บทที่7

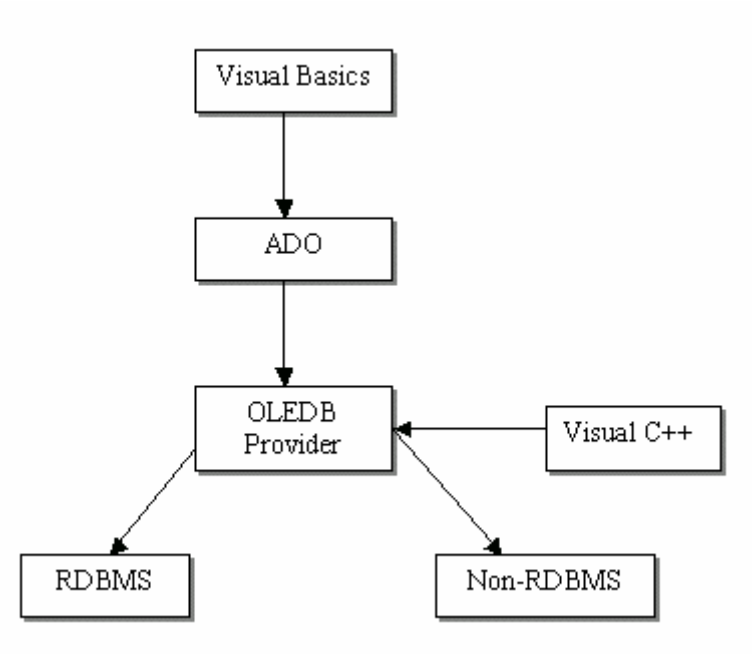
ADO.net

7.1 รู้จัก ADO.net

ตัว ADO.net เป็นเทคโนโลยีอันหนึ่งที่ช่วยในการติดต่อกับ ฐานข้อมูลต่างๆ และ ADO.net ยังมีโครงสร้างเหมือน ADO เพียงแต่การออกแบบมีความแตกต่างกันบ้าง

การเก็บข้อมูลที่เก็บใน ADO เมื่อมีการ select ข้อมูลนั้น ผลลัพธ์ที่ออกมาจะเก็บข้อมูลในโครงสร้างของ COM (Common Object Model) แต่ตัว ADO.net เมื่อเรา select ข้อมูลออกมาแถวต่างๆที่ select ออกมาจะเก็บใว้ในรูปของ XML เพราะฉะนั้น ADO.net จึงเหมาะกับการทำงานที่มีเป็นแอพพลิเคชั่นบนอินเทอร์เน็ต

ถ้าเป็น ADO ปกติ จะเหมาะกับการทำงานที่เป็นไคลเอนต์ / เซิร์ฟเวอร์ ที่อยู่ในเครือข่าย แลน หรือเครือข่ายอินทราเน็ต(Intranet) ในองค์การของเรามากกว่า



รูปที่ 7-1 โครงสร้างของ ADO

การเชื่อมต่อฐานข้อมูลบนเครือข่ายภายในองค์กรนั้น เราสามารถกำหนดการเชื่อมต่อได้ คือ เมื่อปิดแอพพลิเคชั่น เราค่อยมาปิดการเชื่อมต่อฐานข้อมูล ซึ่งเรียกว่าเป็น Connection Oriented แต่เมื่อเอาไปใช้กับการทำงานที่เป็นแอพพลิเคชั่นบนอินเทอร์เน็ต ซึ่งเป็นระบบที่จำนวนผู้ใช้ไม่แน่นอน และสูงกว่าระบบเครือข่ายในบริษัทแน่นอน เพราะฉะนั้นการที่เรามาเชื่อมต่อกับฐานข้อมูลทั้งไว้จึงดูไม่เหมาะกับการทำงานในเครือค่ายอินเทอร์เน็ตนัก ADO.net จึงมีคุณสมบัติที่สนับสนุน Disconnected Connection ด้วย

