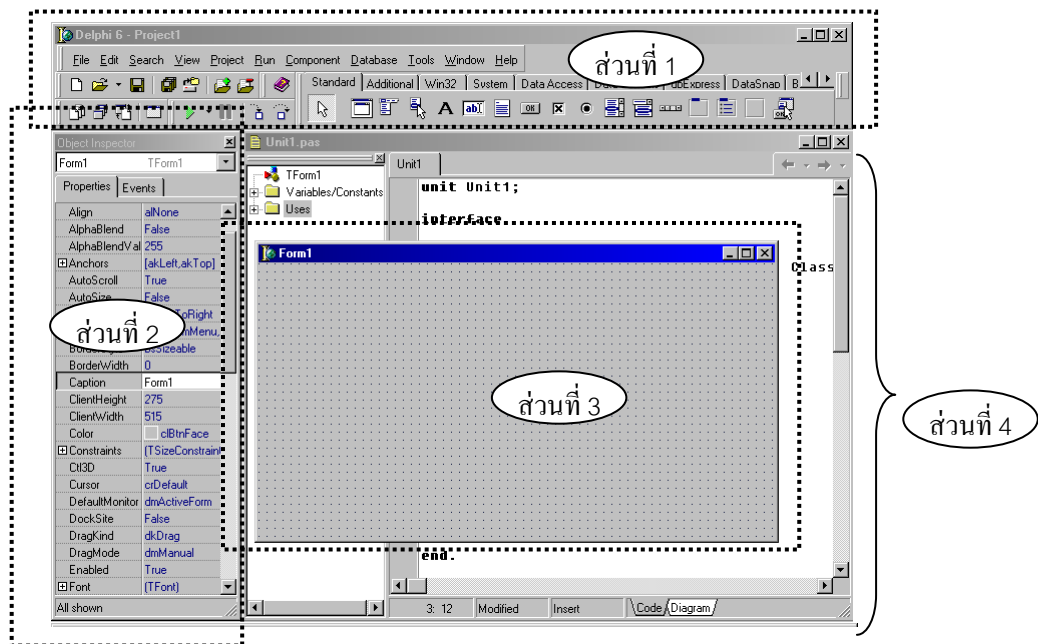
	ทำงานของ BDE เพื่อให้สามารถใช้งานฐานข้อมูลได้ตามต้องการ
Database Desktop	เครื่องมือที่ใช้ในการทำงานกับฐานข้อมูล เช่น สามารถฐานข้อมูล การตารางจัดเก็บข้อมูล รวมถึงการปรับปรุงข้อมูล ได้แก่ การลบข้อมูล แก้ไขข้อมูล เรียบดูข้อมูล หรือเพิ่มข้อมูล
Datapump	เครื่องมือที่ใช้ย้ายข้อมูลจากฐานข้อมูลหนึ่ง ไปฐานข้อมูลหนึ่ง
Database Explorer	เครื่องมือที่ใช้เรียกดูรายละเอียดในฐานข้อมูลทั้งโครงสร้างการออกแบบและรายการข้อมูลในตารางข้อมูล
SQL Monitor	เครื่องมือที่ใช้ในการติดต่อดูการติดต่อระหว่าง BDE และ Database Server ว่ามีการส่งคำสั่ง SQL อะไรไปบ้าง ซึ่งเครื่องมือนี้มีเฉพาะในเดสก์ท็อป แบบ Enterprise Edition เท่านั้น
Image Editor	เครื่องมือที่ใช้ในการตกแต่งภาพ
WinSight32	เครื่องมือที่ใช้ตรวจสอบการทำงานของโปรแกรมที่เราเขียนขึ้น ว่ามีการติดต่ออะไรกับระบบปฏิบัติการอยู่บ้าง
Help	ส่วนที่ใช้แสดงรายละเอียดในการใช้งานโปรแกรมเดสก์ท็อปทั้งหมด

## 2.3 สภาพแวดล้อมของเดสก์ท็อป

การสร้างโปรแกรมในเดสก์ท็อปจะมีรูปแบบที่เรียกว่า Integrated Development Environment (IDE) คือเป็นการสร้างโปรแกรมโดยการทำงานทุกอย่างสามารถทำได้จากที่แห่งเดียวกันโดยไม่จำเป็นต้องใช้แอปพลิเคชันตัวหนึ่งเขียนโค้ด และคอมไพล์ด้วยแอปพลิเคชันอีกตัวหนึ่ง หรือ Command Line ภายนอก ซึ่งทำให้การพัฒนาโปรแกรมทำได้สะดวกมากยิ่งขึ้น

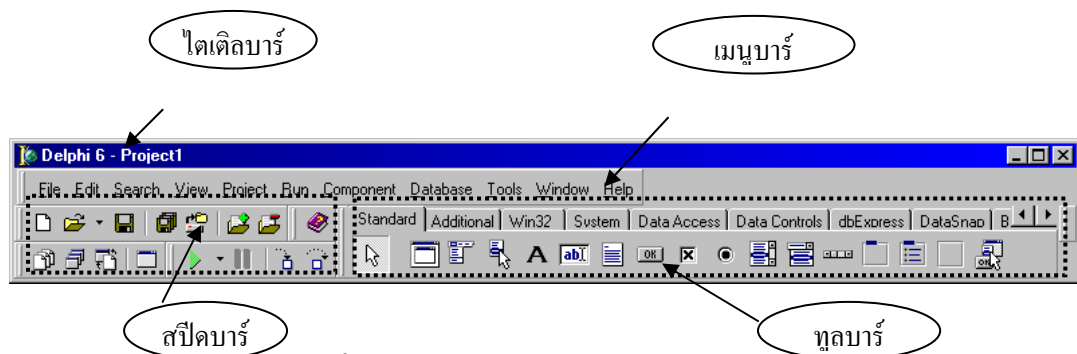
### 2.3.1 ส่วนประกอบภายในสภาพแวดล้อม

เมื่อเรียกใช้โปรแกรมเดสก์ท็อปแล้วจะเห็นหน้าต่างโปรแกรมซึ่งประกอบด้วยวินโดว์ต่าง ๆ ดังรูปที่ 2.1



