

## Технология ADO.NET

### ADO.NET TECHNOLOGY

*ADO.NET* технологиясы Microsoft .NETке негизделген тиркемелер үчүн маалыматка кирүүнү камсыз кылган технология. Ал ADOнун мурунку технологиясын өнүктүрүү менен иштетилген эмес, өз алдынча, .NET фреймворкунун бөлүгү болуп саналат. Бул макалада ADO.NETтин иштөө принцибин карап чыгабыз.

**Ачкыч сөздөр:** База, технология, программалоо, тиркемелер, таблицалар, ADO.NET, байланыш, SQL Server, клиент-сервердик тиркемелеринде, туташуу.

ADO.NET — технология, предоставляющая доступ к данным для приложений, основанных на Microsoft .NET. Является не развитием более ранней технологии ADO, а самостоятельной технологией, частью фреймворка .NET. в этой статье рассмотрим принцип работы ADO.NET.

**Ключевые слова:** База данных, технология, программирования, приложения, таблицы, ADO.NET, связь, SQL Server, клиент-серверных приложениях, соединение.

*ADO.NET - technology that provides data access to applications based on Microsoft .NET. It is not the development of an earlier technology ADO, and independent technology, part of the framework .NET. In this article we consider the principle of the ADO.NET*

**Keywords:** Database, technology, programming, applications, tables, ADO.NET, telecommunications, SQL Server, client-server applications, the connection.

ADO .NET (ActiveX Data Objects .NET) считается набором классов, реализующих программные интерфейсы для облегчения подключения к базам данных из приложения самостоятельно с особенностями осуществления определенной системы управления базами данных и от структуры самой базы данных, а еще самостоятельно от места месторасположения этой самой базы в частности, в расчисленной среде (клиент-серверное приложение) на стороне сервера.

ADO .NET широко употребляется вместе с технологией web-программирования с внедрением объектов ASP .NET для доступа к размещенным на сервере базам данных со стороны клиента.

Подробности осуществления конкретной базы данных в рамках данной СУБД никак не обсуждаются. ADO .NET для того и используется, чтобы максимально скрыть подробности реализации конкретной базы и дать программисту комплект обычных классов, интерфейсов, программных средств, которые превращают ход создания приложения в стандартно научно-технический процесс. Таким образом, с точки зрения .NET:

1. столбец (поле, атрибут):
  - характеризуется конкретным типом (данных),
  - очень много значений столбца считаются значениями одного типа;
2. строка (запись, кортеж):
  - характеризуется кортежем атрибутов,

- складывается из упорядоченного большого количества значений (кортежа) атрибутов;

3. таблица:

- набор данных, доставляющих объекты конкретного типа,
- состоит из большого количества частей столбиков-строчек,
- каждая строчка таблицы УНИКАЛЬНА;

4. первичный ключ таблицы:

- содержит множество столбиков таблицы (может быть, состоять из одного столбика), должное значение (комбинации значений) которых в строках таблицы гарантируют неповторимость каждой строчки в данной таблице;

5. дополнительный ключ таблицы:

- а бог его знает, для чего еще одна гарантия особенности строки в таблице;

внешний ключ таблицы:

- непустое множество столбцов таблицы (может быть, заключающееся из одного столбца), надлежащие значения (комбинации значений) каких в строчках таблицы подходят основному либо дополнительному ключу иной таблицы,

- гарантирует закономерную связь между таблицами.

Работа с БД на уровне приложения .NET – это работа:

- с множествами объявлений классов, которые содержат объявления унаследованных способов и параметров, предназначенных для решения задачи извлечения информации из базы данных;

- с обилием объектов-представителей классов, которые гарантируют работу с базами данных;

- с обилием значений и параметров определенных объектов, отображающих специфику структуры определенной базы данных.

Многофункциональные свойства данной сложной системы взаимодействующих классов гарантируют ЕДИНООБРАЗНУЮ службу с базами данных самостоятельно от порядка управления базой и ее осуществлении.

#### **ADO .NET. Доступ к данным**

Планируется, что к моменту написания приложения подходящая база данных уже сделана.

