

1. Wstęp	
1.1 Wprowadzenie do ADO.NET .....	3
1.2 Architektura ADO.NET .....	3
2. Środowisko .....	5
3. Przykładowy program	
3.1 Tworzenie modelu .....	7
3.2 Odczytywanie i zapisywanie danych .....	7
4. Projekt	
4.1 Rozwiązywany problem.....	8
4.2 Obsługa programu .....	8
5. Literatura	

# 1. Wstęp

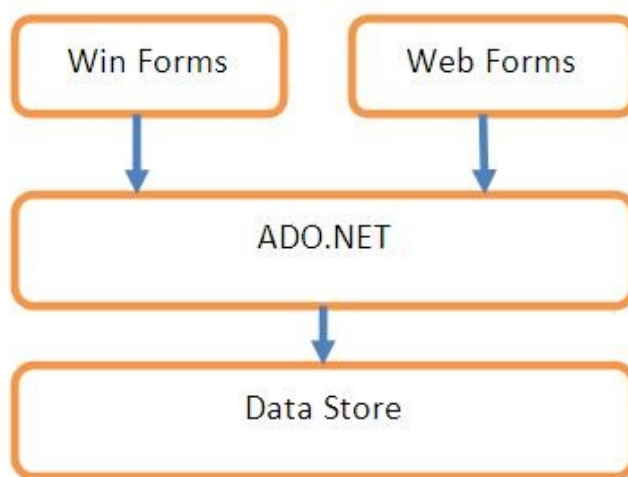
## 1.1 Wprowadzenie do ADO.NET

ADO.NET jest zestawem klas, które ujawniają usługi dostępu do danych dla programistów .NET Framework. ADO.NET zapewnia bogaty zestaw składników do tworzenia aplikacji rozproszonych, udostępnianie danych. Jest integralną częścią programu .NET Framework, zapewniając dostęp do relacyjnych, XML i dane aplikacji. ADO.NET obsługuje szereg rozwoju, w tym tworzenie frontonu bazy danych klientów i używanych przez aplikacje, narzędzi, języków lub przeglądark internetowych obiektów firm warstwy środkowej. ADO.NET zapewnia spójny dostęp do źródeł danych, takie jak SQL Server i XML i źródła danych dostępne za pośrednictwem OLE DB i ODBC. Udostępnianie danych aplikacji komercyjne umożliwia ADO.NET nawiązać połączenie z tymi źródłami danych i pobierania, obsługiwać i zaktualizować dane, które zawierają.

Dostęp do danych w ADO.NET są oddzielone od manipulowanie danymi na osobne składniki, których można użyć pojedynczo lub w tandem. ADO.NET zawiera dostawcy danych .NET Framework do nawiązywania połączenia z bazą danych, wykonując polecenia i pobierania wyników. Te wyniki są albo przetwarzane bezpośrednio, umieszczone w ADO.NET DataSet obiektu w celu będzie widoczna dla użytkownika w sposób ad-hoc w połączeniu z danymi z wielu źródeł lub przekazywane między warstwami. DataSet Obiektu można również używać niezależnie od dostawcy danych .NET Framework do zarządzania danymi lokalne dla aplikacji lub skąd pochodzi z pliku XML.

## 1.2 Architektura ADO.NET

Poniżej wizualizacja dostępu do danych za pomocą ADO.NET



source: codeproject.com

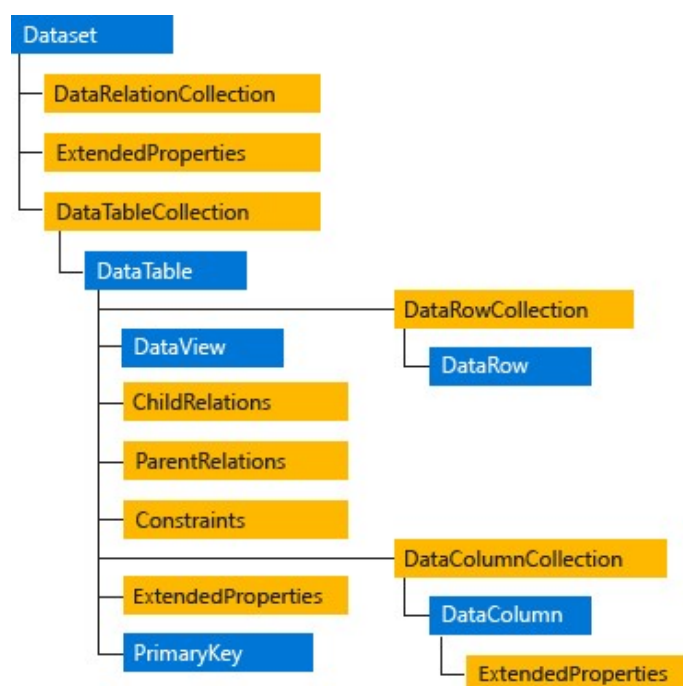
Rys.1.2.1 Wizualizacja dostępu do danych za pomocą ADO.NET

