บทที่ 4
การพัฒนาโปรแกรม

การพัฒนาโปรแกรมจะต้องติดตั้งโปรแกรมไมโครซอฟท์ สปีช เอสดีเค 5.1 จะมีเครื่องมือต่างๆ สำหรับพัฒนาโปรแกรมที่เกี่ยวกับการรู้จำเสียง และ มีตัวอย่างการใช้งาน มีเอนจิ้นที่ใช้แปลงข้อมูลที่อยู่ในรูปแบบตัวอักษร ให้ออกมาเป็นเสียงพูด (TTS: Text To Speech Engines) และเอนจิ้นที่ใช้สำหรับจดจำเสียง (SR: Speech Recognition Engines)

สำหรับเครื่องมือในการพัฒนา ผู้ศึกษาเลือกพัฒนาโดยใช้โปรแกรมไมโครซอฟท์ วิชวล สตูดิโอ ดอทเน็ต โดยเขียนด้วยภาษาซีชาร์ป

ในหัวข้อ 4.1 จะแสดงให้เห็นถึงการทำงานของโปรแกรม และการเตรียมพร้อมเพื่อใช้โปรแกรมการใช้คำสั่งเสียงเพื่อสร้างรายงาน และในหัวข้อ 4.2 ถึง 4.5 แสดงรายละเอียดการพัฒนาโปรแกรม

4.1 ขั้นตอนการใช้งานโปรแกรม

ขั้นตอนการใช้งานโปรแกรม ประกอบไปด้วย ขั้นตอนต่างๆดังนี้

ขั้นตอนที่ 1 ก่อนการใช้งานโปรแกรมจะต้องทำการติดตั้ง Speech Recognition ซึ่งจะแสดงวิธีการติดตั้งไว้ในตอนท้ายของหัวข้อ 4.5

ขั้นตอนที่ 2 ทำการฝึกคอมพิวเตอร์ให้จดจำวิธีการพูดของผู้ใช้ เก็บเป็นประวัติผู้ใช้ โดยการอ่านออกเสียงข้อความฝึกอบรมที่เตรียมไว้ให้ ขั้นตอนการฝึกแสดงในภาคผนวก ข

ขั้นตอนที่ 3 ติดตั้งโปรแกรม VoiceCommandDemo.exe ขั้นตอนแสดงในภาคผนวก ก

ขั้นตอนที่ 4 เปิดโปรแกรม VoiceCommandDemo.exe ทำการกรอกข้อมูลผู้ใช้ รหัสผู้ใช้ และติดต่อฐานข้อมูล จากนั้นเริ่มใช้งานได้ ขั้นตอนการใช้งานแสดงในภาคผนวก ค

ขั้นตอนที่ 5 ถ้าต้องการเพิ่มคำศัพท์จากฐานข้อมูล เพื่อใช้ฝึกออกเสียงเพิ่มเติม สามารถทำได้ตามขั้นตอนในภาคผนวก ง

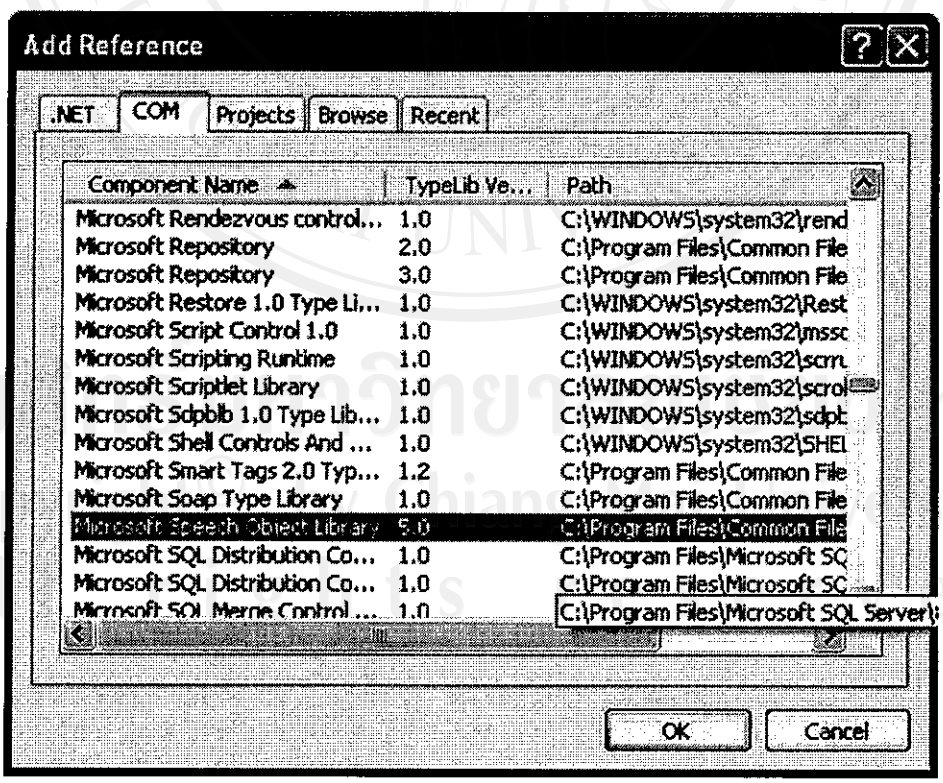
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