รูปที่ 2.1 สภาพแวดล้อมของเดลไฟ

ส่วนที่ 1 วินโดว์หลัก (Main Window) วินโดว์หลักทำหน้าที่เป็นศูนย์กลางสำหรับควบคุมกระบวนการในการพัฒนาโปรแกรม ไม่ว่าจะเป็นการจัดการเกี่ยวกับไฟล์ภายใน แอปพลิเคชัน การคอมไพล์ การตรวจสอบข้อผิดพลาด เมื่อรันโปรแกรมเดลไฟจะเห็นวินโดว์หลักเป็นวินโดว์ที่อยู่บนสุด โดยภายในวินโดว์หลักจะแบ่งเป็น 4 ส่วน คือ ไตเติลบาร์ เมนูบาร์ สปีดบาร์ และ ทูลบาร์



รูปที่ 2.2 วินโดว์หลักของเดลไฟ

- ไตเติลบาร์ (Title Bar)

เป็นส่วนที่อยู่บนสุด แสดงชื่อโปรแกรมเดลไฟ และชื่อโปรเจ็ค (Project) ที่ทำงานอยู่ขณะนั้น

- **เมนูบาร์ (Menu Bar)**

เป็นส่วนประกอบที่จะพบได้ในแอปพลิเคชันบนวินโดวส์ทั่วไป โดยจะอยู่ใต้ไตเติลบาร์ ภายในเมนูบาร์จะแสดงเมนูคำสั่งของเดสก์ทอป ที่ใช้ทำงานทั้งหมด อาทิ เมนู File Edit Search Views Project เป็นต้น ซึ่งแต่ละเมนูจะมีคำสั่งย่อยภายในแสดงในลักษณะ DropDown เมื่อคลิกหัวข้อเมนู

- **สปีดบาร์ (Speed Bar)**

เป็นส่วนแสดงกลุ่มคำสั่งของเมนูบาร์ที่ถูกใช้บ่อย ๆ ด้วยการใช้นิ้วแทนรายการ เมื่อคลิกเมาส์ที่ปุ่มใดจะเหมือนกับการเลือกคำสั่งในเมนูบาร์ และเมื่อเลื่อนเมาส์ไปหยุดบนปุ่มใดจะเห็น กรอบข้อความปรากฏ (tooltips) แสดงชื่อเมนูและฟังก์ชันคีย์ที่เป็นคีย์ลัดสำหรับใช้แทนปุ่มนั้น

ปุ่มที่ควรทราบในการใช้งาน



New : ใช้ในกรณีต้องการสร้างโปรเจ็คใหม่



Open : ใช้ในกรณีเรียกโปรเจ็คเดิมเพื่อปรับปรุงแก้ไข



Save และ Save All : ใช้ในกรณีต้องการจัดเก็บงาน



View Unit : ใช้ในกรณีเลือกแสดง Dialog Code ของ Unit



View Form : ใช้ในกรณีเลือกแสดง Form ของ Unit



Toggle Form : ใช้ในกรณีแสดง Form ของ Unit ปัจจุบัน



New Form : ใช้ในกรณีสร้าง Form และ Unit ใหม่



Run : ใช้ในกรณีประมวลผลโปรเจ็คปัจจุบัน

- **ทูลบาร์ (Toolbar) หรือ คอมโพเนนต์พาเลตต์ (Component Palette)**

เป็นส่วนที่เครื่องมือที่ใช้สร้างและตกแต่งแอปพลิเคชัน โดยจัดกลุ่มเครื่องมือไว้ในแท็บต่าง ๆ อาทิ Standard, Additional, Win32, System เป็นต้น ซึ่งเมื่อคลิกแท็บเพื่อแสดงคอมโพเนนต์ในแต่ละกลุ่ม และเลื่อนเมาส์ไปหยุดในคอมโพเนนต์ตัวใดตัวหนึ่งจะปรากฏกรอบข้อความแสดงชื่อทูลดังกล่าวขึ้นทางจอภาพ

## ส่วนที่ 2 วินโดว์ออบเจ็คอินสเป็คเตอร์ (Object Inspector Window)

วินโดว์ออบเจ็คอินสเป็คเตอร์เป็นวินโดว์ซึ่งอยู่ที่วินโดว์หลักทางด้านซ้าย ใช้สำหรับแสดงและปรับแต่งค่าคุณสมบัติ (Property) และเหตุการณ์ (event) ของคอมโพเนนท์ ซึ่งคุณสมบัติที่ปรากฏในปัจจุบันขึ้นอยู่กับเรากำลังทำงานอยู่ที่คอมโพเนนท์ใด ดังเช่นรูปที่ 1 จากรูปจะเห็นว่าได้ไต่เตลบาร์ของวินโดว์ออบเจ็คอินสเป็คเตอร์ ใช้สำหรับแสดงชื่อ และชนิดของคอมโพเนนท์ที่ทำงานอยู่ขณะนั้น “Form1 : TForm1” แสดงว่าเรากำลังทำงานกับคอมโพเนนท์ชื่อ Form1 และมีชนิดคอมโพเนนท์เป็น TForm1 ส่วนถัดลงมาปรากฏงาน 2 แท็บ คือ Properties และ Events สำหรับการดำเนินงานดังนี้

### ■ คุณสมบัติ ( Properties )

ใช้กำหนดคุณสมบัติของคอมโพเนนท์ โดยภายในแท็บแบ่งออกเป็น 2 คอลัมน์ โดยคอลัมน์ด้านซ้ายแสดงชื่อของคุณสมบัติ เช่น Font, color, caption เป็นต้น ส่วนคอลัมน์ด้านขวาแสดงค่าของคุณสมบัตินั้นที่สามารถแก้ไขปรับปรุงได้

