

ADO.NET. ТЕХНОЛОГИЯ И СРЕДСТВА ДОСТУПА К БАЗАМ ДАННЫХ**ADO.NET. TECHNOLOGIES AND MEANS OF ACCESS TO DATABASES**

Бул макалада заманбап маалымат бериштери менен байланышууга мүмкүнчүлүк берүүчү ADO.NET технологиясы каралып, маалымат алуу учурунда аны ажыратуу мүмкүнчүлүгү ишке ашырылды. Мында ADO.NETтин объектик модели жана анын компоненттери көрсөтүлгөн. Бул технологияны колдонуу менен маалымат базасын тузүү мисалы каралган.

Ачык сөздөр: *ADO.NET, DataSet, DataProvider, Connection, Command, DataAdapter, DataReader.*

В данной статье рассмотрена современная технология доступа к базам данных – ADO.NET, позволяющая осуществить отсоединенный доступ к данным. Представлена объектная модель ADO.NET и ее компоненты. Рассмотрен пример соединения с базой данных.

Ключевые слова: *ADO.NET, DataSet, DataProvider, Connection, Command, DataAdapter, DataReader.*

The article is devoted to the modern technologies of data basis access – ADO.NET, which allows disconnected access to data- Presented object model ADO.NET and its components. The example of connecting with DB, using ADO.NET technology.

Keywords: *ADO.NET, DataSet, DataProvider, Connection, Command, DataAdapter, DataReader.*

ADO.NET. DataSet, DataProvider, Connection, Command, DataAdapter, DataReader.

В настоящее время для обработки информации, хранящейся в базах данных, используется множество разнообразных систем управления базами данных (СУБД), в связи с чем, возникает необходимость обеспечивать поддержку различных источников данных. Также в различных языках программирования различна поддержка работы с той или иной СУБД. Все это создает множество проблем. Для их решения компанией Microsoft была разработана новая мощная технология доступа к данным – ADO.NET.

ADO.NET – это набор библиотек, поставляемый с Microsoft .NET Framework и предназначенный для взаимодействия с различными хранилищами данных из .NET-приложений. Библиотеки ADO.NET включают классы для подсоединения к источнику данных, выполнения запросов и обработки их результатов. Кроме того, ADO.NET можно использовать в качестве надежного, иерархически организованного, отсоединенного кэша данных для автономной работы с данными /1/.

ADO.NET – новый этап в технологии ActiveXDataObjects. В ADO.NET обеспечивается возможность работы со всеми совместимыми с OLE DB источниками данных как в локальных сетях в рамках традиционных Windows-приложений, так и в глобальных сетях Интернет в рамках Web-приложений. Кроме того в ADO.NET изначально заложена возможность работы приложения в состоянии «разрыва» соединения с базой данных. Клиент устанавливает соединение, получает необходимые данные, закрывает соединение, длительное время обрабатывает данные, после чего вновь устанавливает соединение для передачи измененных данных обратно на сервер.

Концепция доступа к данным в ADO .NET основана на использовании двух компонентов:

- набора данных (представляется объектом класса DataSet) со стороны клиента. Это локальное временное хранилище данных;

- провайдера данных (представляется объектом класса DataProvider). Это посредник, обеспечивающий взаимодействие приложения и базы данных со стороны базы данных (в распределенных приложениях – со стороны сервера).

На рис 1. отображена диаграмма, представляющая объектную модель ADO.NET.

Как видно из диаграммы провайдер данных отвечает за связь приложения с источником данных и манипуляцию данными. Основные объекты, входящие в его состав, – это Connection, Command, DataAdapter и DataReader.

Компонент Connection используется для установления соединения с источником данных, а также для управления транзакциями.

Объект Command позволяет манипулировать данными источника, а также выполнять хранимые процедуры. В случае необходимости могут использоваться параметры для передачи данных в обоих направлениях.

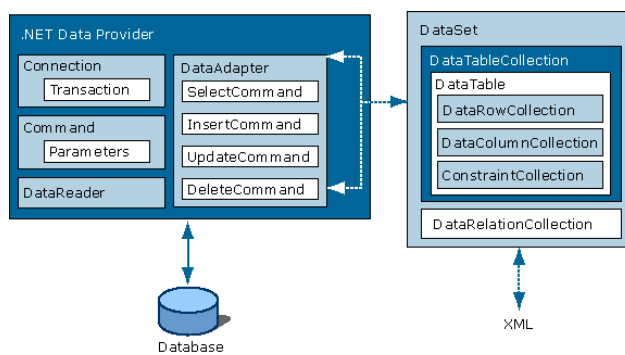


Рис.1. Объектная модель ADO.NET

Для связи между DataSet и источником данных служит компонент DataAdapter. Он использует Command для выполнения команд SQL как для заполнения DataSet данными, так и для обратной передачи измененных клиентом данных к источнику. Для выполнения этих функций объект имеет 4 метода: выборка– SelectCommand; вставка– InsertCommand; обновление– UpdateCommand; удаление– DeleteCommand.

DataReader представляет односторонний поток данных от источника только на чтение и используется если приложение клиента не модифицирует данные и ему достаточно их просмотра.

DataSet представляет собой класс ADO .NET, который отвечает за отображение таблицы (или таблиц, или даже всей базы) используемой базы данных, на компьютере пользователя без непрерывной связи с базой данных. В DataSet загружаются данные, после чего соединение с их источником (или источниками, которых в принципе может быть несколько, поскольку в одном экземпляре DataSet могут объединяться данные из нескольких источников) может быть разорвано. Далее клиент в автономном режиме производит обработку данных, при необходимости модифицирует их, после чего снова устанавливается соединение с источником и модифицированные данные передаются обратно /2/.

Внутреннее устройство DataSet представлено диаграммой, отраженной на рис.2.

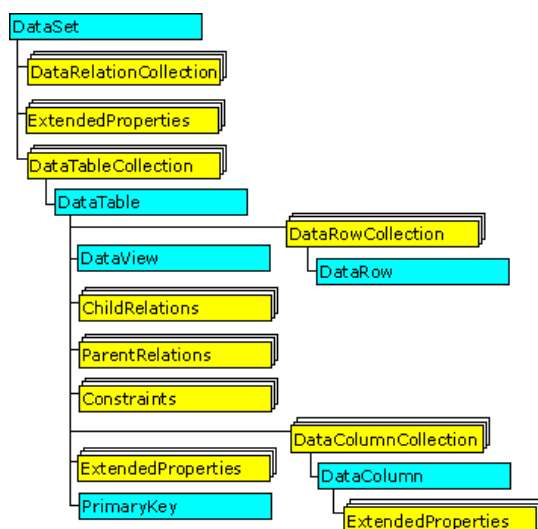


Рис.2. Внутреннее устройство DataSet

Рассмотрим вкратце наиболее важные компоненты DataSet, не углубляясь во второстепенные детали.

DataSet включает две основные коллекции: DataTable Collection и DataRelation Collection.

DataTableCollection это коллекция объектов DataTable, каждый из которых представляет одну таблицу отношения и в свою очередь состоит из коллекций:

1. DataColumnCollection представляет все столбцы таблицы. Помимо данных, столбец может включать формулу для динамического расчета своего содержимого.
2. DataRowCollection представляет набор строк таблицы (возможно, пустой). Следует отметить, что хранятся не только текущие значения данных, но и первоначальные, что позволяет выделить измененные данные.
3. ConstraintCollection набор ограничений целостности данных (например, можно задать, допустимо ли для данного столбца значение NULL или должны ли данные быть уникальными).

Другая коллекция из состава DataSet – это DataRelationCollection, которая служит для навигации по связанному набору таблиц и содержит ограничения ссылочной целостности DataSet. В ней, например, можно задать ограничения, которое не позволит удалить запись главной таблицы, если на нее есть ссылки в подчиненной таблице посредством внешнего ключа /3/.

Как и любая другая технология, ADO .NET состоит из нескольких важных компонентов. Все классы .NET группируются в пространства имен. Все функции, относящиеся к ADO .NET, находятся в пространстве имен System.Data.

ADO.NET поставляется с тремя пространствами имен клиента базы данных: одно для SQL Server, другое для источников данных OpenDatabaseConnectivity (ODBC) и третье для любой базы данных, доступной через OLE DB. С точки зрения программиста, тело ADO.NET составляет базовая сборка с именем System.Data.dll. В этом двоичном файле находится значительное количество пространств имен, многие из которых представляют типы конкретного поставщика данных ADO.NET.

Библиотеки ADO.NET можно применять тремя концептуально различными способами: в подключенном режиме, в автономном режиме и с помощью технологии EntityFramework. При использовании **подключенного уровня (connectedlayer)**, кодовая база явно подключается к соответствующему хранилищу данных и отключается от него. При таком способе использования ADO.NET обычно происходит взаимодействие с хранилищем данных с помощью объектов подключения, объектов команд и объектов чтения данных.

Получив объект DataSet, вызывающий код может просматривать и обрабатывать данные без затрат на сетевой трафик. А если нужно занести изменения в хранилище данных, то адаптер данных (вместе с набором операторов SQL) задействуется для обновления данных — при этом подключение открывается заново для проведения обновлений в базе, а затем сразу же закрывается.

Рассмотрим пример приложения ADO.NET, выполняющий соединение с базой данных, созданной в приложении MSAccess.

Для перемещения данных между их постоянным хранилищем и приложением в первую очередь необходимо создать соединение с источником данных (Connection). В арсенале ADO.NET для этих целей имеется объект *OleDbConnection* – объект, позволяющий создать соединение с любым источником данных (простые текстовые файлы, электронные таблицы, базы данных) через OLEDB/4/.

В разрабатываемом приложении для подключения объекта OleDbConnectionиспользуется следующий код:

```
.....  
Using System.Data.OleDb;
```

```
.....
```

Первое свойство объекта соединения, которое необходимо определить в блоке инициализации для установления связи с базой данных – это строка соединения *ConnectionString*. В строке соединения управляемых поставщиков необходимо, как минимум, указать местоположение базы данных и требуемую аутентификационную информацию. Помимо этого, каждый поставщик данных определяет дополнительные параметры соединения. Если в строке соединения не указаны значения всех возможных параметров, они считаются установленными по умолчанию. В разрабатываемом приложении используется провайдер – Microsoft.Jet.OLEDB.4.0 /5/.

Ниже представлен код программы, выполняющий подключение к базе данныхи реализующий выборку данных из двух связанных таблиц (Магистрант, Группа) базы данных отдела магистратуры КГУСТА, разработанной в приложении MSAccess.

```
...  
NamespaceMagistr  
{  
public partial class Form3 : Form  
{  
  
String ConnectionString = @"Provider=Microsoft.Jet.OLEDB.4.0; "+"Data Source =  
C:/Visual Studio 2008/Projects/Magistr/Magistr/Магистратура.mdb";  
DataSet ds = new DataSet();  
  
private void button1_Click(object sender, EventArgs e)  
{  
stringCommandText = "SELECT Magistr.Familia,Magistr.Ima,Magistr.Telefon,  
Gruppa.NazvaznieGruppi" +  
"FROM Magistr INNER JOIN Gruppa ON Magistr.GruppaID = Gruppa.GruppaId ";  
OleDbDataAdapterdataAdapter = new OleDbDataAdapter(CommandText,  
ConnectionString);  
  
dataAdapter1.Fill(ds1, "Gruppa,Magistr");  
dataGridView1.DataSource= ds1.Tables["Gruppa,Magistr"].DefaultView;
```

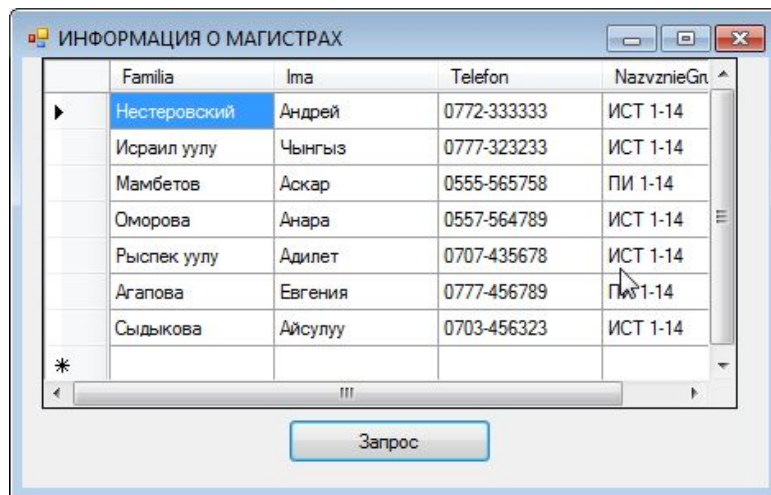


Рис.3. Результат отображения данных в компонент dataGridView

Заключение

Применение новой технологии доступа к данным – ADO.NET дает следующие преимущества:

В ADO .NET используется доступ к отсоединенным данным. До недавних пор применялся доступ к данным через постоянное соединение с источником данных. Приложение открывало соединение с базой данных и не закрывало его по крайней мере до завершения работы с источником данных. В это время соединение с источником поддерживалось постоянно. В ADO .NET соединение устанавливается лишь на то время, которое необходимо для проведения определенной операции над базой данных

ADO.NET является управляемой кодовой библиотекой, и, значит, подчиняется тем же правилам, что и любая управляемая библиотека. Типы, составляющие ADO.NET, используют протокол управления памятью CLR, принадлежат к той же системе типов (классы, интерфейсы, перечисления, структуры и делегаты), и доступ к ним возможен с помощью любого языка .NET. Классы ADO.NET находятся в сборке System.Data.dll /6/.

Одной из ключевых идей, лежащих в основе модели поставщиков ADO.NET, является расширяемость, что позволяет разработчикам создавать собственных поставщиков для патентованных источников данных /7/.

Несмотря на то, что разные поставщики данных .NET используют различные классы, все они некоторым образом стандартизированы. Точнее говоря, каждый поставщик основан на одном и том же наборе интерфейсов и базовых классов. Так, например, объект Connection реализует интерфейс IDbConnection, который определяет такие ключевые методы, как Open() и Close(). Подобная стандартизация гарантирует, что каждый класс Connection будет работать одинаковым образом и предоставит один и тот же набор ключевых свойств и методов. DataSet полностью обобщен; независимо от того, какой поставщик данных применяется, можно извлекать данные и помещать их в полностью автономный DataSet одинаковым образом. Это облегчает отделение кода, извлекающего данные, от кода, обрабатывающего их. В случае смены лежащей в основе базы данных придется изменить только код, извлекающий данные, но если используется DataSet, а информация имеет одну и ту же структуру, модифицировать способ ее обработки не понадобится /8/.

Технология ADO.NET в полной мере способна предоставить механизм для доступа к любому источнику данных, тем самым давая разработчику мощный механизм взаимодействия с базами данных, способный в полной мере реализовать все потребности, возникающие при проектировании ИС.

Список литературы

1. Сеппа Д. Программирование на Microsoft ADO.NET 2.0. [Текст] / Д.Сеппа. - Издательство: Питер, 2007. – 784 с.
2. Дженсен К. Доступ к данным с помощью ADO.NET [Текст]: Пер. с англ./ К.Дженсен. – М.: 2009.
3. Полякова Л.Н. Технологии ADO.NET и ASP.NET [Текст] / Л.Н.Полякова. – Ульяновск, 2006. – 63 с.
4. Шумаков П.В. ADO.NET и создание приложений баз данных в среде MicrosoftVisualStudio.NET. [Текст] / П.В.Шумаков. – М.: Диалог-МИФИ, 2003. – 528 с.
5. Евсеева О.Н. Работа с базами данных на языке С# (технология ADO.NET) [Текст]: Учебное пособие. О.Н.Евсеева, А.Б.Шамшев. – Ульяновск, 2009. – 170 с.
6. Троелсен Э. Язык программирования С# 2010 и платформа .NET 4. [Текст] / Эндрю Троелсен. – М.: Вильямс, 5-е изд. 2010. – 1392 с.
7. <http://www.microsoft.com> – электронный ресурс.
8. <http://club.shelek.ru/viewart.php?id=148> – электронный ресурс.