4.2 การกำหนดคุณสมบัติ ไวยากรณ์ และคำสำคัญในการจดจำเสียง

ในการพัฒนาโปรแกรมเพื่อให้โปรแกรมสามารถจดจำเสียงได้นั้น จะต้องมีการติดตั้งไมโครซอฟท์สปีชไลบารี่เพื่อให้โปรแกรมสามารถจดจำเสียงได้ ดังนั้นเมื่อเปิดโปรแกรมไมโครซอฟท์ วิชวล สตูดิโอ ขึ้นมา จะต้องทำการติดตั้งไมโครซอฟท์สปีชไลบารี่เพื่อเตรียมพร้อมสปีชเอนจิ้นไว้สำหรับจดจำเสียง นอกจากนั้นจะต้องสร้างไวยากรณ์ สร้าง EventHandler และ menuRule เพื่อใช้สำหรับการกำหนดคำที่จะให้ระบบสามารถจดจำได้

4.2.1 การติดตั้งค่าเริ่มต้นสำหรับไมโครซอฟท์ สปีช ไลบารี่

โปรแกรมที่ออกแบบจะสามารถใช้งานไมโครซอฟท์สปีชไลบารี่ได้ต้องทำการเพิ่ม Reference: Microsoft Speech Object Library เพื่อให้เราสามารถเรียกใช้ SAPI ในโปรแกรมได้ผ่านทาง SpeechLib ซึ่งจะทำให้เราใช้ method และ property ต่างๆ กว่า 400 ชนิด ซึ่งถูกจัดอยู่ในคลาสย่อยๆ เช่น คลาส ISpeechRecoResult ทำหน้าที่เก็บผลลัพธ์ จากการจดจำเสียง เมื่อต้องการแปลงคำพูดเป็นข้อความ ก็ทำได้โดยเรียกใช้เมธอด GetText จากคลาส ISpeechRecoResult ได้โดยไม่ต้องเรียกในลักษณะชื่อเต็ม การเพิ่ม Reference ให้อ้างอิงถึงไลบารี่ SAPI จาก C:\Program Files\Common Files\Microsoft Shared\Speech\sapi.dll ดังรูป 4.1



รูป 4.1 แสดงการเพิ่ม Microsoft Speech Object Library เพื่อใช้ในโปรแกรม

จากรูป 4.1 ให้เปิดหน้าต่าง Add Reference จากโปรแกรมที่ทำพัฒนา (Microsoft Visual Studio) โดยเมื่อเปิด Project ขึ้นมาแล้ว ให้ไปที่ Menu bar >> Project >> คลิกที่ Add Reference เมื่อเปิดหน้าต่างขึ้นมาให้เลือกที่แทป COM เลือกรายการ Microsoft Speech Object Library คลิกตกลง จะปรากฏรายการ SpeechLib ขึ้นในหน้าต่าง Reference

4.2.2 การเรียกใช้ออปเจ็กต์ SpeechLib

การเรียกใช้ออปเจ็กต์ SpeechLib ทำได้โดยเพิ่มเนมสเปซ SpeechLib เพื่อที่จะสามารถเรียกใช้ SubClass ในโปรแกรมได้สะดวกยิ่งขึ้น ตัวอย่างเช่น เมื่อต้องการใช้คลาส ISpeechRecoResult ตามปกติถ้าต้องการเรียกใช้ SubClass จะต้องเรียกใช้โดยอ้างคลาสให้ครบ (SpeechLib. ISpeechRecoResult) แต่เมื่อทำการประกาศเนมสเปซ จะทำให้สามารถเรียกใช้ SubClass หรือ method เหล่านั้นโดยไม่ต้องอ้างคลาส SpeechLib ซ้ำทุกครั้งที่ใช้งาน ทำให้เกิดความสะดวกในการเขียนโปรแกรม เช่น ISpeechRecoResult.GetText(); โดยการประกาศเนมสเปซให้ใช้คำสั่งดังนี้

```
using SpeechLib;
```