### ■ เหตุการณ์ ( Events )

ใช้กำหนดการทำงานเมื่อมีเหตุการณ์หรือการกระทำใด ๆ เกิดขึ้นกับคอมโพเนนท์ดังกล่าว ซึ่งอาจเกิดจากการกระทำของผู้ใช้ (จงใจให้เกิด) หรือตัวโปรแกรมเอง เช่น การใช้เมาส์คลิก การกดคีย์ Enter การแก้ไขเปลี่ยนแปลงค่า การเคลื่อนที่ของเมาส์ เป็นต้น

## ส่วนที่ 3 วินโดว์ฟอร์ม (Form Window) ฟอร์มเป็นวินโดว์ที่ใช้สำหรับ

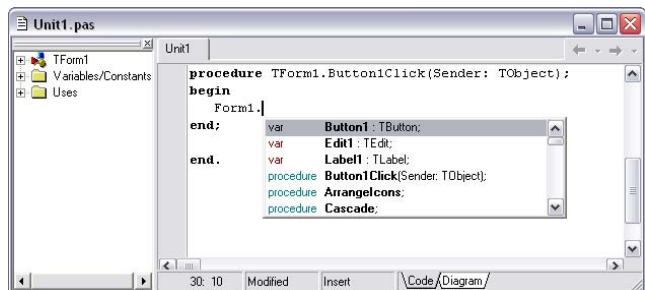
ออกแบบส่วนที่ติดต่อกับผู้ใช้ (Users Interface) โดยการนำคอมโพเนนท์ต่าง ๆ จาก Toolbar มาวางลงบนฟอร์ม ซึ่งการวางและปรับแต่งคุณสมบัติคอมโพเนนท์ดังกล่าว หากประมวลผลโปรแกรม (Run) ผลที่ได้รับทางจอภาพจะมีลักษณะเดียวกัน อย่างไรก็ตาม โปรแกรมหนึ่ง ๆ ของเดลไฟ มักมีการสร้าง Form มากกว่า 1 ฟอร์มเสมอ ดังนั้น Delphi จึงกำหนดให้เริ่มต้นการแสดงที่ Form แรกเสมอ หากต้องการแสดง Form อื่น ๆ ต้องเขียนโค้ดสั่งให้แสดง Form ดังกล่าวเองผ่าน Events

**ส่วนที่ 4** วินโดว์โค้ดเอดิเตอร์ (Code Editor Window) วินโดว์โค้ดเอดิเตอร์มีไว้สำหรับเขียนโปรแกรม โดยอาจจะถูกบังคับให้แสดงอยู่ได้ฟอร์ม เมื่อเลื่อนวินโดว์ฟอร์มออกไปหรือคลิกปุ่ม Minimize ฟอร์ม จะปรากฏวินโดว์โค้ดเอดิเตอร์ขึ้น ซึ่งการเขียนโค้ดสำหรับควบคุมการทำงานของโปรเจ็คและฟอร์ม สามารถเขียนได้ผ่าน Events ของคอมโพเนนท์ หรือเขียนขึ้นใหม่ตามความต้องการในรูปโปรแกรมย่อย (Sub-program) ที่มี 2 รูปแบบ Procedure หรือ Function ซึ่งจะกล่าวต่อไปในหัวข้อที่ 4

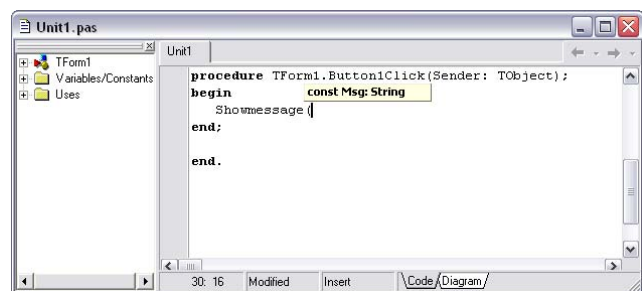
### 2.3.2 เครื่องมือหยั่งรู้ (Code Insight)

เพื่อลดข้อผิดพลาดและเพิ่มความสะดวกรวดเร็วในการเขียนโปรแกรม เดลไฟเตรียมเครื่องมือที่เรียกว่า Code Insight ช่วยอำนวยความสะดวกในการเขียนโค้ดประกอบด้วย

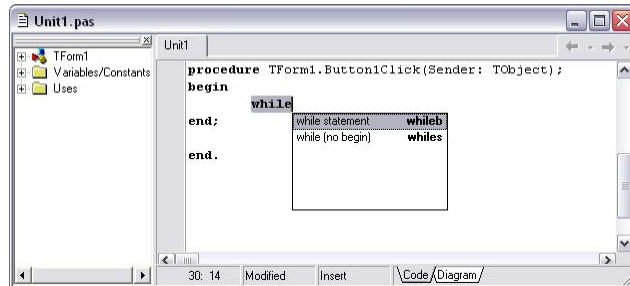
**Code Completion** ใช้สำหรับแสดงคุณสมบัติ พหุติกรรม หรือเหตุการณ์ของออบเจ็คให้เลือก โดยค่าเหล่านี้จะแสดงขึ้นมาหลังจากที่คีย์ชื่อออบเจ็คแล้วตามด้วยจุด เช่น ถ้าโปรแกรมมีออบเจ็คชื่อ Form1 ซึ่งเป็นออบเจ็คประเภท TForm เมื่อคีย์คำว่า Form1. แล้วรอสักครู่จะปรากฏรายการพหุติกรรม และคุณสมบัติของ Form1 ขึ้นมาให้เลือกในลักษณะกล่องรายการ เมื่อเลือกแล้วกดคีย์ Enter ค่าที่เลือกจะถูกเขียนลงโค้ดเอดิเตอร์ให้ทันที



**Code Parameter** ใช้สำหรับแสดงพารามิเตอร์ของโพรซีเยอร์หรือฟังก์ชันเมื่อคีย์ชื่อโพรซีเยอร์หรือฟังก์ชันตามด้วยวงเล็บเปิด จะมีข้อความเล็กๆ ปรากฏชื่อพารามิเตอร์พร้อมชนิดตัวแปรตรงตำแหน่งของเคเซอร์ เพื่อช่วยให้ผู้ใช้ใส่ค่าพารามิเตอร์ได้อย่างถูกต้อง



