



AMIT ACADEMY

for Computer Education

SUBJECT – Desktop Application Development-II

Faculty Name – Ankit Rami

Unit-3

ADO.NET Architecture

AMIT ACADEMY FOR COMPUTER EDUCATION

Nr Vardayini Mataji Temple, Rupal, Gandhinagar-382630

Email – amitacademy1117@gmail.com

Mobile No – 8460467193

YouTube Link –

<https://www.youtube.com/@ankitramijoinar>

Instagram Link –

<https://www.instagram.com/amitacademy17/>

Facebook Link –

<https://www.facebook.com/aramitacademy/>

Difference between ADO and ADO.NET

ADO(ActiveX Data Objects) A programming interface from Microsoft that is designed as "the" Microsoft standard for data access.

1. ADO:

- ✓ ADO was introduced in 1996 by Microsoft as its component of MDAC (Microsoft Data Access Components). It is based on COM (Component Object Modelling). ADO with other components of MDAC serves as a framework for client applications to access data stores. It removes necessitate to know implementation of database and lowers complexity of handling low level code requires for dealing with data.

2. ADO.NET:

- ✓ ADO.NET is an advanced database technology from Microsoft .NET Framework which provides communication between application system and database server. It is a component of the .NET Framework that is designed to work on disconnected model to access data from data store. Some of the .NET applications which are used to connect with database server are ASP.NET web applications, windows applications and console applications



S.No.	ADO	ADO.NET
1.	It is based on COM (Component Object Modelling).	It is a CLR (Common Language Runtime) based library.
2.	It works only when data store is connected.	It does not needs active connection to access data from data store.
3.	It has feature of locking.	It does not have feature of locking.
4.	It access and store data from data source by recordset object.	It access and store data from data source by dataset object.
5.	XML integration is not feasible in ADO.	XML integration is feasible in ADO.NET.
6.	In ADO, data is stored in binary form.	While in this, data is stored in XML.
7.	It allow us to create client side cursors only.	It give us the choice of using weather client side and server side cursors.
8.	It requires SQL JOINS and UNIONs to combine data from multiple tables in a single result table.	It uses DataRelational objects, for combining data from multiple tables without requiring JOINS and UNIONs.
9.	It supports sequential access of rows in a RecordSet.	It allows completely non-sequential data access in DataSet through collection based hierarchy.



Features of ADO.Net

- ✓ It provides some in built classes to connect databases like mysql,sql,oracle etc,
- ✓ In built classes to do operations like insert,update,delete and select.
- ✓ Integration is very tight with XML
- ✓ Disconnected Data architecture for better performance
- ✓ Persistence Ignorance
- ✓ Lazy Loading
- ✓ Self Tracking Entities
- ✓ Model-First Development
- ✓ Model-Defined Functions

1. Bulk Copy Operatio

Bulk copying of data from a data source to another data source is a new feature added to ADO.NET 2.0. Bulk copy classes provides the fastest way to transfer set of data from once source to the other. Each ADO.NET data provider provides bulk copy classes. For example, in SQL .NET data provider, the bulk copy operation is handled by SqlBulkCopy class, which can read a Dataset, Data Table, Data Reader, or XML objects. Read more about Bulk Copy [here](#).

2. Batch Update

Batch update can provide a huge improvement in the performance by making just one round trip to the server for multiple batch updates, instead of several trips if the database server supports the batch update feature. The UpdateBatchSize property provides the number of rows to be updated in a batch. This value can be set up to the limit of decimal.

3. Data Paging

Now command object has a new execute method called `ExecutePageReader`. This method takes three parameters - `CommandBehavior`, `startIndex`, and `pageSize`. So if you want to get rows from 101 - 200, you can simply call this method with start index as 101 and page size as 100.

4. Connection Details

Now you can get more details about a connection by setting `Connection's StatisticsEnabled` property to `True`. The `Connection` object provides two new methods - `RetrieveStatistics` and `ResetStatistics`. The `RetrieveStatistics` method returns a `HashTable` object filled with the information about the connection such as data transferred, user details, curser details, buffer information and transactions.

5. DataSet.RemotingFormat Property

When `DataSet.RemotingFormat` is set to `binary`, the `DataSet` is serialized in binary format instead of XML tagged format, which improves the performance of serialization and deserialization operations significantly.

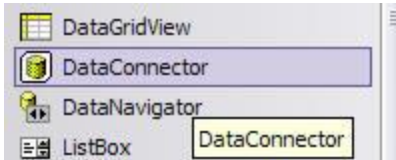
6. DataTable's Load and Save Methods

In previous version of `ADO.NET`, only `DataSet` had `Load` and `Save` methods. The `Load` method can load data from objects such as XML into a `DataSet` object and `Save` method saves the data to a persistent media. Now `DataTable` also supports these two methods.

You can also load a `DataReader` object into a `DataTable` by using the `Load` method.

7. New Data Controls

In Toolbox, you will see these new controls - DataGridView, DataConnector, and DataNavigator. See Figure 1. Now using these controls, you can provide navigation (paging) support to the data in data bound controls.



8. DbProvidersFactories Class

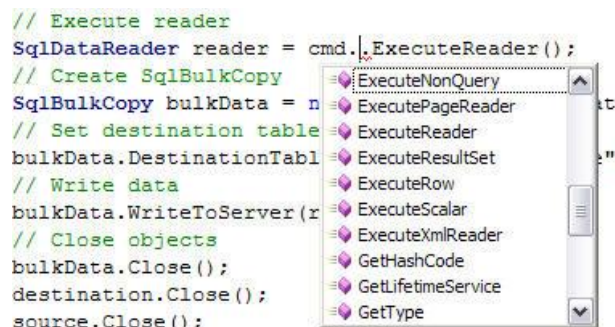
This class provides a list of available data providers on a machine. You can use this class and its members to find out the best suited data provider for your database when writing a database independent applications.

9. Customized Data Provider

By providing the factory classes now ADO.NET extends its support to custom data provider. Now you don't have to write a data provider dependent code. You use the base classes of data provider and let the connection string does the trick for you.

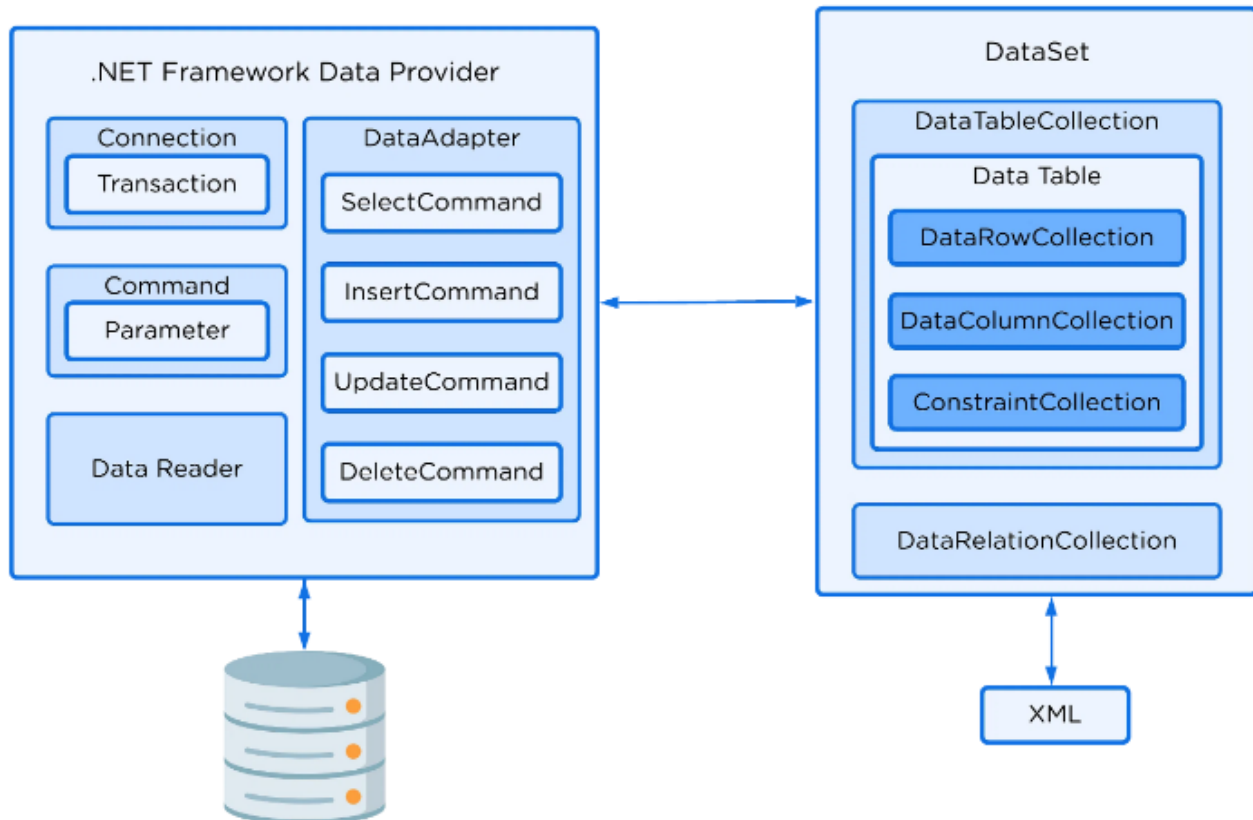
10. DataReader's New Execute Methods

Now command object supports more execute methods. Besides old ExecuteNonQuery, ExecuteReader, ExecuteScalar, and ExecuteXmlReader, the new execute methods are ExecutePageReader, ExecuteResultSet, and ExecuteRow. Figure 2 shows all of the execute methods supported by the command object in ADO.NET 2.0.



ADO.NET Architecture

- ✓ ADO.NET offers a link between the front-end interfaces and the back-end databases. All data access activities are encapsulated as ADO.NET objects, which are then interacted with by controls to show data while concealing the specifics of data transfer.



Connection Object

- ✓ ADO.NET's Connection object should be our initial port of call. It is through connections that ADO.NET is connected to data sources. Between a Data Adapter and a Data Source, there is a Connection object. When a Command object is attached, it may perform SQL commands that can be used to get data from a data source, update, or remove data.

- ✓ An important part of a transaction is the creation of the connection. The good functionality for working with transactions, like commit and rollback, is kept in transaction objects.

Command Object

- ✓ Executing SQL statements and data structures is possible with the Command object. A DataSet or a Data Reader object returns data from SQL statements. SELECT, INSERT, UPDATE, and DELETE Database queries are used to obtain, add, and remove data. These statements may be found in a DataAdapter created using VS .NET IDE.

Data Adapter Object

- ✓ The DataAdapter's function is implicit in its title: it conducts the operations required to transfer data from the server's data source to the database maintained by the DataSet. Data Adapter: We may define instructions for retrieving and updating data using the Data Adapter. To connect a data source to the Dataset, use the Data Adapter. The Data Adapter is aware of the DataSet and the appropriate methods for populating it. It is also known to the Adapter that a link to the data source has been established.

Data Reader Object

- ✓ One of the two techniques offered by ADO.NET, the DataReader object, is used to read information from the data store. As we will recall, the DataReader object offers high efficiency, a read-only, forward-only mechanism for retrieving data as a stream of data from a data repository while maintaining a connection to the data source. The Data Reader is constrained yet extremely efficient.

DataSet Object

- ✓ The ADO.NET DataReader object is one of the two techniques provided by ADO.NET. This object is utilized to obtain data from the data store. As we will recall, the DataReader object offers a read-only, forward-only, high-performance mechanism for retrieving information from the data store as a data stream while maintaining a connection with the data source. This framework is read-only because it can only read data written after it has been read. The Data Reader has several limitations, but it's very well-tuned.

ADO.NET SqlConnection Class

- ✓ It is used to establish an open connection to the SQL Server database. It is a sealed class so that cannot be inherited. SqlConnection class uses SqlDataAdapter and SqlCommand classes together to increase performance when connecting to a Microsoft SQL Server database.
- ✓ Connection does not close explicitly even it goes out of scope. Therefore, you must explicitly close the connection by calling Close() method.
- ✓ **SqlConnection Constructors**

Constructors	Description
SqlConnection()	It is used to initializes a new instance of the SqlConnection class.
SqlConnection(String)0	It is used to initialize a new instance of the SqlConnection class and takes connection string as an argument.
SqlConnection(String, SqlCredential)	It is used to initialize a new instance of the SqlConnection class that takes two parameters. First is connection string and second is sql credentials.

✓ SqlConnection Methods

Method	Description
BeginTransaction()	It is used to start a database transaction.
ChangeDatabase(String)	It is used to change the current database for an open SqlConnection.
ChangePassword(String, String)	It changes the SQL Server password for the user indicated in the connection string.
Close()	It is used to close the connection to the database.
CreateCommand()	It enlists in the specified transaction as a distributed transaction.
GetSchema()	It returns schema information for the data source of this SqlConnection.
Open()	It is used to open a database connection.
ResetStatistics()	It resets all values if statistics gathering is enabled.

Ex-

```
Imports System.Data
Imports System.Data.Sql
Imports System.Data.SqlClient
Public Class register
    Dim cn As New SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=D:\SMS\SMS\SMS\Database1.mdf;Integrated
Security=True;User Instance=True")
    Private Sub register_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        cn.Open()
        If cn.State = ConnectionState.Open Then
            MsgBox("Show Message Login Successfully")
        Else
            MsgBox("Login Fail", MsgBoxStyle.Critical)
        End If
        cn.Close()
    End Sub
End Class
```

Output-

“Show Message Login Successfully”

ADO.NET SqlCommand Class

- ✓ This class is used to store and execute SQL statement for SQL Server database. It is a sealed class so that cannot be inherited.

✓ SqlCommand Constructors

Constructor	Description
SqlCommand()	It is used to initialize a new instance of the SqlCommand class.
SqlCommand(String)	It is used to initialize a new instance of the SqlCommand class with a string parameter.
SqlCommand(String, SqlConnection)	It is used to initialize a new instance of the SqlCommand class. It takes two parameters, first is query string and second is connection string.
SqlCommand(String, SqlConnection, SqlTransaction)	It is used to initialize a new instance of the SqlCommand class. It takes three parameters query, connection and transaction string respectively.
SqlCommand(String, SqlConnection, SqlTransaction, SqlCommandColumnEncryptionSetting)	It Initializes a new instance of the SqlCommand class with specified command text, connection, transaction, and encryption setting.

✓ SqlCommand Methods

Method	Description
BeginExecuteNonQuery()	It is used to Initiate the asynchronous execution of the SQL statement described by this SqlCommand.
Cancel()	It tries to cancel the execution of a SqlCommand.
Clone()	It creates a new SqlCommand object that is a copy of the current instance.
CreateParameter()	It creates a new instance of a SqlParameter object.
ExecuteReader()	It is used to send the CommandText to the Connection and builds a SqlDataReader.
ExecuteXmlReader()	It is used to send the CommandText to the Connection and builds an XmlReader object.
ExecuteScalar()	It executes the query and returns the first column of the first row in the result set. Additional columns or rows are ignored.
Prepare()	It is used to create a prepared version of the command by using the instance of SQL Server.
ResetCommandTimeout()	It is used to reset the CommandTimeout property to its default value.

Ex-

```
cn.Open()  
Dim cmd As New SqlCommand  
("Insert into susers (name,username,password,email,mobile)  
Values  
('" & Ankit & "',''" & arinfoway & "',''" & 123456 & "',''" &  
arinfoway@gmail.com & "',''" & 8460467193 & "'')", cn)  
cmd.ExecuteNonQuery()  
MsgBox("REGISTER SUCCESSFULL", MsgBoxStyle.Information)  
cn.Close()
```

Output-

“REGISTER SUCCESSFUL”

ADO.NET SqlDataAdapter Class

- ✓ The DataAdapter works as a bridge between a DataSet and a data source to retrieve data. DataAdapter is a class that represents a set of SQL commands and a database connection. It can be used to fill the DataSet and update the data source.

✓ **SqlDataAdapter Constructors**

Constructors	Description
DataAdapter()	It is used to initialize a new instance of a DataAdapter class.
DataAdapter(DataAdapter)	It is used to initializes a new instance of a DataAdapter class from an existing object of the same type.

✓ SqlDataAdapter Methods

Method	Description
CloneInternals()	It is used to create a copy of this instance of DataAdapter.
Dispose(Boolean)	It is used to release the unmanaged resources used by the DataAdapter.
Fill(DataSet)	It is used to add rows in the DataSet to match those in the data source.
FillSchema(DataSet, SchemaType, String, IDataReader)	It is used to add a DataTable to the specified DataSet.
GetFillParameters()	It is used to get the parameters set by the user when executing an SQL SELECT statement.
ResetFillLoadOption()	It is used to reset FillLoadOption to its default state.
ShouldSerializeAcceptChangesDuringFill()	It determines whether the AcceptChangesDuringFill property should be persisted or not.
ShouldSerializeFillLoadOption()	It determines whether the FillLoadOption property should be persisted or not.
ShouldSerializeTableMappings()	It determines whether one or more DataTableMapping objects exist or not.
Update(DataSet)	It is used to call the respective INSERT, UPDATE, or DELETE statements.

ADO.NET SqlDataSet Class

- ✓ It is a collection of data tables that contain the data. It is used to fetch data without interacting with a Data Source that's why, it is also known as disconnected data access method. It is an in-memory data store that can hold more than one table at the same time. We can use DataRelation object to relate these tables. The DataSet can also be used to read and write data as XML document.
- ✓ ADO.NET provides a DataSet class that can be used to create DataSet object. It contains constructors and methods to perform data related operations.

✓ SqlDataSet Constructors

Constructor	Description
DataSet()	It is used to initialize a new instance of the DataSet class.
DataSet(String)	It is used to initialize a new instance of a DataSet class with the given name.
DataSet(SerializationInfo, StreamingContext)	It is used to initialize a new instance of a DataSet class that has the given serialization information and context.
DataSet(SerializationInfo, StreamingContext, Boolean)	It is used to initialize a new instance of the DataSet class.

✓ SqlDataSet Properties

Properties	Description
CaseSensitive	It is used to check whether DataTable objects are case-sensitive or not.
DataSetName	It is used to get or set name of the current DataSet.
DefaultViewManager	It is used to get a custom view of the data contained in the DataSet to allow filtering and searching.
HasErrors	It is used to check whether there are errors in any of the DataTable objects within this DataSet.
IsInitialized	It is used to check whether the DataSet is initialized or not.
Locale	It is used to get or set the locale information used to compare strings within the table.
Namespace	It is used to get or set the namespace of the DataSet.
Site	It is used to get or set an ISite for the DataSet.
Tables	It is used to get the collection of tables contained in the DataSet.

✓ SqlDataSet Methods

Method	Description
BeginInit()	It is used to begin the initialization of a DataSet that is used on a form.
Clear()	It is used to clear the DataSet of any data by removing all rows in all tables.
Clone()	It is used to copy the structure of the DataSet.
Copy()	It is used to copy both the structure and data for this DataSet.
CreateDataReader(DataTable[])	It returns a DataAdapter with one result set per DataTable.
CreateDataReader()	It returns a DataAdapter with one result set per DataTable.
EndInit()	It ends the initialization of a DataSet that is used on a form.
GetXml()	It returns the XML representation of the data stored in the DataSet.
GetXmlSchema()	It returns the XML Schema for the XML representation of the data stored in the DataSet.
Load(IDataReader, LoadOption, DataTable[])	It is used to fill a DataSet with values from a data source using the supplied IDataReader.
Merge(DataSet)	It is used to merge a specified DataSet and its schema into the current DataSet.
Merge(DataTable)	It is used to merge a specified DataTable and its schema into the current DataSet.
ReadXml(XmlReader, XmlReadMode)	It is used to read XML schema and data into the DataSet using the specified XmlReader and XmlReadMode.
Reset()	It is used to clear all tables and removes all relations, foreign constraints, and tables from the DataSet.
WriteXml(XmlWriter, XmlWriteMode)	It is used to write the current data and optionally the schema for the DataSet using the specified XmlWriter and XmlWriteMode.

Ex-

cn.Open()

```
Dim da As New SqlDataAdapter("Select * from results", cn)
```

```
Dim ds As New DataSet
```

```
da.Fill(ds, "results")
```

```
DataGridView1.DataSource = ds.Tables("results")
```

cn.Close()

Output-

“Show all Data in Grid View”

	name	roll_no	course
▶	Manzar	101	MCA
	Akhtar	102	MCA
	Mandeep	103	M-Tech
	Gumam	104	B-Tech
*			



ADO.NET SqlDataReader Class

- ✓ This class is used to read data from SQL Server database. It reads data in forward-only stream of rows from a SQL Server database. It is sealed class so that cannot be inherited. It inherits DbDataReader class and implements IDisposable interface.

✓ SqlDataReader Properties

Property	Description
Connection	It is used to get the SqlConnection associated with the SqlDataReader.
Depth	It is used to get a value that indicates the depth of nesting for the current row.
FieldCount	It is used to get the number of columns in the current row.
HasRows	It is used to get a value that indicates whether the SqlDataReader contains one or more rows.
IsClosed	It is used to retrieve a boolean value that indicates whether the specified SqlDataReader instance has been closed.
Item[String]	It is used to get the value of the specified column in its native format given the column name.
Item[Int32]	It is used to get the value of the specified column in its native format given the column ordinal.
RecordsAffected	It is used to get the number of rows changed, inserted or deleted by execution of the Transact-SQL statement.
VisibleFieldCount	It is used to get the number of fields in the SqlDataReader that are not hidden.

✓ SqlDataReader Methods

Method	Description
Close()	It is used to closes the SqlDataReader object.
GetBoolean(Int32)	It is used to get the value of the specified column as a Boolean.
GetByte(Int32)	It is used to get the value of the specified column as a byte.
GetChar(Int32)	It is used to get the value of the specified column as a single character.
GetDateTime(Int32)	It is used to get the value of the specified column as a DateTime object.
GetDecimal(Int32)	It is used to get the value of the specified column as a Decimal object.
GetDouble(Int32)	It is used to get the value of the specified column as a double-precision floating point number.
GetFloat(Int32)	It is used to get the value of the specified column as a single-precision floating point number.
GetName(Int32)	It is used to get the name of the specified column.
GetSchemaTable()	It is used to get a DataTable that describes the column metadata of the SqlDataReader.
GetValue(Int32)	It is used to get the value of the specified column in its native format.
GetValues(Object[])	It is used to populate an array of objects with the column values of the current row.
NextResult()	It is used to get the next result, when reading the results of SQL statements.
Read()	It is used to read record from the SQL Server database.

Ex-

```

cn.open()
Dim cmd As New SqlCommand
("select * from susers where username='" & TextBox1.Text & "'
and password='" & TextBox2.Text & "'", cn)
Dim dr As SqlDataReader = cmd.ExecuteReader
If dr.Read = True Then
    TextBox1.Clear()
    TextBox2.Clear()
    MsgBox("Login success", MsgBoxStyle.Information)
    MainMenu.Show()
End If
cn.close()
  
```

Output-

“Login Success”

Reference Link

1. <https://www.c-sharpcorner.com/article/top-10-new-features-in-ado-net-2-0/>
2. <https://www.simplilearn.com/tutorials/asp-dot-net-tutorial/ado-dot-net>
3. <https://www.javatpoint.com/ado-net-connection>
4. <https://www.javatpoint.com/ado-net-command>
5. <https://www.javatpoint.com/ado-net-datareader>
6. <https://www.javatpoint.com/ado-net-dataset>
7. <https://www.javatpoint.com/ado-net-dataadapter>

 **Any Query Contact Us**

Faculty Name- Ankit Rami

Email – ankitramiblog@gmail.com

Contact No – +91 8460467193

Website - amit.arinfoway.com



Subscribe Our YouTube Channel

<https://www.youtube.com/channel/UCWbJh2iQ8w-8nrU0Xpjpw7g>