4.2.3 การประกาศตัวแปรเพื่อใช้งาน SubClass

การประกาศตัวแปรเพื่อใช้งาน SubClass ของ SpeechLib ซึ่งประกอบไปด้วย 3 Subclass คือ SpSharedRecoContext, ISpeechRecoGrammar และ ISpeechGrammarRule โดยประกาศดังนี้

- ประกาศตัวแปร objRecoContext เพื่อเรียกใช้งานคลาส SpSharedRecoContext ใช้สำหรับสร้าง EventHandler ต่างๆ ในการจดจำคำ

```
private SpeechLib.SpSharedRecoContext objRecoContext = null;
```

- ประกาศตัวแปร grammar เพื่อเรียกใช้งานคลาส ISpeechRecoGrammar ใช้สำหรับสร้างไวยากรณ์ ซึ่งในโปรแกรมสามารถมี grammar ได้มากกว่าหนึ่งชุด

```
private SpeechLib.ISpeechRecoGrammar grammar = null;
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- ประกาศตัวแปร menuRule เพื่อเรียกใช้งานคลาส ISpeechGrammarRule ใช้เพื่อกำหนดกฎ ให้ SR(Speech Recognition) สามารถตรวจสอบและจดจำได้

```
private SpeechLib.ISpeechGrammarRule menuRule = null;
```

4.2.4 ขั้นตอนการกำหนด อีเวนต์แฮนเดิล ขั้นตอนการกำหนด อีเวนต์แฮนเดิล

เพื่อให้โปรแกรมสามารถรองรับคำสั่งเสียงต่างๆได้ เมื่อมีการอินพุตเสียงเข้ามา โดยกำหนดตัวแปร Hypo_Event เพื่อรองรับเหตุการณ์เมื่อ SR (Speech Recognition) ทำการตรวจสอบอินพุตเสียง และ ตัวแปร Reco_Event เพื่อรองรับเหตุการณ์เมื่อ SR สามารถจำหรือรู้จักเสียงนั้นๆ ได้โดยกำหนดดังนี้

```
objRecoContext.Hypothesis+=new
_ISpeechRecoContextEvents_HypothesisEventHandler(Hypo_Event);
objRecoContext.Recognition += new
_ISpeechRecoContextEvents_RecognitionEventHandler(Reco_Event);
```

Hypothesis Event จะพยายามคาดเดาเสียงที่เข้ามา โดยสร้างคำขึ้นมาชั่วคราว เพื่อนำไปตรวจสอบกับคำที่ถูกกำหนดไว้ใน menuRule และ Hypo_Event ถูกกำหนดขึ้นมาเพื่อทำหน้าที่ต่อจากนั้น ซึ่งจะนำไปอธิบายในหัวข้อที่ 4.2.1 และ Recognition Event จะเกิดขึ้นเมื่อคำนั้นๆ ถูก SR จดจำคำนั้นได้ เหตุการณ์นั้นจะถูกนำไปใช้ต่อที่ฟังก์ชั่น Reco_Event ซึ่งจะนำไปอธิบายในหัวข้อที่ 4.2.2

4.2.5 กำหนดออปเจกต์ grammar

โดยสามารถกำหนดผ่านเมธอด CreateGrammar() ซึ่งการใช้งานออปเจกต์ grammar สามารถกำหนดได้หลายตัว แต่ในที่นี้จะใช้การกำหนด Grammar ชุดเดียว และเมื่อต้องการเปลี่ยนคำสำคัญในแต่ละขั้นตอน จะใช้วิธีการ reset grammar การประกาศส่วนนี้เพื่อสร้างไวยากรณ์ เมื่อกำหนดขึ้นแล้วต้องนำไปสร้างชุดคำสั่งจะประกาศในหัวข้อ 4.1.6 โดยการกำหนด grammar ทำได้ดังนี้