**Code Templates** ใช้สำหรับเรียกใช้คำสั่งจากเทมเพลต ซึ่งเดลไฟจะเก็บโครงสร้างของคำสั่งต่าง ๆ ไว้ โดยที่เราสามารถเพิ่มเติมหรือลบคำสั่งในเทมเพลตได้ การใช้งานก็เพียงแค่คียตัวอักษรของคำสั่งจากนั้นกดคีย Ctrl+J จะปรากฏรายการของคำสั่งในโค้ดเทมเพลตให้เลือก เมื่อกดคีย Enter โครงสร้างคำสั่งดังกล่าวจะถูกเขียนเป็นโครงร่างในโค้ดเอดิเตอร์ทันที

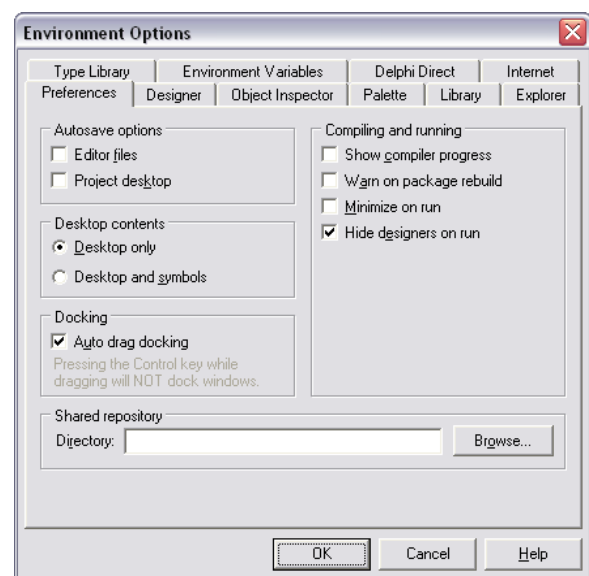


### 2.3.3 การกำหนดสภาพแวดล้อมในการพัฒนา

สภาพแวดล้อมของเดลไฟสามารถยืดหยุ่นต่อการใช้งานมาก เราสามารถปรับแต่งเพื่อสะดวกแก่การใช้งานก่อนจะเริ่มสร้างแอปพลิเคชัน โดยมีหัวข้อที่ควรให้ความสำคัญ 2 หัวข้อ ได้แก่

**Environment Options** การเปลี่ยนแปลงสภาพแวดล้อมในการทำงานปกติ โดยเลือกจากเมนู Tools > Environment Options จะปรากฏไดอะล็อกบ็อกซ์ Environment Options สำหรับการปรับแต่งสภาพแวดล้อมทั่วไปจะรวมกันอยู่ในหัวข้อ Preferences โดยในหน้านี้จะมีค่าให้กำหนดมากมาย แต่มีรายการที่ควรพิจารณาหลัก ๆ คือ

- **Autosave Options** กำหนดให้บันทึกข้อมูลโดยอัตโนมัติ ซึ่งจะมีค่าให้กำหนด 2 ค่า คือ Editor Files คือการบันทึกโปรเจ็คให้โดยอัตโนมัติทุกครั้งก่อนจะรัน และ Project Desktop คือการบันทึกขนาดและตำแหน่งของวินโดว์ต่าง ๆ รวมทั้งรายการไฟล์ที่เปิดอยู่ก่อนที่จะออกจากโปรแกรมเดลไฟ
- **Form Designer** กำหนดรูปแบบการจัดเก็บข้อมูลฟอร์ม ซึ่งมี 2 รูปแบบคือ New Form as Text หมายถึงกำหนดให้บันทึกหน้าต่างของฟอร์มใหม่ที่

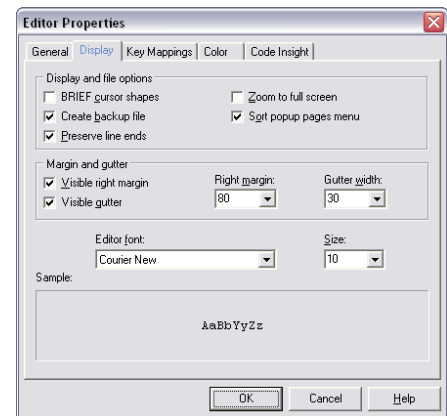




สร้างขึ้น (ไฟล์ .dfm) ในแบบข้อความแทนแบบไบนารี และ Auto Create Forms หมายถึง กำหนดให้เพิ่มคำสั่งสร้างฟอร์มไว้ในโปรเจ็คเลยหรือใหม่ หรือจะต้องเขียนโค้ดเพื่อสร้างฟอร์มเอง

**Editor Options** การปรับเปลี่ยนสภาพแวดล้อมด้านเอดิเตอร์ที่ใช้สำหรับ

เขียนชุดคำสั่ง เช่น การกำหนดระยะช่องไฟ การปรับขนาดตัวหนังสือ รูปแบบตัวหนังสือ สีที่แสดง เป็นต้น โดยเลือกจากเมนู Tools > Editor Options จะปรากฏไดอะล็อกบ็อกซ์ที่มีหัวข้อดังต่อไปนี้