Jak widać na najwyższym poziomie znajduje się aplikacja, może to być projekt Win Forms, Web Forms, Silverlight, aplikacja konsolowa, projekt Web API itp. Na najniższym poziomie znajduje się warstwa bazy danych w której możemy przechowywać, zmieniać oraz

wyszukiwać dane. Warstwa **ADO.NET** pełni rolę pośrednika pomiędzy tymi warstwami udostępniając wystandaryzowany szereg narzędzi.

Dostawcy danych .NET Framework są składnikami, które zostały jawnie zaprojektowane do manipulowania danymi i szybkie tylko do przodu, tylko do odczytu dostępu do danych. **Connection** Obiektu zapewnia łączność ze źródłem danych. **Command** Obiekt umożliwia dostęp do bazy danych poleceń, aby zwrócić dane, modyfikowania danych, uruchamianie procedur składowanych i wysyłać lub odbierać informacje o parametrach. **DataReader** Zapewnia strumienia o wysokiej wydajności, dane ze źródła danych. Na koniec **DataAdapter** zapewnia połączenie między **DataSet** obiektu i źródła danych. **DataAdapter** Używa **Command** obiektów do wykonywania poleceń SQL w źródle danych, zarówno obciążenie **DataSet** z danymi i uzgadniają zmiany, które zostały wprowadzone do danych w **DataSet** wstecz do źródła danych.

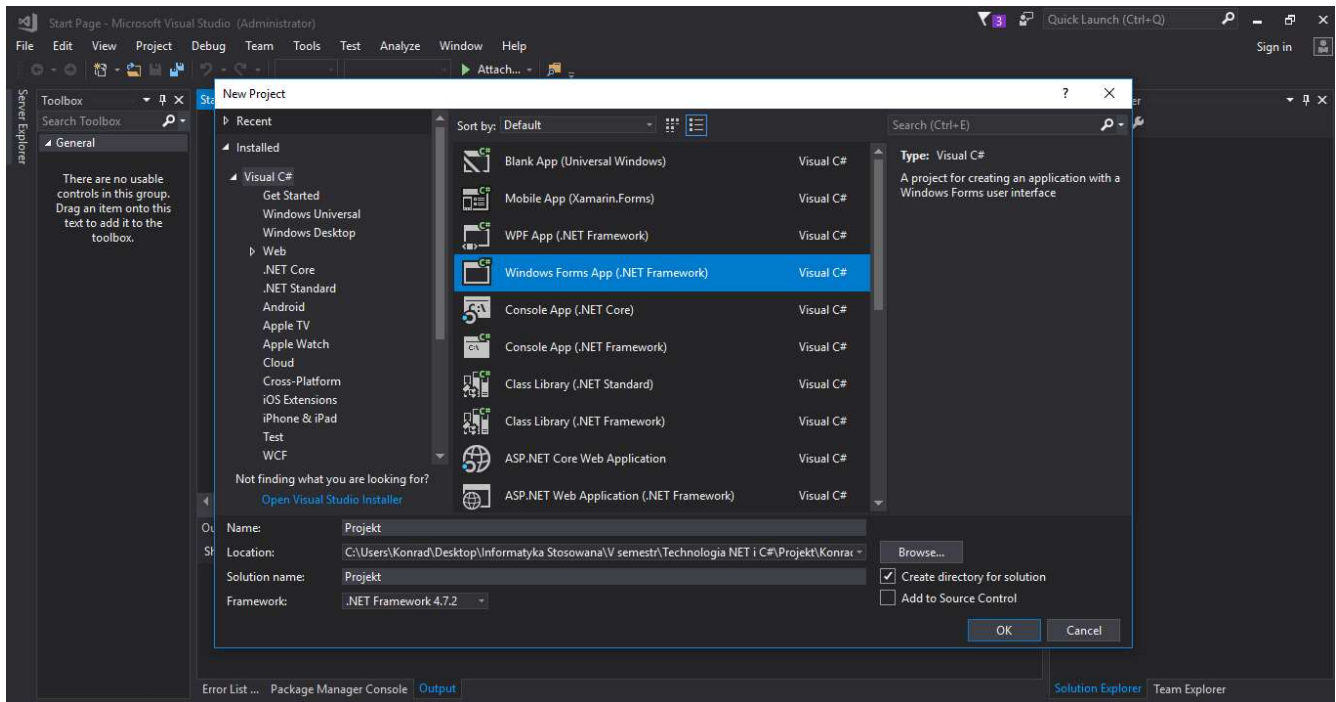
**ADO.NET DataSet** jawnie zaprojektowano pod kątem dostępu do danych niezależnie od dowolnego źródła danych. W rezultacie mogą być używane z wieloma i danych różnych źródeł, używany z danymi XML lub używany do zarządzania danymi lokalne dla aplikacji