```
grammar = objRecoContext.CreateGrammar(0);
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

4.2.6 การสร้างไวยากรณ์สำหรับการจดจำเสียง

ใช้วิธีการกำหนดคุณลักษณะ(Attributes) ผ่านคลาสของ SpeechRuleAttributes ซึ่งในโปรแกรมนี้กำหนด Attributes ขึ้นมา 2 ชนิดคือ SRATopLevel ซึ่งหมายถึง ไวยากรณ์ที่กำหนดขึ้นมาจะสามารถกำหนดให้ activated or deactivated ได้ และ Attribute SRADynamic หมายถึง Grammar จะสามารถถูกเปลี่ยนแปลงได้ตลอดในขณะใช้งาน การสร้างเมนูขึ้นมาเพื่อรองรับคำสำคัญ และเพิ่มคำสำคัญเข้าไป และหลังจากเพิ่มคำสำคัญครบ ให้สั่ง Commit Rule เพื่อสร้างกฎ ตัวอย่างการสร้าง menuRule และ กำหนดคำสั่ง "Log in" ทำได้ดังนี้

```
menuRule = grammar.Rules.Add("MenuCommands",
SpeechRuleAttributes.SRATopLevel | SpeechRuleAttributes.SRADynamic, 1);
object PropValue = "";
menuRule.InitialState.AddWordTransition(null, "Log in", " ",
SpeechGrammarWordType.SGLexical, "log in", 1, ref PropValue, 1.0F);
grammar.Rules.Commit();
grammar.CmdSetRuleState("MenuCommands",
SpeechRuleState.SGDSActive);
```

4.3 การสร้างฟังก์ชั่น เพื่อรองรับเหตุการณ์เมื่อมีการนำข้อมูลสียงเข้า

การนำข้อมูลเสียงมาใช้เพื่อออกคำสั่งต่างๆ จะต้องกำหนดฟังก์ชั่นเพื่อมารองรับเหตุการณ์เมื่อมีการนำเข้าข้อมูลเสียง ในที่นี้จะกำหนดสองฟังก์ชั่นขึ้นมา

4.3.1 ฟังก์ชั่น Hypo_Event

ฟังก์ชั่น Hypo_Event รองรับเหตุการณ์ ขณะคาดเดาอินพุตเสียงเทียบกับ คำสำคัญที่กำหนด โดยจะแสดงการคาดเดาผ่าน txtHyp.text

```
private void Hypo_Event(int StreamNumber, object StreamPosition,
ISpeechRecoResult Result)
        {
      txtHyp.Text = Result.PhraseInfo.GetText(0, -1, true);
        }
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

4.3.2 ฟังก์ชั่น Reco_Event

ฟังก์ชั่น Reco_Event รองรับเหตุการณ์ เมื่ออินพุตเสียงที่ได้ ตรงกับคำสำคัญที่กำหนด ซึ่งเราสามารถนำอีเวนต์ที่เกิดขึ้นเพื่อไปใช้ในการสร้างประโยคสำหรับติดต่อฐานข้อมูลได้ โดยคำสำคัญที่ถูกจดจำได้ จะนำมาแสดงผลผ่าน txtReco.Text

```
private void Reco_Event(int StreamNumber, object StreamPosition,
SpeechRecognitionType RecognitionType, ISpeechRecoResult Result)
{
 txtReco.Text = Result.PhraseInfo.GetText(0, -1, true);
 //โดยที่เราสามารถนำศัพท์เหล่านี้ ไปรวมกันเพื่อสร้างเป็นประโยคต่างๆ ได้
  โดยผ่านเงื่อนไขต่างๆ ดังข้อ 4.3
}
```

4.4 การสร้างเงื่อนไขเพื่อรองรับเหตุการณ์ต่างๆ

เมื่ออีเวนต์ Reco_Event เกิดขึ้นให้นำผลที่เกิดขึ้นมาใช้ในการสร้างเหตุการณ์ต่างๆ ดังนี้

- เมื่อคำสำคัญตรงกับคำว่า “Log in” ในฟอร์มล็อกอิน ให้ทำการติดต่อฐานข้อมูล จัดเก็บชื่อตารางจากฐานข้อมูล ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ สำหรับเรียกใช้ชื่อตาราง “From Table” เปิดฟอร์มสำหรับคิวรี่ติดต่อฐานข้อมูลโดยใช้ข้อมูลที่ user ป้อนค่าจาก TextBox เช่น