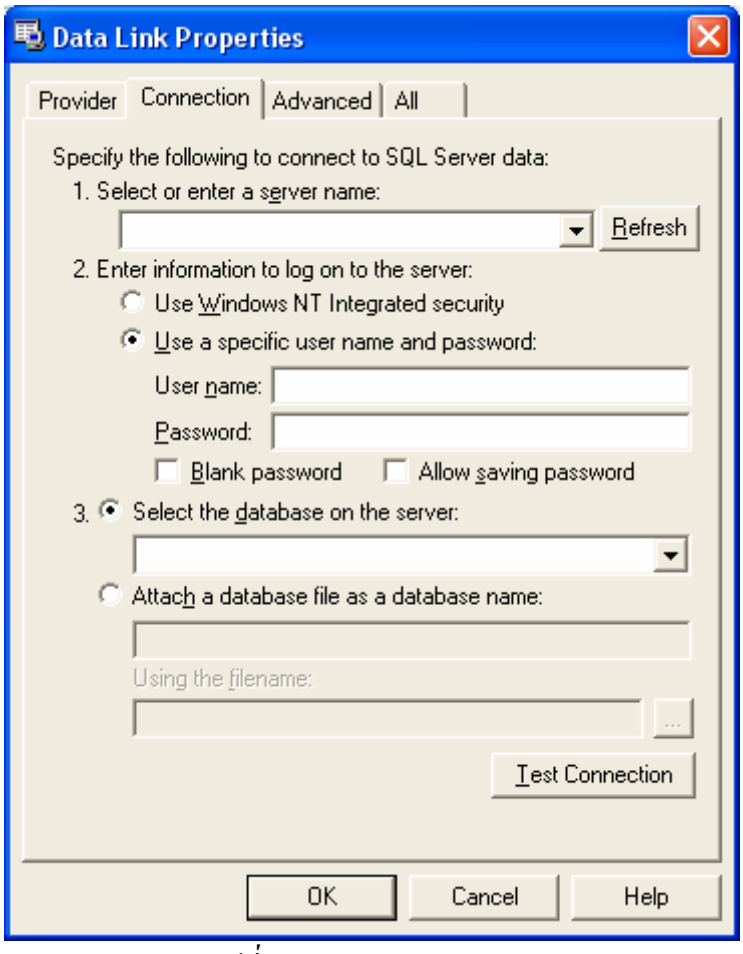
นอกจากนั้นยังสามารถใช้กับ Data Source หลายๆ แบบเหมือนเดิม เช่น Access กับ OLE DB provider ของหลายๆ แบบ หรือกระทั่งข้อมูลใน Outlook Exchange เราก็สามารถใช้ ADO.net ไปselest ออกมาได้เช่นเดียวกัน

7.2 ออบเจ็กต์ที่สำคัญใน ADO.net

ออบเจ็กต์ที่ใช้บ่อยๆ ใน ADO.net ซึ่งเราจะกล่าวถึงในส่วนนี้ ก็คือ Connection ,Command ,dataset และ DataAdater

7.2.1 ออบเจ็กต์ Connection

ออบเจ็กต์ตัวแรกคือ Cinnection เป็น ออบเจ็กต์ที่อยู่ใน ADO.net หน้าที่คือ การเปิดการเชื่อมต่อ (Connectuion) ระหว่างตัวไคลเอ็นต์ของเรากับ DataSource เราสามารถกำหนดรูปแบบการเชื่อมต่อไปยังฐานข้อมูลที่ต้องการ พร้อมค่ารายละเอียดต่างๆ เช่น รหัสผ่าน เมื่อจะเชื่อมต่อกับฐานข้อมูล ซึ่งยังคงคล้ายกับ ADO แบบเดิมนั่นเอง



