ADO.NET

ARCTECH INFO PRIVATE LIMITED

ADO.NET Overview

Provides consistent access to data sources such as SQL Server, Oracle, MySql, XML, etc.

Use by data-sharing applications to connect to these data sources and retrieve, handle, and update the data that they contain.

ADO.NET includes **data providers** for connecting to a database, executing commands, and retrieving results.

Those results are either processed directly or placed in an ADO.NET DataSet object.

The ADO.NET classes are found in System.Data namespace

ADO.NET provides the most direct method of data access within.NET

.NET Architecture

The two main components of ADO.NET for accessing and manipulating data are

- .NET Framework data providers
  - These components have been explicitly designed for data manipulation and fast, forward-only, read-only access to data
- DataSet
  - is explicitly designed for data access independent of any data source.
  - As a result, it can be used with multiple and differing data sources, used with XML data, or used to manage data local to the application.

.NET data providers

The `Connection` object provides connectivity to a data source.

The `Command` object enables access to database commands to return data, modify data, run stored procedures, and send or retrieve parameter information

The DataReader provides a high-performance stream of data from the data source.

Finally, the DataAdapter provides the bridge between the DataSet object and the data source.

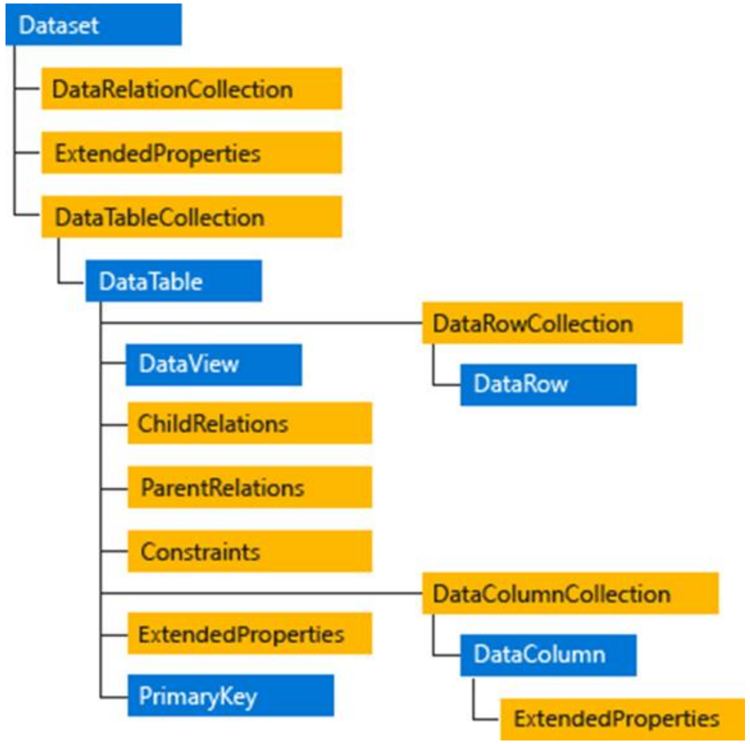
- The DataAdapter uses Command objects to execute SQL commands at the data source to both load the DataSet with data and reconcile changes that were made to the data in the DataSet back to the data source.

DataSet

The **DataSet** contains a collection of one or more **DataTable** objects.

The **DataTable** consists of rows and columns of data, and also primary key, foreign key, constraint, and relation information about the data in the **DataTable** objects.

The diagram on the right illustrates the relationship between a .NET Framework data provider and a **DataSet**.



Choosing DataReader or DataSet

USE DATASET IF YOU NEED THE FOLLOWING

Cache data locally in your application so that you can manipulate it.

Interact with data dynamically such as binding to a Windows Forms control or combining and relating data from multiple sources.

Perform extensive processing on data without requiring an open connection to the data source, which frees the connection to be used by other clients.

The DataAdapter uses the DataReader to fill the contents of a DataSet.

ELSE USE DATAREADER

If you only need to read the results of a query, the DataReader is the better choice.

DataReader returns your data in a forward-only, read-only manner, which improves application performance

By using the DataReader directly, you can boost performance because

- you will save memory that would be consumed by the DataSet
- avoid the processing that is required to create and fill the contents of the DataSet.

Retrieve data using a DataReader

Steps involved in retrieving data

- Open connection to Database Server
- Create a Command with query or StoredProcedure for the current Connection
- Execute the Command to get a reference to the DataReader
- Read all records one by one and access the fields

See Example in Visual Studio

Load DataSet from a DataAdapter

Steps involved in populating DataSet

- Open connection to Database Server
- Create a SqlDataAdapter with query or StoredProcedure for the current Connection
- Create a DataSet object
- Fill the DataSet using the SqlDataAdapter

See Example in Visual Studio