```
        strConnection = "Data Source = ชื่อเครื่องที่ติดต่อ; Initial Catalog = Northwind ; User
        Id=sa ; Password =Pwd"
จัดเก็บชื่อตารางโดยใช้คำสั่ง
        SELECT id,name FROM syscolumns WHERE id IN (SELECT id FROM sysobjects
        WHERE xtype = 'U'
ทำการ reset grammar และเพิ่มคำสำคัญโดยใช้คำสั่ง
        grammar.Reset(0);
        grammar = objRecoContext.CreateGrammar(0);
เพิ่มคำสำคัญ “From Table” โดยใช้คำสั่ง
        menuRule.InitialState.AddWordTransition(null, "From Table", " ",
        SpeechGrammarWordType.SGLexical, "From Table", 1, ref PropValue, 1.0F);
        grammar.Rules.Commit();
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- เมื่อคำสำคัญตรงกับคำว่า “From Table” ในฟอร์มคิวรี่ ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ “ชื่อตารางทั้งหมด” เก็บค่า “From” ไว้ในตัวแปร QueryStatement

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);
menuRule.InitialState.AddWordTransition(null, "Table Name_1", " ",
SpeechGrammarWordType.SGLexical, "Table Name_1", 1, ref PropValue, 1.0F);
menuRule.InitialState.AddWordTransition(null, "Table Name_2", " ",
SpeechGrammarWordType.SGLexical, "Table Name_2", x, ref PropValue, 1.0F);
...
menuRule.InitialState.AddWordTransition(null, "Table Name_x", " ",
SpeechGrammarWordType.SGLexical, "Table Name_x", x, ref PropValue, 1.0F);
grammar.Rules.Commit();
string QueryStatement = null;
QueryStatement += "From"
```

- เมื่อคำสำคัญตรงกับคำว่า “ชื่อ Table” ในฟอร์มคิวรี่ ทำการติดต่อฐานข้อมูล เพื่อจัดเก็บชื่อคอลัมน์ที่อยู่ในตารางที่เลือกจากฐานข้อมูล ทำการรีเซ็ต grammar และเพิ่มคำสำคัญสำหรับเรียกใช้ชื่อคอลัมน์ “Select Star” และ “Select Column” เก็บค่า “ชื่อ Table” ไว้ในตัวแปร QueryStatement

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);
menuRule.InitialState.AddWordTransition(null, "Select Star", " ",
SpeechGrammarWordType.SGLexical, " Select Star ", 1, ref PropValue, 1.0F);
menuRule.InitialState.AddWordTransition(null, "Select Column", " ",
SpeechGrammarWordType.SGLexical, " Select Column", 2, ref PropValue, 1.0F);
grammar.Rules.Commit();
QueryStatement += " " + " Table Name_x"
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- เมื่อคำสำคัญตรงกับคำว่า “Select Column” ในฟอร์มคิวรี่ ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ “ชื่อคอลัมน์ทั้งหมด” เก็บค่า “Select” ไว้ในตัวแปร QueryStatement1

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);
menuRule.InitialState.AddWordTransition(null, "Column Name_1", " ",
SpeechGrammarWordType.SGLexical, "Column Name_1", 1, ref PropValue, 1.0F);
menuRule.InitialState.AddWordTransition(null, "Column Name_2", " ",
SpeechGrammarWordType.SGLexical, "Column Name_2", x, ref PropValue, 1.0F);
...
...
menuRule.InitialState.AddWordTransition(null, "Column Name_x", " ",
SpeechGrammarWordType.SGLexical, "Column Name_x", x, ref PropValue, 1.0F);
grammar.Rules.Commit();
string QueryStatement1 = null;
string QueryStatement2 = null;
QueryStatement1 = "Select"
QueryStatement2 = QueryStatement1 + QueryStatement ;
```

- เมื่อคำสำคัญตรงกับคำว่า “Select Star” ในฟอร์มคิวรี่ ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ “Show Result” และเก็บค่า “Select *” ไว้ในตัวแปร QueryStatement