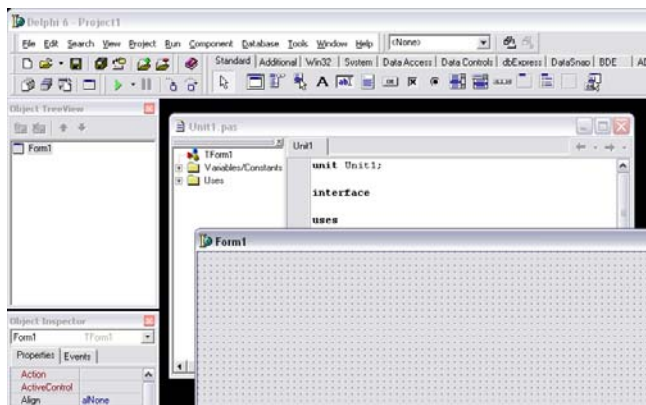
- General กำหนดรูปแบบของโค้ดเอดิเตอร์ เช่น การกำหนดระยะห่างของช่องไฟที่ได้จากการกดคีย์ Tab เป็นต้น
- Display กำหนดรูปแบบของอักษรและสีที่ต้องการใช้ในวินโดว์เอดิเตอร์
- Key Mappings กำหนดคีย์ต่าง ๆ ที่ใช้ในการแก้ไขโค้ด
- Color กำหนดสีของคำสั่งต่าง ๆ ในวินโดว์โค้ดเอดิเตอร์
- Code Insight กำหนดคุณสมบัติของ Code Insight ที่เกี่ยวกับ Code Completion, Code Parameters และ Code Templates ดังที่กล่าวไว้แล้ว

## 2.4 ขั้นตอนการพัฒนาโปรแกรม

การพัฒนาแอปพลิเคชันด้วยเดลไฟเป็นสิ่งทำได้ง่าย เนื่องจากเดลไฟเป็นโปรแกรมเชิงภาพที่มีสภาพแวดล้อมและคอมโพเนนท์ที่อำนวยความสะดวกต่อผู้ใช้ ทำให้การพัฒนาไม่ยุ่งยากซับซ้อนโดยมีขั้นตอนหลัก ๆ ตามลำดับดังนี้ สร้างโปรเจ็คใหม่ วางคอมโพเนนท์ลงบนฟอร์ม กำหนดค่าคุณสมบัติและพฤติกรรม กำหนดเหตุการณ์และ/หรือเขียนคำสั่งควบคุมการทำงาน บันทึกโปรเจ็ค และ ประมวลผลโปรแกรม จนกระทั่งปิดโปรแกรมและสามารถเปิดขึ้นมาใหม่เพื่อดำเนินการพัฒนาต่อเนื่อง ซึ่งขั้นตอนหลักที่กล่าวข้างต้น เราสามารถอธิบายโดยพัฒนาโปรแกรมตัวอย่างการทักทายได้ตามลำดับดังนี้

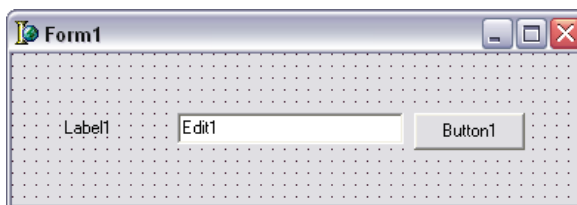
### 2.4.1 สร้างโปรเจ็คใหม่

เมื่อเราต้องการสร้างแอปพลิเคชันใหม่ด้วยการเลือกหัวข้อ New Application ในเมนูบาร์แล้ว เดลไฟจะจัดการสร้างโปรเจ็ค (Project) ให้หนึ่งโปรเจ็ค ซึ่งประกอบด้วยฟอร์มหนึ่งฟอร์ม และยูนิตหนึ่งยูนิต ดังนั้นโปรเจ็คหนึ่งก็คือแอปพลิเคชันนั่นเอง โดยปกติแล้วเมื่อมีการสร้างโปรเจ็คใหม่ทุกครั้ง เดลไฟจะตั้งชื่อเป็นมาตรฐานไว้เสมอได้แก่โปรเจ็คชื่อ Project1 ฟอร์มชื่อ Form1 และยูนิตชื่อ Unit1



### 2.4.2 วางคอมโพเนนต์ลงบนฟอร์ม

เมื่อเราสร้างโปรเจ็คใหม่ขึ้นมาแล้วในขั้นตอนต่อมาคือการสร้างส่วนที่ติดต่อกับผู้ใช้ โดยการวางคอมโพเนนต์ลงบนฟอร์มตามต้องการ ซึ่งขึ้นอยู่กับการออกแบบและความจำเป็นในการทำงาน ในที่นี้เราจะสร้างการทักทายจึงต้องใช้คอมโพเนนต์ของบางกลุ่มตามลำดับดังนี้



- (1) คลิกกลุ่มคอมโพเนนต์ชื่อ Standard
- (2) เลือกคอมโพเนนต์ Label
- (3) จากนั้นคลิกลงบนฟอร์ม ณ ตำแหน่งที่ต้องการวาง
- (4) เลือกคอมโพเนนต์ Edit
- (5) จากนั้นคลิกลงบนฟอร์ม ณ ตำแหน่งที่ต้องการวาง
- (6) เลือกคอมโพเนนต์ Button

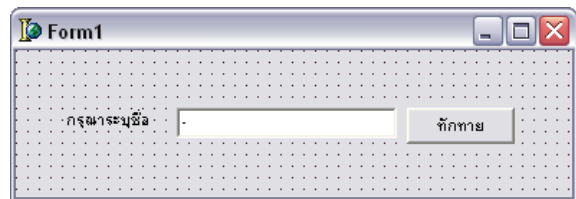
(7) จากนั้นคลิกลงบนฟอร์ม ณ ตำแหน่งที่ต้องการวาง

(8) ปรับแต่งตำแหน่งของคอมโพเนนต์ตามต้องการได้ในภายหลัง

ซึ่งการวางคอมโพเนนต์บนฟอร์มเป็นการสร้าง Instance ของออบเจ็คนั้น เช่น คอมโพเนนต์ Edit เมื่อนำมาวางบนฟอร์มก็จะได้ Instance ของออบเจ็คชนิด Edit เป็นต้น ซึ่งกรณีที่มีการใช้คอมโพเนนต์ชนิดเดียวกันมากกว่า 1 ตัวก็就会有การสร้าง Instance ของออบเจ็คขึ้นใหม่ตามจำนวนที่ใช้จริง