รูปที่7- 2 Data Link Properties

ตัวอย่างโค้ดการใช้ออบเจ็กต์ Connection เชื่อมต่อฐานข้อมูล

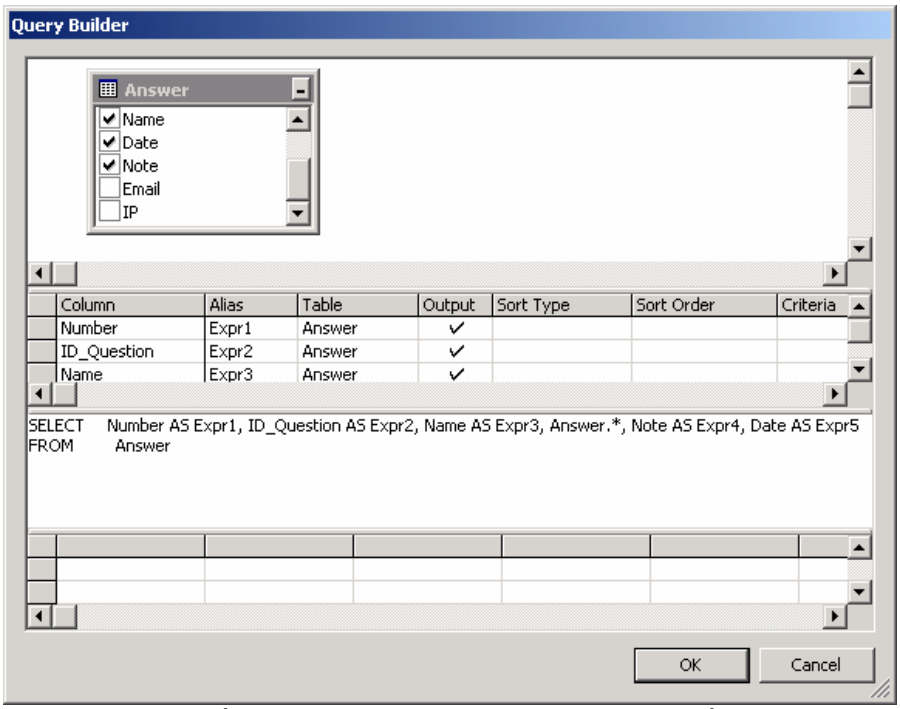
```
Using System.Data.sqlClient
SQLConnection Cnn = new SQLconnection();
Cnn.ConnectionString = " server= localhost ; uid= sa; database =pubs";
Cnn.Open();
```

ตัวอย่างเป็นการเขียนโค้ดด้วย Visual C# บรรทัดแรก คือการเรียกใช้ Name Space ของ System.data.SQL.Client จากนั้นก็สร้าง Instant ของมันขึ้น แล้วกำหนด ConnectionString คือระบุชื่อเซิร์ฟเวอร์ ชื่อของฐานข้อมูล ชื่อของฐานข้อมูล ชื่อของ User พร้อมกำหนดชนิด ของ OLE DB คือชื่อของ Data Source ซึ่งต้องกำหนดในนี้เหมือนกัน จากนั้น ก็สั่ง open เพื่อเปิดการเชื่อมต่อ

7.2.2 ออบเจ็กต์ Command

ออบเจ็กต์ ตัวนี้มีหน้าที่เอาไว้เขียนคำสั่ง SQL จากไคลเอ็นต์ไปทำงานบน Data Source เช่นเดียวกับออบเจ็กต์ Connection คือใน ADO ปกติ ยังมีออบเจ็กต์นี้อยู่

การใช้งานออบเจ็กต์ Command ใน ADO.net นี้ก็เหมือนกันกับการทำงานใน ADO เดิม คือเวลาที่เราจะส่งคำสั่ง SQL ที่ไม่ใช่ Select เช่น Insert ,Update หรือ Delect ที่ไม่มีการ Return ค่ากลับมานั้น ส่วนใหญ่ในทุกฐานข้อมูลจะมีโครงสร้างภาษา SQL ให้เราสามารถสร้างฟังก์ชั่นฝังในตัวฐานข้อมูลเลย ซึ่งส่วนใหญ่ในทุกฐานข้อมูลจะมีโครงสร้างภาษา SQL ให้เราสามารถสร้างฟังก์ชั่นฝังไว้ในตัวฐานข้อมูลได้