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);
menuRule.InitialState.AddWordTransition(null, "Show Result", " ",
SpeechGrammarWordType.SGLexical, "Show Result", 1, ref PropValue, 1.0F);
grammar.Rules.Commit();
string QueryStatement1 = null;
string QueryStatement2 = null;
QueryStatement1 = "Select *"
QueryStatement2 = QueryStatement1 + QueryStatement ;
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- เมื่อคำสำคัญตรงกับคำว่า “ชื่อ Column” ในฟอร์มคิวรี่ ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ “Show Result” เก็บค่า Column ไว้ในตัวแปร QueryStatement

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);

menuRule.InitialState.AddWordTransition(null, "Show Result", " ",
SpeechGrammarWordType.SGLexical, "Show Result", 1, ref PropValue, 1.0F);

grammar.Rules.Commit();

QueryStatement1 += " Column Name_x "
QueryStatement2 = QueryStatement1 + QueryStatement ;
```

- เมื่อคำสำคัญตรงกับคำว่า “Show Result” ในฟอร์มคิวรี่ ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ “order by” ไว้ใช้ในฟอร์มรายงาน เปิดฟอร์มรายงาน นำ QueryStatement มาใช้ในการเรียกข้อมูลจากฐานข้อมูล นำผลมาแสดงใน datagrid ในฟอร์มรายงาน

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);

menuRule.InitialState.AddWordTransition(null, "order by", " ",
SpeechGrammarWordType.SGLexical, "order by", 1, ref PropValue, 1.0F);

grammar.Rules.Commit();
OleDbDataAdapter da = new OleDbDataAdapter(QueryStatement2,Conn);
DataSet ds =  new DataSet();
da.Fill(ds, "Report");
dataGridView1.DataSource = ds;
string Query_report = "";
Query_report = QueryStatement2 + " order by ";

Form frmReport = new ReportPage();
frmReport.ShowDialog();
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- เมื่อคำสำคัญตรงกับคำว่า “order by” ในฟอร์มรายงาน ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ “ชื่อคอลัมน์”

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);

menuRule.InitialState.AddWordTransition(null, "Column Name_x", " ",
SpeechGrammarWordType.SGLexical, "Column name_x", 1, ref PropValue, 1.0F);
grammar.Rules.Commit();
Query_report += " Column name_x"
```

- เมื่อคำสำคัญตรงกับคำว่า “ชื่อ column” ในฟอร์มรายงาน ทำการรีเซ็ต grammar และเพิ่มคำสำคัญ “order by” ทำการเรียงลำดับข้อมูลตามคอลัมน์ที่เลือก

```
grammar.Reset(0);
grammar = objRecoContext.CreateGrammar(0);

menuRule.InitialState.AddWordTransition(null, "order by", " ",
SpeechGrammarWordType.SGLexical, "order by", 1, ref PropValue, 1.0F);

grammar.Rules.Commit();
OleDbDataAdapter da = new OleDbDataAdapter(QueryStatement2,Conn);
DataSet ds =  new DataSet();
da.Fill(ds, "Report");
dataGridView1.DataSource = ds;
Query_Report  = QueryStatement2 + " order by ";
```

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

4.5 สร้างคำสำคัญเพื่อการฝึกหัด

ในการใช้โปรแกรมคำสำคัญบางคำที่ผู้ใช้ออกเสียงไม่ถูกต้อง เราสามารถทำการฝึกหัดให้โปรแกรมเรียนรู้เพิ่มเติมได้โดยเพิ่มคำศัพท์สำหรับการฝึกหัดใส่ไว้ใน textfile สำหรับใช้ในการฝึกหัด

- เพิ่มเนมสเปซ System.IO เพื่อใช้ในการเปิดอ่าน และเขียนไฟล์ลงใน textfile

```
using System.IO;
```

- เมื่อคลิกที่คำศัพท์ให้เก็บคำศัพท์นั้นเพิ่มในไฟล์ Voice-Training.txt ที่สร้างออปเจกต์ sr มาจากออปเจกต์ StreamReader เพื่ออ่านข้อมูลเดิมจากไฟล์ Voice-Training.txt และเขียนคำศัพท์ที่เลือก(ListToSelect) เพิ่มเข้าไปเก็บในไฟล์