Теория доступа к данным в ADO .NET базируется в использовании двух частей:

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

- **ПРОВАЙДЕРА ДАННЫХ** (передается объектом класса DataProvider). Это посредник, гарантирующий реакцию приложения и базы данных со стороны базы данных (в распределении приложениях со стороны сервера).

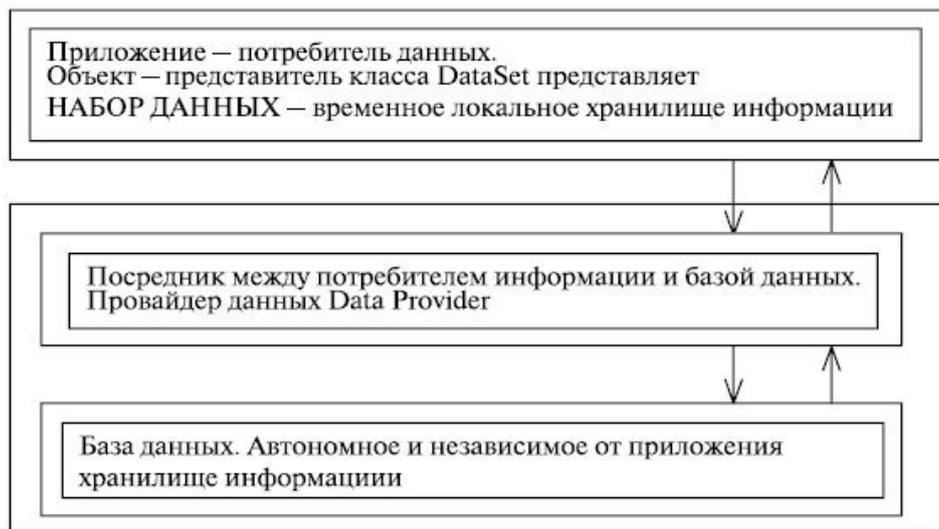


Рис. 1. Теория доступа к данным в ADO .NET

### Практическое применение ADO.NET

Знание использовать на практике теоретические познания увеличивает их цену, потому мы посвящаем примерам работы с ADO.NET самостоятельную область, где рассмотрим более обычные задачи, разбив их на отдельные установки: слияние с источником данных, подборка, удаление и обновление информации, вывод содержимого на экран, работа с XML и т.д. В свойстве платформы про возведение интерфейса будет применена ASP.NET, а как язычок программирования забран C#.

#### Managed Providers.

ADO.NET поддерживает два вида источников данных - SQL Managed Provider и ADO Managed Provider. SQL Managed Provider используется для работы с Microsoft SQL Server 7.0 и выше, ADO Managed Provider - для всех других баз данных.

SQL Managed Provider - работает по особому протоколу, называемому TabularData Stream (TDS) и не использует ни ADO, ни ODBC, ни тот или другой или еще технологию. Направленный специально на MS SQL Server, протокол разрешает увеличить скорость передачи данных и тем самым улучшить общую производительность приложения.

ADO Managed Provider - предназначен для работы с свободной базой данных. Но из-за счет универсальности есть проигрыш по сравнению с SQL Server Provider, так что при работе с SQL Server рекомендуется применять специальные классы. В данном обзоре мы коснемся ADO Managed Provider только мимоходом, показав лишь имеющиеся незначимые отличия, так будто более употребимой базой данных представляется SQL Server 7.0 либо 2000, а отличие содержится, в основном, в именовании.