### 2.4.3 กำหนดค่าคุณสมบัติและพฤติกรรม

เมื่อเรานำคอมโพเนนต์ต่าง ๆ มาวางลงบนฟอร์มแล้ว ต่อไปจะต้องการกำหนดคุณสมบัติให้กับคอมโพเนนต์แต่ละตัวด้วย เพื่อให้ส่วนที่ติดต่อกับผู้ใช้มีลักษณะสมบูรณ์ขึ้น ซึ่งการกำหนดคุณสมบัติของคอมโพเนนต์ก็คือการให้ค่าคุณสมบัติแก่ออบเจ็คซึ่งแสดงอยู่ในวินโดว์ออบเจ็คอินสเป็คเตอร์นั่นเอง เราจะกำหนดค่าคุณสมบัติให้กับออบเจ็คที่อยู่บน Form1 ดังนี้



(1) คลิกที่คอมโพเนนต์ Label ที่อยู่บนฟอร์ม

(2) พิจารณาในวินโดว์ออบเจ็คอินสเป็คเตอร์ปรากฏชื่อคอมโพเนนต์ว่า Label1.TLabel เพื่อกำหนดค่าคุณสมบัติ Caption

(3) ลบค่าคุณสมบัติปัจจุบันออกและระบุค่าใหม่ว่า “กรุณาระบุชื่อ”

(4) คลิกที่คอมโพเนนต์ Edit ที่อยู่บนฟอร์ม

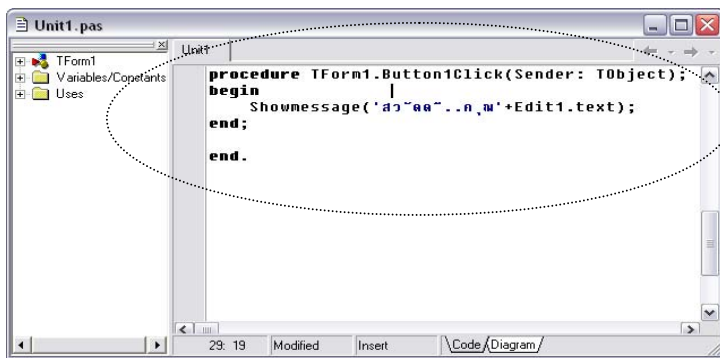
(5) พิจารณาในวินโดว์ออบเจ็คอินสเป็คเตอร์ปรากฏชื่อคอมโพเนนต์ว่า Edit1.TEdit เพื่อกำหนดค่าคุณสมบัติ Text ให้เป็น “-”

(6) คลิกที่คอมโพเนนต์ Button ที่อยู่บนฟอร์ม

(7) พิจารณาในวินโดว์ออบเจ็คอินสเป็คเตอร์ปรากฏชื่อคอมโพเนนต์ว่า Button1.TButton เพื่อกำหนดค่าคุณสมบัติ Caption ให้เป็น “พักหายใจ”

#### 2.4.4 กำหนดค่าเหตุการณ์และ/หรือเขียนคำสั่งควบคุมการทำงาน

เมื่อเราได้หน้าต่างของแอปพลิเคชันแล้ว ต่อไปจะเป็นการเพิ่มคำสั่งควบคุมการทำงานให้แอปพลิเคชันให้สามารถโต้ตอบกับผู้ใช้งานได้ โดยกำหนดการทำงานไว้ว่า ถ้าผู้ใช้ป้อนชื่อใน Edit1 บนฟอร์มแล้วคลิกปุ่ม ทักทาย แล้วให้ปรากฏข้อความว่า “สวัสดี..คุณ(ชื่อที่ป้อน)” บนจอภาพ ซึ่งเราสามารถเขียนคำสั่งควบคุมการทำงานได้ดังนี้



- (1) คลิกคอมโพเนนต์ Button
- (2) พิจารณาในวินโดว์ออกแบบ  
เจ็คอินสเป็คเตอร์ปรากฏ  
ชื่อคอมโพเนนต์ว่า  
Button1.TButton จาก  
นั้นคลิกที่แท็บ Events

เพื่อกำหนดเหตุการณ์

- (3) ดับเบิลคลิกที่ช่องว่างด้านขวาของเหตุการณ์ OnClick เพื่อให้ปรากฏชื่อ Button1Click และเดลไฟจะจัดการสร้างโพรซีเยอร์ดังกล่าวไว้ในวินโดว์โค้ดเอดิเตอร์เพื่อให้เราเขียนคำสั่งควบคุมลงไประหว่าง Begin และ End;
- (4) เขียนคำสั่งระหว่าง Begin End; ดังนี้

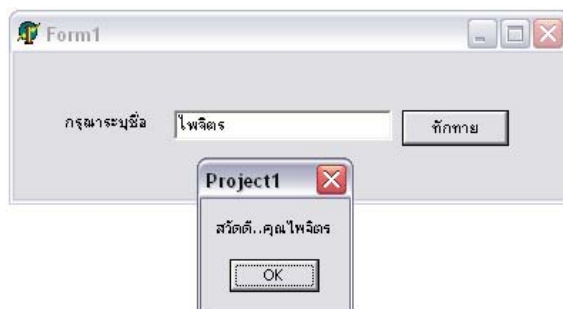
#### 2.4.5 บันทึกโปรเจ็ค

การบันทึกโปรเจ็คเป็นการจัดเก็บข้อมูลที่เราสร้างขึ้นมา ซึ่งควรจะบันทึกโปรเจ็คก่อนประมวลผลโปรแกรมเสมอ เพราะหากเกิดข้อผิดพลาดจะทำให้ไม่สามารถทำงานกับเดลไฟได้ต่อไป ในการจัดเก็บข้อมูลของโปรเจ็คหนึ่ง ๆ แบ่งออกเป็นหลายไฟล์ ซึ่งเดลไฟจะกำหนดการจัดเก็บหลัก ๆ ตามชื่อไฟล์ได้แก่ ไฟล์ที่บันทึกข้อมูลของยูนิตคือ ชื่อยูนิต(.pas) ไฟล์ที่บันทึกรายละเอียดของโปรเจ็คคือ ชื่อโปรเจ็ค(.dpr) ซึ่งชื่อของไฟล์ต้องไม่เหมือนกัน ดังขั้นตอนต่อไป