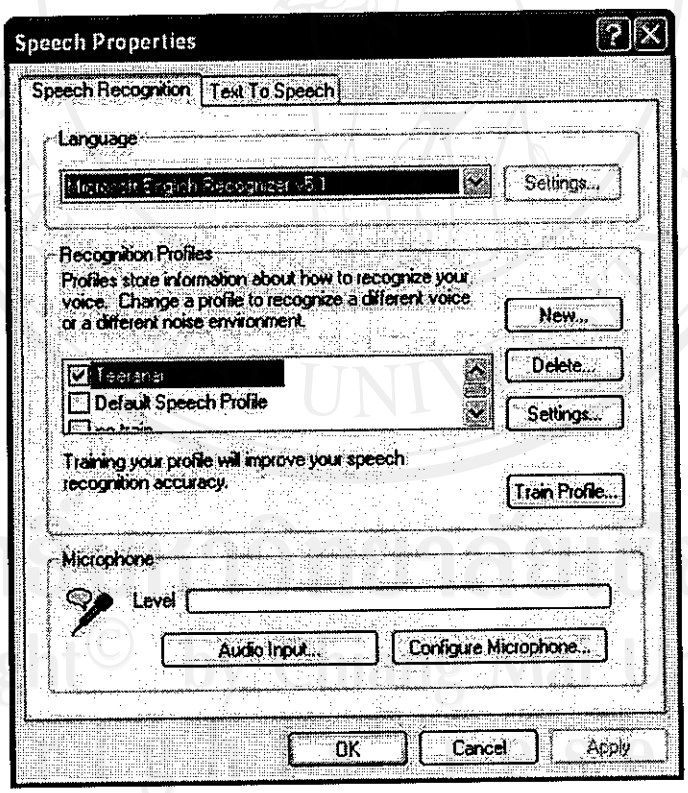
```
strPath = "./Voice-Training.txt";

StreamReader sr = new StreamReader(strPath);
string tempReader = sr.ReadToEnd();
sr.Close();
StreamWriter sw = new StreamWriter(strPath, false);
sw.WriteLine(tempReader.ToString() + " " +ListToSelect.ToString());
sw.Flush();
sw.Close();
- เรียกใช้โปรแกรมฝึกหัด
string RecoPath = "./Reco.exe";
System.Diagnostics.Process.Start(RecoPath);
```

ภายหลังการพัฒนาโปรแกรมสำเร็จ ผู้ใช้สามารถนำตัวติดตั้งโปรแกรมไปทำการติดตั้งยังเครื่องคอมพิวเตอร์ได้ เมื่อทำการติดตั้งที่เครื่องคอมพิวเตอร์ใดๆแล้ว ผู้ใช้จะต้องตรวจสอบการเรียกใช้ Speech Recognition มีการเรียกใช้อยู่แล้วหรือไม่ โดยตรวจสอบได้จาก Control Panel จากนั้นเปิดโปรแกรม Speech ขึ้นมา ให้สังเกตที่แทป Speech Recognition จะเป็นดังรูป 4.2 หากไม่พบแท็บดังกล่าว ผู้ใช้จะไม่สามารถเรียกใช้คุณสมบัติ Speech Recognition ได้ ดังนั้นผู้ใช้จะต้องทำการเรียกใช้ Speech Recognition ก่อนการใช้งาน โดยที่ Speech Recognition มีให้ใช้เฉพาะใน Microsoft Office รุ่นภาษาจีนแบบประยุกต์ จีนแบบดั้งเดิม ภาษาอังกฤษ (สหรัฐ) และภาษาญี่ปุ่น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การติดตั้ง Speech Recognition ด้วยตนเองสามารถทำได้ดังนี้

1. ที่ Control Panel ของ Microsoft Windows คลิกสองครั้งที่ Add/Remove Programs
2. คลิก Change or Remove Programs เลือก Microsoft Office XP(แล้วแต่รุ่นที่เลือกใช้) จากนั้นคลิก Change
3. คลิก เพิ่มหรือเอาคุณลักษณะออก จากนั้นคลิก ถัดไป
4. เลือกกล่องกาเครื่องหมาย เลือกการกำหนดรายละเอียดโปรแกรมด้วยตนเองขั้นสูง แล้ว คลิก ถัดไป
5. ใต้หัวข้อ เลือกตัวเลือกการปรับปรุงสำหรับโปรแกรมประยุกต์และเครื่องมือ ดูที่ คุณลักษณะการใช้ข้อมูลร่วมกันใน Office แล้วคลิก
6. หน้า Alternative User Input ให้คลิก คำพูด จากนั้นเลือกชนิดของการติดตั้งที่ต้องการ
7. คลิก ปรับปรุง



รูป 4.2 แสดงการตรวจสอบการเรียกใช้งาน Speech Recognition

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved