Unit-6

Databases Access using ADO.NET

Introduction

- Data access is making the application interact with a database, where all the data is stored.
- Different applications have different requirements for database access.
- ADO.NET is Design to provide consistent access to data sources through ADO.NET data providers.

ADO .NET(Active X Data Object)

- ADO.NET is a data access technology from Microsoft .Net Framework , which provides communication between relational and non-relational systems through a common set of components .
- ADO.NET was built for a disconnected architecture , so it enables truly disconnected data access and data manipulation through its Dataset Object, which is completely independent from the Data Source.

- ADO.NET provides
  - Methods for connecting to data sources
  - Retrieving
  - Manipulating
  - Updating.

Goal of ADO.NET

- Provide a disconnected data architecture in addition to supporting connected operation.
- Integrate tightly with XML
- Interact with a variety of data sources through common data representation
- Optimize data source access

Why ADO.NET?

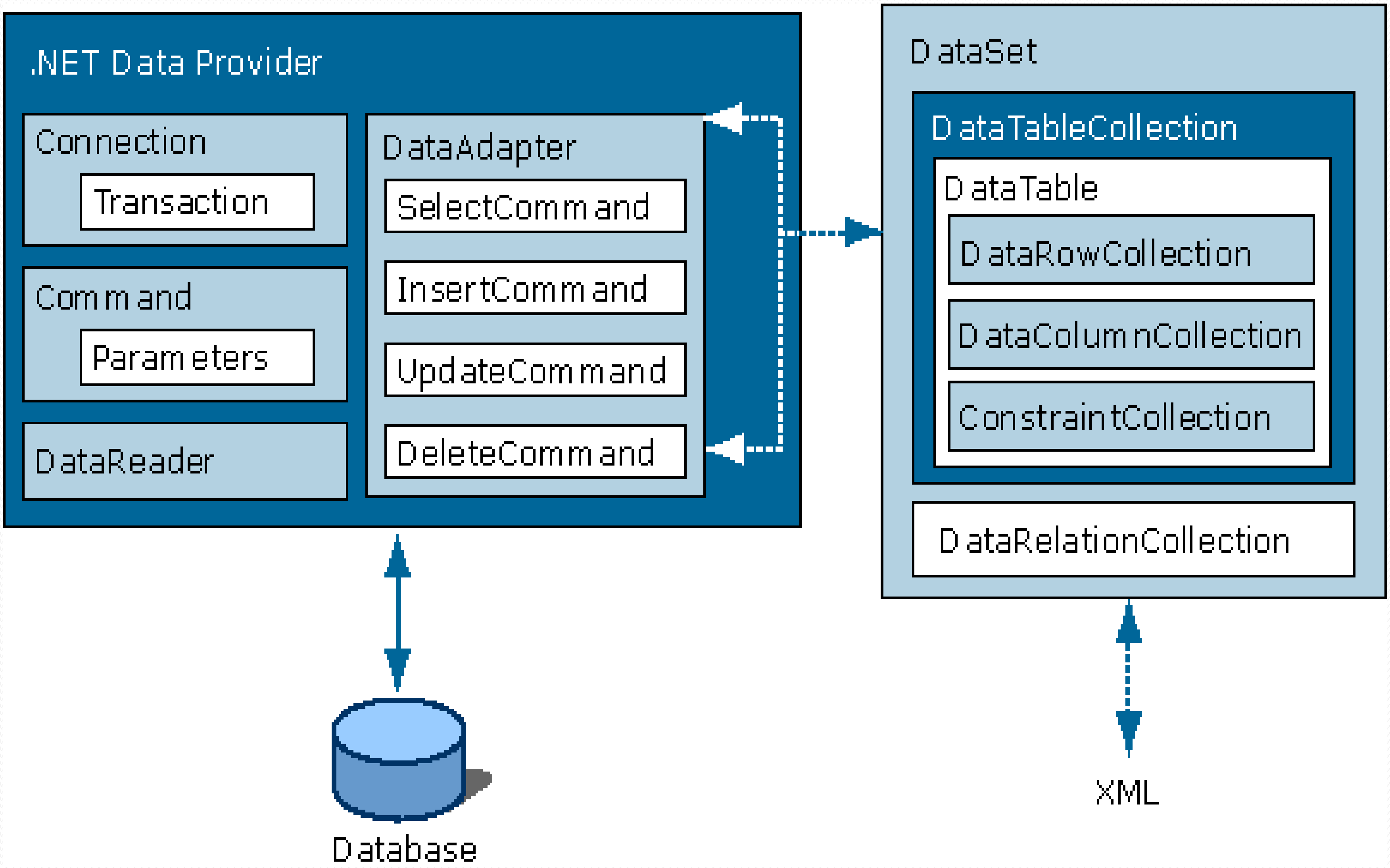
- It Provide a disconnected data architecture in addition to supporting connected operation.
- Disconnected database access model which means, when an application interacts with the database, the connection is opened to serve the request of the application and is closed as soon as the request is completed.
- By keeping connections open for only a minimum period of time, ADO .NET conserves system resources and provides maximum security for databases.

Architecture of ADO.NET

- Two components of ADO.NET :
  - Data Provider
    - The Data Provider is responsible for providing and maintaining the connection to the database.
  - Dataset
    - The dataset is a disconnected, in-memory representation of data.

Connected

Disconnected



.NET Data Provider

Connection

Transaction

Command

Parameters

DataReader

DataAdapter

SelectCommand

InsertCommand

UpdateCommand

DeleteCommand

Database

DataSet

DataTableCollection

DataTable

DataRowCollection

DataColumnCollection

ConstraintCollection

DataRelationCollection

XML

ADO.NET Data Providers

- Data Provider is a set of ADO.NET classes that allows you to access specific database ,execute SQL command and retrieve the data.
- A data provider is a bridge between your application and a data source.

Four Data Providers for ADO.NET.

- OLEDB Provider:(Object Linking and Embedding Database)
  - provides access to any data source that has oledb driver.(like Access)
- ODBC Provider: (Open Database Connectivity)
  - provides access to any data source that has an odbc driver.
- SQLServer Provider: provides access to SQL server database
- Oracle Provider: provides access to Oracle database.

- ADO.NET supports multiple data provider types by the following data provider namespaces:
  - SqlClient
    - members provide high performance connectivity to SQL Server.
  - OracleClient
    - Members deliver functionality similar to SqlClient for Oracle databases.

- OleDb
  - Members provide a direct connection to COM(Component Object Model)-based OLE DB data providers for databases and data sources other than SQL Server and Oracle.
- Odbc
  - Members provide connectivity to legacy data sources that don’t have OLE DB data providers.

Objects of Dataprovider

- The four Objects from the .Net Framework provide the functionality of Data Providers in ADO.NET.
  - Connection Object
    - The Connection Object provides physical connection to the Data Source.
  - Command Object
    - The Command Object uses to perform SQL statement or stored procedure to be executed at the Data Source.

Objects of Dataprovider

- DataReader Object
  - The DataReader Object is a stream-based , forward-only, read-only retrieval of query results from the Data Source, which do not update the data.
- DataAdapter Object.
  - Finally the DataAdapter Object , which populate a Dataset Object with results from a Data Source

ADO.NET provides two types of object

- Connection based
    - Object such as Connection, Command , DataAdapter and Data Reader are connection based.
- Content based
    - Object such as Dataset, DataColumn, DataRow, DataRelation are content based because they are completely independent of the type of data source.

ADO.NET Namespaces

- **System.data**

  Core namespace, defines types that represent data

- **System.Data.Common**

  Types shared between managed providers

- **System.Data.OleDb**

  Types that allow connection to OLE DB compliant data sources

ADO.NET Namespaces

- **System.Data.SqlClient**

  Types that are optimized to connect to Microsoft® SQL Server

- **System.Data.SqlTypes**

  Native data types in Microsoft® SQL Server

- **For SQL Example:**

```
Imports System.Data
Imports System.Data.SQLClient
```

Connection

- Connection class allows you to establish a connection to data source.
- The connection string is a series of name/value setting separate by semicolon(;)
- The few information required in connection string are
  - Server where database is located
  - Database you want to use
  - Database should authenticate you

Connection

- Connections can be opened in two ways:
  - Explicitly by calling the Open method on the connection
  - Implicitly when using a DataAdapter.
- When creating a connection object you can pass the connection string as a constructor parameter.
- You can also set the ConnectionString property by hand.

Common properties and methods

Method/Property	Use
Open()	open the connection
Close()	Close the connection()
ConnectionString	The string use to make connection with database

Common properties and methods

Ex: connection with SqlClient

- Without argument constructor

```
Dim conn as new SQLConnection()
Conn.ConnectionString=conn string
```

- With Argument constructor

```
Dim conn as new SQLConnection(conn string)
```

Command

- The command class allows you to execute any type of SQL Command.
- You can also use command class to perfrom user defined task such as create table, alter table etc.

Command

- Command objects provide three methods that are used to execute commands on the database:
  - ExecuteNonQuery: Executes commands that have no return values such as INSERT, UPDATE or DELETE
  - ExecuteScalar: Returns a single value from a database query.
  - ExecuteReader: Returns a result set by way of a DataReader object.

Data Reader

- Provides methods and properties that deliver a forward-only stream of data rows from a data source.
- When a DataReader is used, parts of the ADO.NET model are cut out, providing faster and more efficient data access.
- A DataReader allows you to read the data returned by a SELECT command one record at a time in a read only-Forward only stream

Data Reader

- Comman properties and methods of datareader are below
  - Read()
  - Close()
- ExecuteReader() is used to read data using datareader.

Data Adapter

- DataAdapter is a part of the ADO.NET Data Provider. Dataset represents a collection of data retrieved from the Data Source and saving data to the Data Source.
- We can use Dataset in combination with DataAdapter class.
- The DataAdapter serves as a bridge between a single DataTable in DataSet and the DataSource.
- It contains all the command for querying and updating the datasource.

Data Adapter

- These two objects combine to enable both data access and data manipulation capabilities.
- Provides a set of methods and properties to retrieve and save data between a DataSet and its source data store.
- Methods of DataAdapter
  - Fill()
  - FillSchema()
  - Update()

DataSet

- The DataSet is the heart of disconnected data access.
- The ADO.NET DataSet contains DataTableCollection and their DataRelationCollection . It represents a collection of data retrieved from the Data Source.
- We can use Dataset in combination with DataAdapter class.
- The DataSet object offers a disconnected data source architecture.

DataSet

- A DataSet Represents a cache of data that contains tables, columns, relationships, and constraints, just like a database.
- Regardless of where the source data comes from, data can all be placed into DataSet objects.
- Tracks changes that are made to the data it holds before updating the source data.
- DataSet are also fully XML-featured.