- (1) คลิกเมนู File > Save All
- (2) เลือกไดเรกทอรีที่ต้องการบันทึกโปรเจ็ค
- (3) ใส่ชื่อยูนิทในช่อง File name โดยปกติเป็น Unit1 คลิกปุ่ม Save
- (4) กรณีที่มีการสร้างงานหลายยูนิทจะปรากฏให้ใส่ชื่อยูนิทตามจำนวน
- (5) ใส่ชื่อโปรเจ็คในช่อง File name โดยปกติเป็น Project1 คลิกปุ่ม Save

#### 2.4.6 ประมวลผลโปรแกรม

การคอมไพล์เป็นการตรวจสอบว่าคำสั่งต่างๆ ที่เราเขียนขึ้นนั้นถูกต้องหรือไม่ และสร้างไบนารีไฟล์ (.dcu) ขึ้นมา ซึ่งกรณีที่มีการบันทึกโปรแกรมก่อนประมวลผลโปรแกรม และโปรแกรมไม่พบข้อผิดพลาดใด ๆ เดลไฟจะสร้างไฟล์ .exe ที่มีชื่อเดียวกับโปรเจ็ค วิธีการคอมไพล์โปรแกรมสามารถทำได้ดังนี้



- (1) คลิกเมนู Project > Compile (ปรากฏชื่อไฟล์โปรเจ็คปัจจุบัน) หรือ
- (2) คลิกเมนู Run > Run หรือ
- (3) คลิกปุ่ม Run ใน สปีดบาร์

#### 2.4.7 การปิด/เปิดโปรแกรม

เมื่อเสร็จสิ้นการพัฒนาสามารถปิดไฟล์/โปรเจ็คโดยมีทางเลือก 2 ทาง คือ

- (1) คลิกเมนู File > Close เพื่อปิดเฉพาะไฟล์ที่แสดงอยู่ในขณะนั้น หรือ
- (2) คลิกเมนู File > Close All เพื่อปิดทุกไฟล์ที่ทำงานอยู่ในขณะนั้น รวมทั้งโปรเจ็คด้วย

ส่วนการเปิดโปรเจ็คเมื่อต้องการพัฒนาต่อหรือปรับปรุงแก้ไข มีคำสั่งให้เลือก 3 ทาง คือ

- (1) คลิกเมนู File > Open เพื่อเปิดไฟล์ที่เก็บไว้ขึ้นมาทำงาน ซึ่งจะสามารถเปิดได้ทั้งโปรเจ็ค (.dpr) ฟอรัม (.dfm) และยูนิต (.pas)
- (2) คลิกเมนู File > Open Project เพื่อเปิดโปรเจ็ค (.dpr) ที่บันทึกไว้ขึ้นมาทำงาน
- (3) คลิกเมนู File > Reopen เพื่อเปิดไฟล์ (.pas) หรือโปรเจ็ค (.dpr) ที่เพิ่งปิดไปขึ้นมาทำงาน

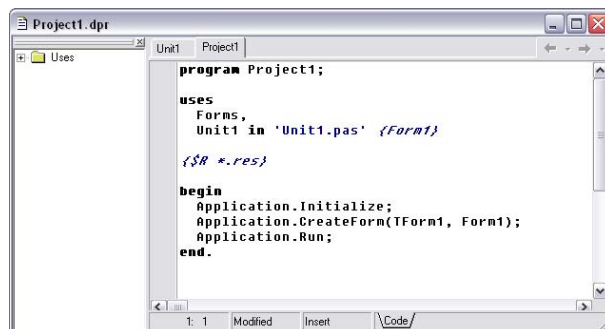
## 2.5 ประเภทไฟล์ของเดลไฟ

ในโปรเจ็คหนึ่ง ๆ ที่เราสร้างขึ้นจะประกอบไปด้วยไฟล์ต่าง ๆ ที่มีหน้าที่และการทำงานที่แตกต่างกัน บางไฟล์ถูกสร้างขึ้นตอนออกแบบ บางไฟล์ถูกสร้างขึ้นหลังจากการบันทึก และบางไฟล์ถูกสร้างขึ้นในขณะที่ทำการคอมไพล์โปรเจ็ค ซึ่งแต่ละไฟล์มีการทำงานหลัก ๆ ดังต่อไปนี้

### 2.5.1 ไฟล์โปรเจ็ค (Project File)

คือไฟล์ที่รวบรวมรายละเอียดของโปรเจ็ค ว่าภายในโปรเจ็คหนึ่งนั้น

ประกอบด้วยฟอรัมหรือยูนิตอะไรบ้าง นอกจากนี้ยังทำหน้าที่สร้างฟอรัมและเรียกฟอรัมหลักของแอปพลิเคชันขึ้นมาทำงานอีกด้วย ในแอปพลิเคชันหนึ่ง ๆ จะมีโปรเจ็คไฟล์เพียงไฟล์เดียว โดยมีนามสกุลเป็น .dpr และจะถูกสร้างขึ้นเมื่อ



```

program Project1;
uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1}
  {$R *.res};
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

```

เราสร้างโปรเจ็คใหม่ ซึ่งถ้าเป็นโปรเจ็คแรก เดลไฟจะกำหนดชื่อมาตรฐานสำหรับโปรเจ็คเป็น Project1.dpr และเมื่อคอมไพล์โปรเจ็คเราจะได้ไฟล์ .exe ที่มีชื่อเดียวกับโปรเจ็คไฟล์

### 2.5.2 ยูนิทไฟล์ (Unit File)

คือไฟล์ที่จัดเก็บมอดูลย่อยของโปรแกรม เรียกว่า ยูนิท (Unit) ซึ่งแต่ละยูนิทจะเก็บรายละเอียดของตัวแปร ค่าคงที่ ชนิดข้อมูล โพรซีเยอร์ และฟังก์ชัน เพื่อใช้ในการทำงานภายในยูนิทนั้น ๆ ปกติเดลไฟ จะกำหนดให้ยูนิทไฟล์มีนามสกุลเป็น .pas ซึ่งความจริงสามารถกำหนดเป็นนามสกุลอื่นก็ได้ เพียงแต่ .pas เป็นนามสกุลมาตรฐานของภาษาปาสกาล หลังจากคอมไพล์หรือรันโปรแกรม เดลไฟจะสร้างไบনারีไฟล์ให้สำหรับแต่ละยูนิท โดยมีชื่อเหมือนกับชื่อยูนิทไฟล์แต่นามสกุลเป็น .dcu

โดยปกติฟอร์มหนึ่ง ๆ จะประกอบด้วยยูนิทไฟล์หนึ่งไฟล์ ซึ่งทำงานสัมพันธ์กับฟอร์มนั้น โดยสิ่งที่อยู่ในยูนิทฟอร์มมีโครงสร้างดังนี้

```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs,  
  StdCtrls;  
type  
  TForm1 = class(TForm)  
    Label1: TLabel;  
    Edit1: TEdit;  
    Button1: TButton;  
    procedure Button1Click(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
var  
  Form1: TForm1;  
implementation  
{$R *.dfm}
```

```

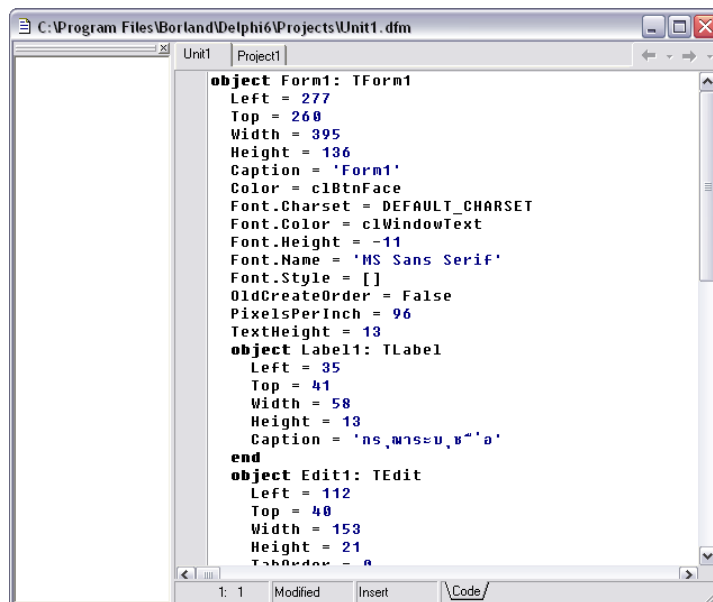
procedure TForm1.Button1Click(Sender: TObject);
begin
    Showmessage('สวัสดี..คุณ'+Edit1.text);
end;
end.

```

### 2.5.3 ฟอর্মไฟล์ (From File)

เดลไฟจะเก็บรายละเอียดของออบเจ็คที่อยู่บนฟอर्मรวมทั้งตัวฟอर्म เช่น

คุณสมบัติของฟอर्म คุณสมบัติของคอมโพเนนท์ที่อยู่บนฟอर्म เป็นต้น โดยแยกเก็บไว้ในอีกไฟล์ต่างหาก เมื่อเราบันทึกโปรเจ็ค เดลไฟจะสร้างโดยให้มีชื่อเหมือนกับยูนิิต แต่มีนามสกุลเป็น .dfm ซึ่งสามารถดูรายละเอียดของไฟล์นี้ได้โดยคลิกเมาส์ขวาบนฟอर्मแล้วเลือก View as Text จะปรากฏหน้าต่างของฟอर्मไว้ดังรูป เมื่อคลิกเมาส์ขวาและเลือก View as Text อีกครั้งก็จะกลับมายังวินโดว์ปกติ



### 2.5.4 ไฟล์ทรัพยากร (Resource File)

เป็นรูปแบบการเก็บข้อมูลที่ใช้สำหรับวินโดว์ ซึ่งปกติจะใช้เก็บส่วนประกอบต่าง ๆ ที่เป็นไบนารี เช่น ไอคอนของแอปพลิเคชันที่เรากำหนดในไดอะล็อกบ็อกซ์ Project Option เป็นต้น โดยปกติเมื่อเราสร้างโปรเจ็ค เดลไฟจะสร้างไฟล์นี้ให้หนึ่งไฟล์โดยใช้ชื่อเดียวกับชื่อโปรเจ็ค



แต่มีนามสกุลเป็น .res เพื่อใช้สำหรับจัดเก็บไอคอนที่กำหนดไว้ใน Project Option และใส่ไว้ในโปรเจ็คต์ด้วยคำสั่ง {\$R\*.RES} เราสามารถสร้างไฟล์ใหม่ขึ้นเองสำหรับเก็บทรัพยากรอื่น เช่น ภาพ บิตแมพ เคอร์เซอร์ ไอคอน หรือข้อความต่าง ๆ แล้วนำมารวมไว้ในแอปพลิเคชันด้วยคำสั่ง {\$R file} นี้ได้

### 2.5.5 ไฟล์โปรเจ็คกรุป (Project Group File)

เป็นไฟล์ที่ใช้เก็บรวบรวมรายการโปรเจ็คต์ที่ได้กำหนดไว้ให้อยู่ในโปรเจ็คกรุปเดียวกัน ดังที่จะแสดงให้เห็นในหัวข้อโปรเจ็คต์เมนเนเจอร์ ไฟล์นี้จะมีนามสกุลเป็น .bpg และเมื่อเราสั่ง Open Project เราก็สามารถเลือกเปิดโปรเจ็คต์เดี่ยว ๆ (ไฟล์นามสกุล .dpr) หรือเปิดโปรเจ็คกรุป (ไฟล์นามสกุล .bpg) ได้