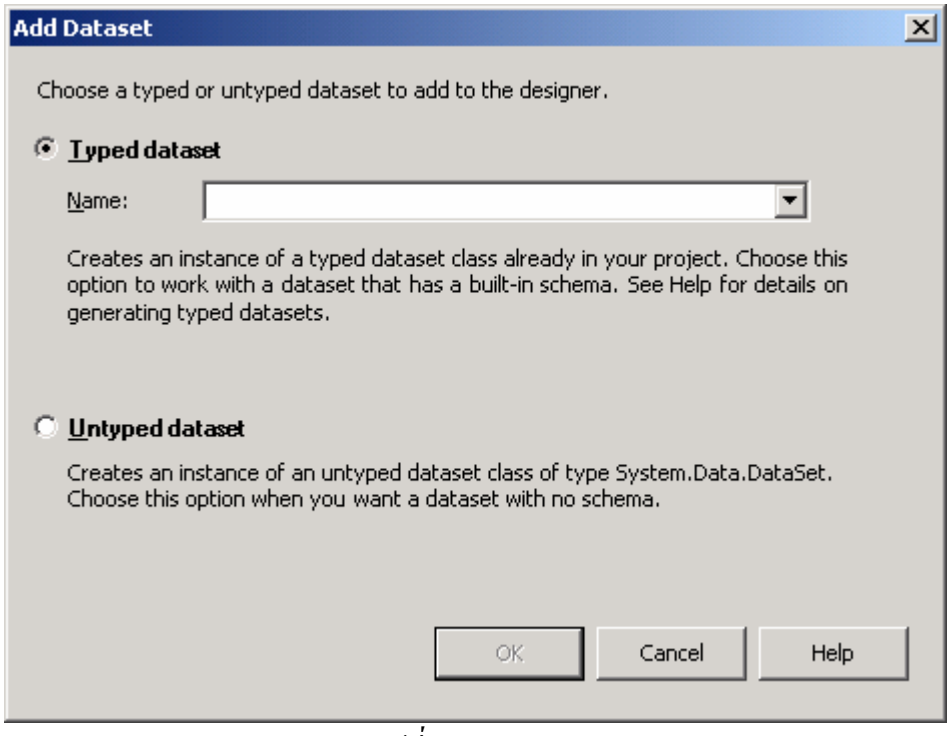


รูปที่7-3 Query Builder แสดงการใช้รูปแบบละคำสั่ง

7.2.3 ออบเจ็กต์ Dataset

ออบเจ็กต์ Dataset นี้มีเฉพาะใน ADO .net เท่านั้น Dataset เปรียบเสมือนฐานข้อมูลทั้งก้อนเลย คือเมื่อเรา select ออกมา เราไม่ได้เฉพาะแถวของข้อมูลออกมา แต่ได้ข้อมูลอันใหม่อันหนึ่งโดยเป็นฐานข้อมูลที่อยู่ในหน่วยความจำของเรา

ภายใน Dataset นั้นจะมองว่าเป็นเหมือนฐานข้อมูลหนึ่งก้อนในนั้นมีทั้งตาราง (table) คอลัมน์ (columns) และแถว (rows) ต่างๆ อยู่ภายใน มีความสัมพันธ์ระหว่างตารางได้ มีการกำหนด constrain ,index key ,public key ได้ โดยการเก็บข้อมูลของ Dataset จะเก็บทั้ง Schema และ ข้อมูลของตัวมันด้วย



รูปที่ 7-4 Dataset

เราสามารถ select ข้อมูลออกมาเก็บไว้ใน Dataset จากนั้นก็สามารถออกจากการเชื่อมต่อได้โดยเราก็จะมี Dataset เอาไว้ใช้งาน จากภายในโปรแกรมเราก็สามารถใช้คำสั่ง insert,update,delete ลงไปใน dataset ได้ จนเสร็จรีบร้อยแล้วเราสามารถกลับมาเชื่อมต่อกับฐานข้อมูลอีกที่หนึ่ง แล้วทำการซิงค์ (Synch) ข้อมูลกลับขึ้นไปได้ โดยตัว ADO.net จะคอยจัดการความแตกต่างระหว่างข้อมูลด้วยกันเอง

การซิงค์ข้อมูลกลับขึ้นไปในลักษณะนี้ คือการเขียนโปรแกรมแบบ Disconnection Programming โดยไม่จำเป็นต้องเชื่อมต่อฐานข้อมูลตลอดเวลา เพราะแอพพลิเคชั่นบนอินเทอร์เน็ตนั้น ถ้าเราเชื่อมต่อฐาข้อมูลไว้ตลอดเวลา จะเป็นการสิ้นเปลืองทรัพยากรของเซิร์ฟเวอร์มาก

นอกจากนี้ เนื่องจากใน Dataset มีหลายตาราง และแต่ละตารางอาจจะมาจากคนละ Data Source เช่น

1. ตารางแรกมาจากฐานข้อมูล SQL
2. ตารางที่ 2 อาจ Select มาจากฐานข้อมูล Oracle ตาราง Active memory ก็ได้

นี่คือความสามารถของ ADO.net ซึ่ง Dataset จะมองเหมือนฐานข้อมูลทั้งก้อนในหน่วยความจำของเรานั่นเอง

