

BỘ GIÁO DỤC VÀ ĐÀO TẠO

VISUAL STUDIO VÀ DATA ACCESS
PHẦN 1



NHÀ XUẤT BẢN TRẺ - 2010

Visual Studio và Data Access – Phần 1

Với phiên bản mới của Visual studio đưa ra vài cách mới để truy cập dữ liệu trong các ứng dụng của bạn. Phần này sẽ bàn luận về một số cách mà Visual Studio.NET cho phép dữ liệu được hợp nhất trong GUI, để bạn có thể tương tác với dữ liệu.

Các công cụ cho phép bạn tạo một sự kết nối cơ sở dữ liệu là sử dụng các lớp *OleDbConnection* hay *SqlConnection*. Lớp mà bạn sẽ dùng phụ thuộc vào cơ sở dữ liệu nào bạn muốn kết nối. Khi định nghĩa một sự kết nối, bạn có thể tạo một *DataSet* và định vị nó từ bên trong Visual studio.NET. Vấn đề này sẽ tạo ra một tập tin XSD cho *DataSet* như là chúng ta đã làm bằng

tay trong chương trước và tự động phát ra các mã .cs cho bạn. Kết quả này nằm trong sự tạo thành của một type-safe *DataSet*.

Trong phần này ta sẽ học cách tạo một sự kết nối, chọn một số dữ liệu và tạo ra một *DataSet*, và sử dụng tất cả đối tượng được tạo ra để làm một ứng dụng đơn giản.

Tạo một sự kết nối

Để bắt đầu phần này, ta phải tạo một ứng dụng Windows. Khi tạo bạn sẽ thấy một form trống. Công việc đầu tiên là tạo một sự kết nối cơ sở dữ liệu mới. Mở *Server Explorer* bằng cách gõ *Ctrl+Alt+S* hay chọn mục *Server Explorer* từ menu. Cửa sổ sẽ hiển thị như sau:

Trong cửa sổ này bạn có thể quản lý nhiều khía cạnh khác nhau của việc truy cập dữ liệu. Theo ví dụ này, bạn cần tạo một sự kết nối đến cơ sở dữ liệu *Northwind*. Chọn *Add Connection...* từ menu trên mục *Data Connections* sẽ tự động hiện lên một trình thông minh để bạn có thể chọn OLEDB

provider nào được dùng- ở đây ta chọn *Microsoft OLEDB Provider* cho SQL server, khi bạn sẽ được kết nối với cơ sở dữ liệu *Northwind* được cài đặt như một phần của mẫu Framework SDK. Trang thứ hai của hộp thoại *Data Link* như sau:

Phụ thuộc vào cách bạn cài đặt các cơ sở dữ liệu mẫu Framework thì bạn sẽ có một thể hiện của cơ sở dữ liệu *Northwind* trong SQL Server, và một thể hiện trong một cơ sở dữ liệu local MSDE (Microsoft Data Engine), hay cả hai.

Để kết nối với cơ sở dữ liệu MSDE thì gõ *(local)\NETSDK* và tên của server. Để kết nối một thể hiện của SQL server bạn gõ *(local)* như hiện ở trên cho bộ máy hiện tại hay tên của server muốn kết nối trên mạng.

Tiếp theo, bạn cần chọn thông tin đăng nhập. Bạn phải chọn lại một lần nữa phụ thuộc vào cách cơ sở dữ liệu của bạn được cài đặt. Đối với cơ sở dữ liệu local MSDE, bạn có thể dùng một username và Password đặc biệt tương ứng với *QSUser* và *QSPassword*.

Chọn cơ sở dữ liệu *Northwind* từ danh sách cơ sở dữ liệu, và để chắc rằng bạn đã cài đặt mọi thứ chính xác thì click vào nút *Test Connection*. Hành động này sẽ kết nối cơ sở dữ liệu và hiện một hộp tin khi hoàn tất. Dĩ nhiên, bạn phải cài server trên cấu hình của máy bạn. vì thế Username, password và tên server sẽ khác nhau.