Rys.1.2.2 Relacja między .NET Framework dostawcy danych i DataSet

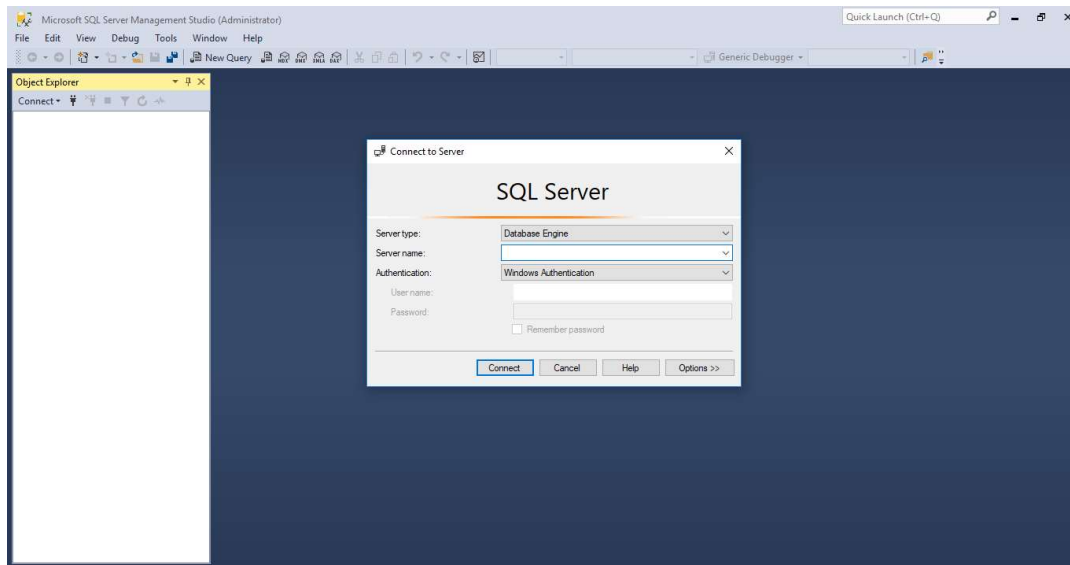
## 2. Środowisko

W trakcie tworzenia projektu będziemy potrzebowali dwóch programów. Pierwszym z nich jest Visual Studio w wersji 2017. Aby utworzyć nową aplikację uruchamiamy środowisko programistyczne, następnie wybieramy z głównego menu: **Plik -> Nowy -> Project** wybieramy **Windows Forms App (.NET Framework)**, wprowadzamy nazwę swojego projektu, wybieramy ścieżkę zapisu naszej aplikacji i klikamy **OK**.