Для работы с нашими примерами вам понадобится создать базу данных dotSITE и выполнить следующие скрипты для создания структуры и минимального наполнения.

```

if exists (select * from sysobjects
where id = object_id(N'[dbo].[Articles]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Articles]
GO
  
```

```

if exists (select * from sysobjects
where id = object_id(N'[dbo].[Authors]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Authors]
GO
  
```

```

CREATE TABLE [dbo].[Articles] (
[Theme] [char] (50) NOT NULL ,
[Author] [int] NULL
  
```

```
) ON [PRIMARY]
```

```
GO
```

```
CREATE TABLE [dbo].[Authors] (  
  [AuthorID] [int] IDENTITY (1, 1) NOT NULL ,  
  [Name] [char] (50) NULL ,  
  [SirName] [char] (50) NULL
```

```
) ON [PRIMARY]
```

```
GO
```

```
insert into Authors(Name,SirName) values ('Michael','Pliskin')  
insert into Authors(Name,SirName) values ('Andrew','Filev')  
insert into Authors(Name,SirName) values ('Oleg','Stepanov')  
insert into Articles(AuthorID,Theme) values (1,'ASPNET Authentication')  
insert into Articles(AuthorID,Theme) values (2,'ASPNET Overview')  
insert into Articles(AuthorID,Theme) values (3,'Soap')
```

Теперь, когда мы создали базу данных, можно переходить непосредственно к примерам. Во всех случаях, кроме особо оговоренных будет использоваться SQL Managed Provider.

Установление соединения с базой данных: Чтобы установить соединение с нашей базой данных следует выполнить следующий код:

```
SqlConnectionmyConnection=newSqlConnection("server=localhost;uid=sa;pwd=;data  
base  
=dotSITE");  
myConnection.Open();
```

Как видно, при использовании SQL Managed Provider не указывается DSN. Для работы с произвольной базой данных строка соединения указывается как в ADO. В этом случае используются ADO ориентированные классы: ADOConnection, ADOCommand, ADODataReaderи др.

Подведем результат. ADO.NET дает новенькую текстуру объектов и новую образец прохода к данным, позволяющую повысить производительность более часто встречающихся приложений. Это дает нам причины предвещать ADO.NET как основную технологию доступа данным в веб приложениях.

### **Преимущества и нововведения в ADO.NET**

Использование разъединенной модели доступа к данным.

В клиент-серверных приложениях обычно используется технология доступа к источнику данных, около которой соединение с базой поддерживается непрерывно. Однако после широкого распространения приложений, нацеленных на Интернет, обнаружили некие недочеты такого расклада. Попытаемся обнаружить некие из них.

Соединения с базой данных требуют выделения системных ресурсов, что может существовать критично при большой нагрузке сервера. Хотя постоянное соединение позволяет едва ускорить работу приложения, общий убыток от растраты системных ресурсов сводит достоинство на нет.

Особенность интернет приложений не позволяет серверу в любой момент времени знать, что необходимо пользователю. То есть до следующего запроса сервер не имеет представления, необходимо ли еще поддерживать соединение.

Опыт разработчиков продемонстрировал, что приложения с постоянным соединением с источником данных очень тяжелой поддаются масштабированию.

Надлежит принять, что новая технология время от времени все ведь проигрывает традиционной. Для этих ситуации рекомендуется (не только мною, но и Microsoft) использовать ADO. Образцами таковых приложений служат программы, проводящие частые и объемные изменения содержания записей - заказ билетов.

### Список литературы

1. *Сахил Малик*. Microsoft ADO.NET 2.0 для профессионалов = Pro ADO.NET 2.0. — М.: «Вильямс», 2006. — С. 560 стр. — ISBN 1-59059-512-2.
2. *Эндрю Троелсен*. Язык Программирования C#2010 и платформа .NET 4.0 = PRO C# 2010 AND THE .NET 4 PLATFORM. — 5-е издание. — Москва • Санкт-Петербург • Киев:«Вильямс», 2011. — С. 1392 стр. — ISBN 978-5-8459-1682-2.

Ссылки:

1. <https://ru.wikipedia.org/wiki/ADO.NET>
2. <http://www.cyberguru.ru/microsoft-net/ado-net/adonet-overview-page2.html>
3. [http://www.codingclub.net/Articles/ADO\\_NET/Tehnologiyu\\_ADONET\\_preimuschestva\\_i\\_primeri\\_ispolzovanie](http://www.codingclub.net/Articles/ADO_NET/Tehnologiyu_ADONET_preimuschestva_i_primeri_ispolzovanie)