Để tạo một đối tượng kết nối, click và kéo server mới đến cửa sổ ứng dụng chính. Nó sẽ tạo một biến thành viên của kiểu *System.Data.SqlClient.SqlConnection*, hay *System.Data.OleDb.OleDbConnection* nếu bạn chọn một provider khác và thêm đoạn mã sau vào phương thức *InitializeComponent* của form chính:

```
this.sqlConnection1 = new System.Data.SqlClient.SqlConnection();
```

```
//  
  
// sqlConnection1  
  
//  
  
this.sqlConnection1.ConnectionString = "data source=skinnerm\\NETSDK;"  
+  
    "initial catalog=Northwind;" +  
    "user id=QSUer;password=QSPassword;" +  
    "persist security info=True;" +  
    "workstation id=SKINNERM;" +  
    "packet size=4096";
```

Như bạn thấy, sự kết nối thông tin chuỗi được gắn trực tiếp trong đoạn mã.

Khi bạn thêm đối tượng này và dự án bạn sẽ chú ý đối tượng *sqlConnection1* xuất hiện trong vùng bên dưới của cửa sổ visual studio.

Chọn dữ liệu

Khi bạn định nghĩa một sự kết nối dữ liệu, bạn có thể chọn một bản từ danh sách và kéo bảng đó đến một form trên dự án của bạn.

Ví dụ, ta chọn bảng *Customer*. khi bạn kéo đối tượng này vào dự án của bạn nó sẽ thêm một đối tượng vào form của bạn được thừa hưởng từ *SqlDataAdapter*, hay *OleDbDataAdapter* nếu bạn không dùng SQL Server.

Data adapter đã tạo ra chứa đựng các lệnh SELECT, INSERT, UPDATE, và DELETE. Đoạn mã tạo trình thông minh sẽ thực hiện ngay lúc này nhưng visual studio.NET thêm đoạn mã sau vào tập tin .cs của bạn.

```
private System.Data.SqlClient.SqlCommand sqlSelectCommand1;
```

```
private System.Data.SqlClient.SqlCommand sqlInsertCommand1;
```

```
private System.Data.SqlClient.SqlCommand sqlUpdateCommand1;
```

```
private System.Data.SqlClient.SqlCommand sqlDeleteCommand1;
```

```
private System.Data.SqlClient.SqlDataAdapter sqlDataAdapter1;
```

Có một đối tượng đã định nghĩa cho mọi lệnh SQL và một *sqlDataAdapter*. Trong phương thức *InitializeComponent()*, trình thông minh tạo ra đoạn mã để tạo mọi lệnh này và data adapter. Đoạn mã thì đông dài, vì thế tôi chỉ đưa ra một đoạn ở đây.

Có hai khía cạnh của đoạn mã được tạo bởi Visual studio.NET là các giá trị được nhìn thấy từ các thuộc tính *UpdateCommand* và *InsertCommand*. Đây là một phiên bản tóm tắt hiện thông tin thích đáng:

```
// sqlInsertCommand1
```



```
//
```

```
this.sqlInsertCommand1.CommandText = @"INSERT INTO  
dbo.Customers
```

```
(CustomerID, CompanyName, ContactName,
```

```
ContactTitle, Address, City, Region,
```

```
PostalCode, Country, Phone, Fax)
```

```
VALUES(@CustomerID, @CompanyName, @ContactName,  
@ContactTitle,
```

```
@Address, @City, @Region, @PostalCode, @Country, @Phone,  
@Fax);
```

```
SELECT CustomerID, CompanyName, ContactName, ContactTitle,  
Address,
```

```
City, Region, PostalCode, Country, Phone, Fax
```

```
FROM dbo.Customers WHERE (CustomerID =  
@Select2_CustomerID);
```

```
this.sqlInsertCommand1.Connection = this.sqlConnection1;
```

```
//
```

```
// sqlUpdateCommand1
```

```
//
```

```
this.sqlUpdateCommand1.CommandText = @"UPDATE dbo.Customers
```

```
    SET    CustomerID    =    @CustomerID,    CompanyName    =
```

```
@CompanyName,
```

```
    ContactName = @ContactName, ContactTitle = @ContactTitle,
```

```
    Address = @Address, City = @City, Region = @Region,
```

```
    PostalCode = @PostalCode, Country = @Country,
```

```
    Phone = @Phone, Fax = @Fax
```

```
WHERE (CustomerID = @Original_CustomerID)
```

```
AND (Address    =    @Original_Address) AND (City    =
```

```
@Original_City)
```

```
AND (CompanyName = @Original_CompanyName)
```

```
AND (ContactName = @Original_ContactName)
```