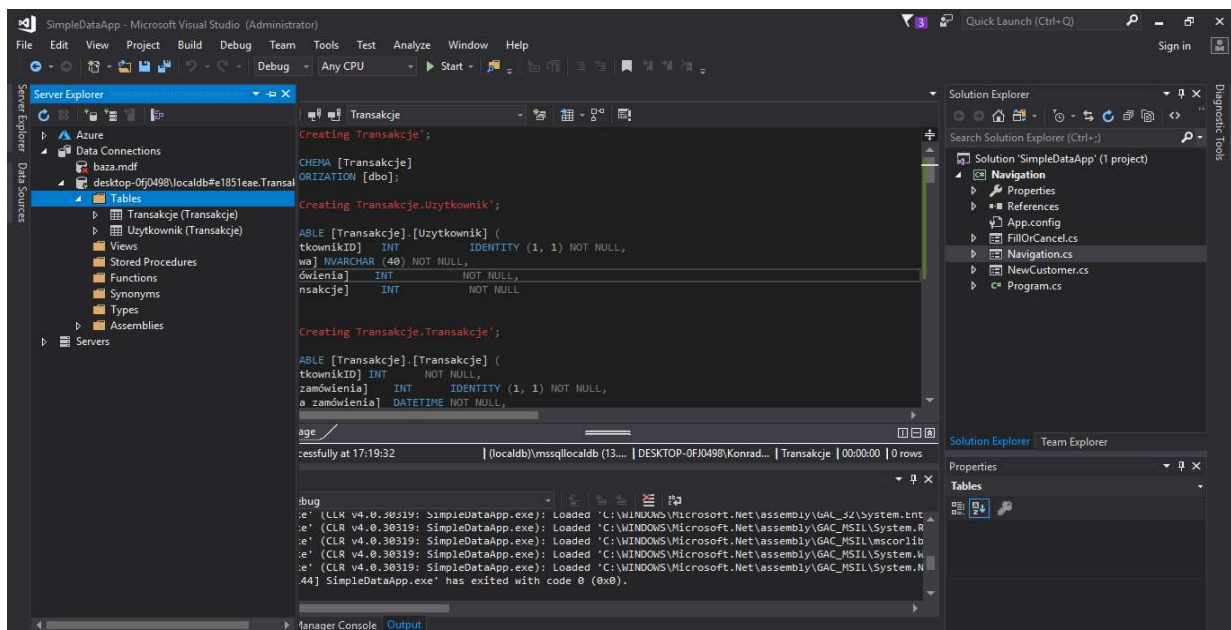
Rys.2.1 Zrzut ekranu - utworzenie nowego projektu Visual Studio 2017

Drugim programem jest Microsoft SQL Server Management Studio 2017, którym tworzymy bazę danych potrzebną do funkcjonowania naszego projektu.



Rys.2.2 Zrzut ekranu - Okno łączenia z serwerem - Microsoft SQL Server Management Studio 2017

Jeżeli nie mamy powyższego programu możemy skorzystać ze zintegrowanej opcji w środowisku programistycznym Visual Studio. Zakładka **View -> Server Explorer**.



Rys.2.3 Zrzut ekranu - utworzenie nowej bazy danych Visual Studio 2017/Server Explorer

## 3. Przykładowy program

### 3.1 Tworzenie modelu

Na początku definiujemy prosty model przy użyciu dwóch klas. Klasy piszemy w jednym pliku ale będą one widnieć jako osobne klasy. Dodajemy instrukcję **“using System.Data.Entity”** w górnej części pliku.

### 3.2 Odczytywanie i zapisywanie danych

Wprowadzamy metody w “Main”. Następnie używamy zapytania LINQ, aby pobrać wszystkie piosenki z bazy danych. Piosenki zostały uporządkowane w kolejności alfabetycznej według swoich tytułów.

```
namespace przyklad
{
    static class Program
    {
        static void Main()
        {
            Console.WriteLine("Wprowadź nazwę piosenki: ");
            var nazwa = Console.ReadLine();

            var piosenka = new piosenka { Name = nazwa };
            db.piosenki.Add(piosenka);
            db.SaveChanges();

            var query = from b in db.piosenki
                        orderby b.Name
                        select b;
            Console.WriteLine("Wszystkie piosenki w bazie danych:");
            foreach (var item in query)
            {
                Console.WriteLine(item.Name);
            }

            Console.WriteLine("Naciśnij dowolny klawisz, aby wyjść");
            Console.ReadKey();
        }
    }
}
```

Rys.3.2.1 Zrzut ekranu - część kodu przykładowy program - Visual Studio 2017

Po utworzeniu powyższej części kodu możemy nacisnąć klawisz F5 i sprawdzić czy poprawnie został uruchomiony program

Na poniższym zrzucie ekranu widzimy, że program poprawnie wykonał wszystkie operacje.

```
Wprowadź nazwę piosenki: Marek Grechuta Korowod  
Wszystkie piosenki w bazie danych:  
Marek Grechuta Korowod  
Naciśnij dowolny klawisz, aby wyjść
```

Rys.2.3 Zrzut ekranu - Wykonanie programu Visual Studio 2017

## 4. Projekt

### 4.1 Rozwiązywany problem

Projekt będzie w prosty sposób przedstawiał technologię ADO.NET. Dzięki aplikacji która została przedstawiona poniżej mała firma może zarządzać swoimi zamówieniami. Możemy dodawać Konta użytkowników, kompletować oraz anulować zamówienia.

Środowisko:

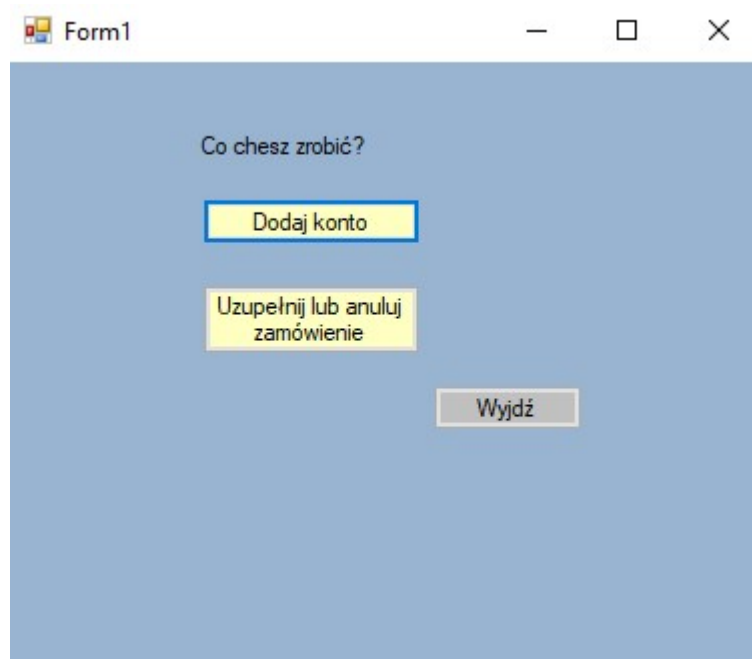
- Microsoft Visual Studio 2017
- Microsoft SQL Server Management Studio 2017

### 4.2 Obsługa programu

Wszystkie wprowadzone dane zostają zapisane w bazie danych. Pracownik może Dodać konto, wprowadzić nazwę użytkownika, wybrać odpowiednią ilość sztuk zamawianego towaru i zapisać całość w bazie. Każde nowo dodane zamówienie dostaje automatycznie swój indywidualny numer ID.

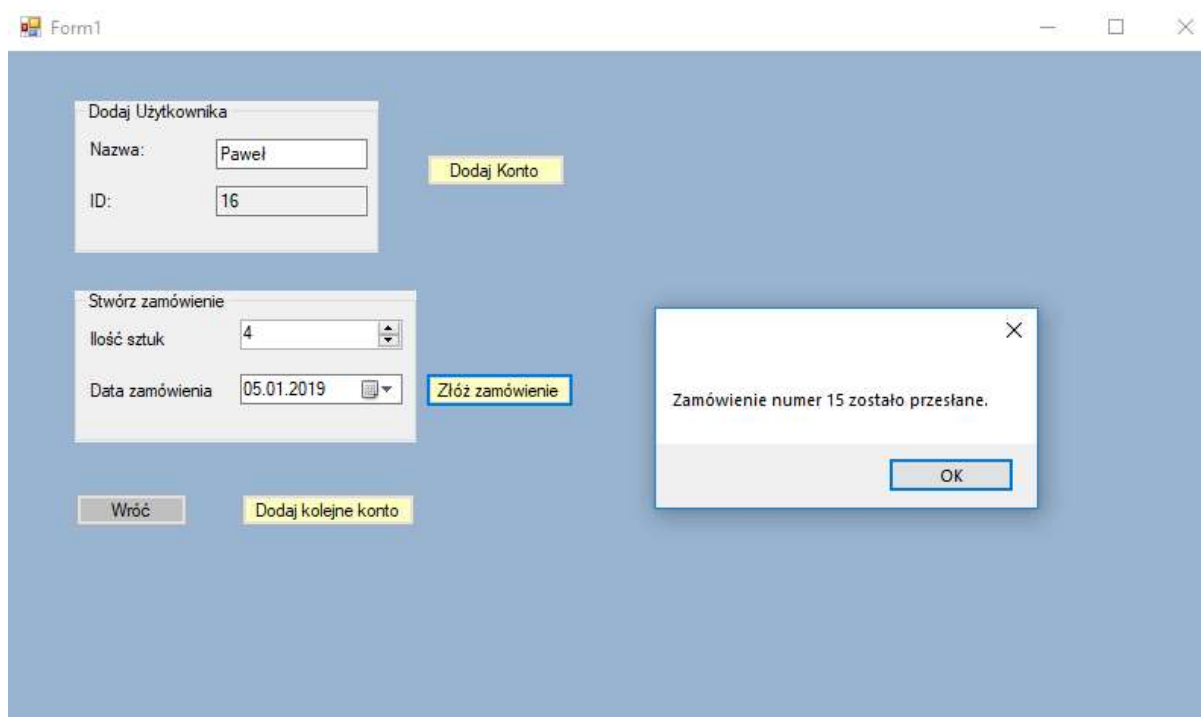
Jeżeli zamówienie zostało zamówione, pracownik który kompletuje zamówienie przystępuje do pakowania produktu, a następnie wprowadza w bazie danych zmiany: wprowadza datę wypełnienia zamówienia. Jeżeli zamówienie zostało zrealizowane pracownik wybiera opcję “Wypełnij zamówienie” jeżeli nie - “Anuluj zamówienie”. W każdej chwili można również wrócić do poprzedniego okna klikając przycisk “wróć”.

Po uruchomieniu aplikacji pojawia nam się pierwsze okno z kilkoma możliwościami: **Dodaj konto, Uzupełnij lub anuluj zamówienie.** Możemy wyjść z programu za pomocą przycisku **Wyjdź**.



Rys.4.2.1 Zrzut ekranu - Okno początkowe - wyboru

Po wybraniu pierwszej opcji dodajemy użytkownika:



Rys.4.2.2 Zrzut ekranu - Dodanie użytkownika do bazy danych

Wypełniamy dane, po wprowadzeniu Nazwy i kliknięciu w przycisk **Dodaj Konto**, automatycznie zostanie wprowadzone ID naszego nowego klienta.



Poniżej znajduje się “GroupBox” z możliwością wprowadzenia ilości sztuk danego produktu oraz Data złożenia zamówienia oraz przycisk **Złóż zamówienie**. W tym oknie możemy również **Dodać kolejne konto** lub wrócić do pierwotnego okna.

Jeżeli jednak chcemy uzupełnić lub anulować nasze zamówienie, klikamy w odpowiedni przycisk, który przenosi nas do nowego okna:

OrderID	OrderDate	FilledDate	Status
15	05.01.2019 21:17	05.01.2019 21:18	X

Rys.4.2.3 Zrzut ekranu - Edycja zamówienia

W powyższym oknie po wprowadzeniu ID zamówienia i kliknięciu w przycisk **Znajdź zamówienie** dostajemy odpowiedź w “DataGridView” z naszym ID klienta, ID zamówienia, datą zamówienia, datą wypełnienia zamówienia, statusie oraz ilości zamówionego towaru. Po kliknięciu w przycisk **Anuluj zamówienie** zostanie zmieniony status zamówienia na “X”, jednak jeżeli klikniemy w **Wypełnij zamówienie** dostaniemy status “F”. Możemy wrócić do pierwotnego okna po kliknięciu w przycisk **Wróć**.

## Literatura

- Programowanie w .NET, “C# - *ADO.NET*”, 12.27.2018  
[https://www.plukasiewicz.net/Artykuly/ADO\\_NET](https://www.plukasiewicz.net/Artykuly/ADO_NET)
- Wikipedia, “*ADO.NET*” 5.09.2018  
<https://en.wikipedia.org/wiki/ADO.NET>
- Język C# 2010 i platforma .NET 4 **Troelsen Andrew**
- Barker F.Scott Visual Basic.NET and ADO.NET Tips, Tutorials and Code