7.2.4 ออบเจ็กต์ DataAdapter

ออบเจ็กต์ DataAdapter มีเฉพาะใน ADO.net เช่นกัน ออบเจ็กต์ DataAdapter ก็เหมาะสำหรับการ Select ทั่วๆ ไป ออบเจ็กต์ DataAdapter เปรียบเสมือนเป็นสะพานเชื่อมระหว่าง DataSource ทีมีข้อมูลกับตัว Dataset ของเรา โดยลำเลียงข้อมูลจาก Data Source เอามาเก็บใน DataSEr จากนั้นจึงสามารถใช้คำสั่ง SQL ใน Dataset ทำการเปลี่ยนแปลงข้อมูลในแถวนั้นๆ ได้ และหากต้องการซิงค์ข้อมูลจาก Dataset กลับไปใน Data Source ใหม่ เราก็ใช้ DataAdapter ทำการอัพเดตข้อมูล กลับขึ้นไปใน Data Source

DataAdapter มีเมธอด อยู่ 2 ตัว คือ fill กับ update

- Fill : ใช้อ่านข้อมูลจาก Data Source กลับมาไว้ใน Dataset
- Update: ใช้ดึงข้อมูลกลับ

หากเปรียบภาพของ DataAdapter ก็เหมือนสะพานเชื่อมโยงระหว่างฐานข้อมูลของเรากับ Dataset นั่นเอง

ตัวอย่างการสร้าง DataAdapter ในภาษา visual C#

```
Dataset pubs;
SQLDataAdapter objDataAdapter = new SQLDataAdapter( “Select * from authors “,cnn);

ObjDataAdapter.Fill(pubs,”authors”);
Pubs.Tables[“Authors”].Rows[0][“au_lname”]=’smith”;
ObjDataAdapter.update(pubs,”authors”);
```

จากตัวอย่าง การที่จะเอาข้อมูลมาเก็บไว้ใน Dataset เราต้องสร้างออบเจ็กต์ขึ้นมา 2 ตัว ตัวแรกเป็น Dataset ชื่อ pubs ตัวที่ 2 เป็น Dataadapter หลังสร้างเสร็จแล้ว ให้ใช้คำสั่ง SQL เพื่อดึงข้อมูลมา จากนั้นใช้คำสั่ง Fill เพื่อนำข้อมูลที่ได้จากการอ่านเข้ามาใส่ใน Dataset

เวลาที่เราอ้างถึงแถวหรือคอลัมน์ต่างๆ นั้น ต้องทำตามรูปแบบโครงสร้างดังนี้

ชื่อ Dataset ตามด้วย Tables แล้วก็ใช้ชื่อของตาราง ตามด้วย Rows และอินเด็กต์ของแถวที่ต้องการ และตามด้วยชื่อของคอลัมน์ เช่น

```
Pubs.Tables["Authors"].Rows[0]["au_lanme"]
```

จากตัวอย่างเป็นการอ้างถึง Dataset ชื่อ pubs ที่ตาราง Athors ในแนวแรกของตาราง และคอลัมน์ชื่อว่า au_lname

7.3 การสนับสนุนมาตรฐาน XML

การสนับสนุนมาตรฐาน XML เป็นคุณลักษณะหนึ่งของ ADO.net ที่เพิ่มขึ้นมาจาก ADO ซึ่ง XML เป็นเทคโนโลยีพื้นฐานอยู่เบื้องหลังไมโครซอฟท์อย่างมาก เพราะ Web Service ก็ใช้ XML รับส่งข้อมูล มีฟังก์ชั่นอยู่ 4 ตัว ที่เกี่ยวกับ XML คือ

- Read XML ()
- Read XML Schema ()
- Write XML ()
- Write XML Schema ()

เพื่อใช้ในการอ่านข้อมูลไฟล์ XML มาเก็บใน Dataset โดยรวมกับ Schema ไว้ใน Dataset หรือว่าจะเขียนลงเป็น XML ใน DataSEt ก็ได้

7.4 สรุปตารางเปรียบเทียบ ระหว่าง ADO กับ ADO.net

คุณลักษณะ	ADO	ADO.net
สถาปัตยกรรมที่ใช้	บนพื้นฐาน COM การเชื่อมโยงแบบ connection orientd	XML กับ HTTP การเชื่อมโยงแบบ Connectionless
การเข้าถึงระบบป้องกัน Firewall	ไม่ผ่านระบบ Firewall	ใช้ HTTP และสามารถ ผ่าน Firewall ได้เป็นอย่างดี
ออบเจ็กต์ที่ใช้	Connection Command Recordset	Connection Command Dataset DataSetview DataView เป็นต้น

ตารางที่ 7-1 สรุปตารางเปรียบเทียบ ระหว่าง ADO กับ ADO.net