

## ТЕХНОЛОГИЯ ADO.NET

К.С.АБДЫРАИМОВА

*E.mail.* [ksucta@elcat.kg](mailto:ksucta@elcat.kg)

*Макалада ADO.NET технологиясы жана .NET платформасы каралат. ADO.NET маалыматтарына жетүү технологиясын колдонуу актуалдуулугу колдонулуучу МББСнын көптүктүүлүгү жана алар үчүн маалыматтарга жетүүнү уюштуруу зарылчылыгы өкүмдөлөт.*

*В статье рассматриваются технологии ADO.NET и платформа .NET. Актуальность применения технологии доступа к данным ADO.NET диктуется множественностью применяемых СУБД и необходимостью организации доступа к данным для них.*

*In given article technologies ADO .NET and a platform .NET are considered. The urgency of application of technology of access to data ADO .NET is dictated by plurality applied CSDB and necessity of the organization of access to data for them.*

Век высоких технологий и научных достижений диктует высокий темп развития всех отраслей и наук, что привело к автоматизации большинства сфер жизни современного человека. С точки зрения любого руководителя автоматизация уменьшает издержки на производстве и повышает эффективность работы всей организации в целом. Но с точки зрения людей, которые разрабатывают системы автоматизации, ситуация достаточно сложна, так как в основе любой информационной системы лежит база данных, а точнее СУБД. И выбор той или иной СУБД будет влиять на дальнейшие функциональные возможности информационной системы, как и то, как она будет спроектирована. На сегодняшний день насчитывается большое количество различных СУБД, и перед разработчиком стоит сложный выбор, какую СУБД использовать. А знать особенности проектирования при каждой СУБД – непосильная задача даже для целого отдела разработчиков.

Ситуация еще усугубляется, когда необходимо обеспечить поддержку различных источников данных. Причем каждый из таких источников данных может хранить и обрабатывать данные по-своему. Еще необходимо учитывать, что в различных языках программирования различна поддержка работы с той или иной СУБД. То есть еще возникает проблема несоответствия обработки информации большинством СУБД и способом обработки информации различными языками программирования. Решение выдвинутых проблем найдено в новой технологии ADO.NET, разработанной компанией Microsoft и включенной в их новую платформу .NET Framework.

ADO.NET – это часть Microsoft .NET Framework, т.е. набор средств и слоев, позволяющих приложению легко управлять и взаимодействовать со своим файловым или серверным хранилищем данных /1/.

ADO.NET – это основная технология доступа к данным для платформы .NET. Более конкретно – это набор объектов, при помощи которых программист может осуществить подключение к серверу баз данных, выборку данных и их модификацию /2/.

ADO.NET – новый этап в технологии ActiveX Data Objects. 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.

В объектной модели ADO.NET можно выделить несколько уровней.

**Уровень данных.** Это, по сути дела, базовый уровень, на котором располагаются сами данные. На данном уровне обеспечивается физическое хранение информации на магнитных носителях и манипуляция данными на уровне исходных таблиц (выборка, сортировка, добавление, удаление, обновление) /3/.

**Уровень бизнес-логики.** Это набор объектов, определяющих, с какой базой данных предстоит установить связь и какие действия необходимо будет выполнить с содержащейся в ней информацией. Для установления связи с базами данных используется объект `SqlConnection`. Для хранения команд, выполняющих какие-либо действия над данными, используется объект `DataAdapter`. И, наконец, если выполнялся процесс выборки информации из базы данных, для хранения результатов выборки используется объект `DataSet` /4/.

**Уровень приложения.** Это набор объектов, позволяющих хранить и отображать данные на компьютере конечного пользователя. Для хранения информации используется объект `DataSet`, а для отображения данных имеется довольно большой набор элементов управления (`DataGrid`, `TextBox`, `ComboBox`, `Label` и т.п.). В Visual Studio .Net можно вести разработку двух типов приложений. В первую очередь, это традиционные Windows-приложения (на основе Windows-форм), которые реализованы в виде exe-файлов, запускаемых на компьютере пользователя, и Web-приложения (на основе Web-форм), которые работают в оболочке браузера. Для хранения данных на уровне обоих типов приложений используется объект `DataSet`. Обмен данными между приложениями и уровнем бизнес-логики происходит с использованием формата XML, а средой передачи данных служат либо локальная сеть (Инtranет), либо глобальная сеть (Интернет) /5/.

В NET Framework библиотеки ADO.NET находятся в пространстве имени `System.Data`. Эти библиотеки обеспечивают подключение к источникам данных, выполнение команд, а также хранилище, обработку и выборку данных (рис. 1).



Рис. 1. Подключение к источникам данных

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 ряд объектов:

- `SqlConnection` – объект, позволяющий создать соединение с базами данных MS SQL Server;
- `OleDbConnection` – объект, позволяющий создать соединение с любым источником данных (простые текстовые файлы, электронные таблицы, базы данных) через OLE DB;
- `OdbcConnection` – объект, позволяющий создать соединение с ODBC-источниками данных.

Жизненный цикл объекта `Connection` состоит из таких этапов, как объявление объекта соединения; создание объекта соединения; определение строки соединения; использование соединения, например, для создания команды; открытие соединения; выполнение команды; закрытие соединения; обработка полученных данных; изменение

команды; повторное открытие соединения; выполнение команды; закрытие соединения /6/.

Концептуально технология ADO.NET может быть разделена на две различные составляющие: механизм доступа к данным и система хранения данных.

Все основные классы, интерфейсы и типы, используемые в ADO.NET, определены в пространстве имен второго уровня System.Data. Классы, соответствующие системе хранения данных, являются самостоятельными классами, которые можно использовать в любом приложении ADO.NET. Такие классы включают в себя DataColumn, DataRelation, DataRow, DataSet и DataTable, из которых наиболее важным является класс DataSet.

В отличие от механизма хранения, который определен в терминах классов, механизм доступа к данным определен на языке интерфейсов. Эти интерфейсы реализованы специальными классами, связанными с конкретным механизмом доступа к данным. В среде .Net Framework существует пять механизмов доступа к данным. Они связаны со следующими пространствами имен третьего уровня: System.Data.SqlClient, System.Data.OleDb, System.Data.Odbc, System.Data.SqlServerCE и System.Data.OracleClient /7/.

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

### **Список литературы**

1. Вилдермьус, Шон. Практическое использование ADO.NET. Доступ к данным в Internet /Пер. с англ. – М.: Издательский дом «Вильямс», 2003.
2. Полякова Л.Н. Технологии ADO.NET и ASP.NET. – Ульяновск, 2006.
3. Дэвид С. Плат. Знакомство с Microsoft NET. Русская редакция Microsoft, 2001.
4. Дженсен К. Доступ к данным с помощью ADO.NET /Пер. с англ. – М., 2009.
5. Герберт Шилдт . Полный справочник по C#. Издательский дом "Вильямс", 2004. – 752 с.
6. Эндрю Троелсен. Язык программирования C# 2010 и платформа .NET 4. – 5-е изд. – М.: Вильямс, 2010. – 1392 с.
7. Герберт Шилдт C# 4.0: Полное руководство. – М.: Вильямс», 2010. – 1056 с.