

An Introduction to ADO.Net

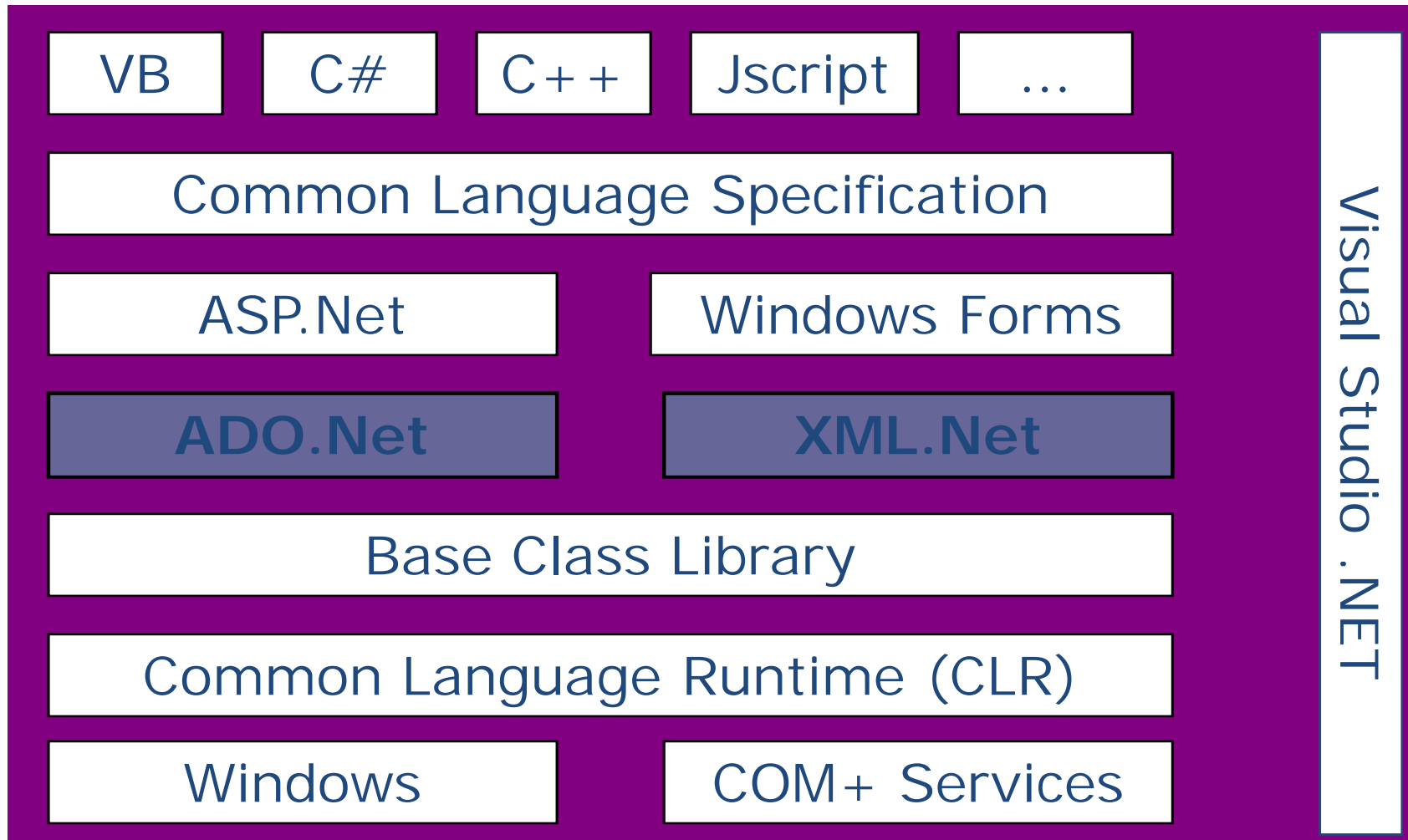
Contents

- What is ADO.Net?
- Why ADO.Net?
- Data Providers in ADO.Net.
- The ADO.Net object structure
- Connecting
- Commanding
- Readers and DataSets

What is ADO.Net?

- ADO.Net is Object-oriented set of libraries included in .Net f/w that helps developers to communicate with various data source from .Net Applications.
- It is one step forward for Ms data access technology, it give more powerful control to developers over how their code interacts with data.

Where does ADO.Net sit?



Why ADO.Net ?

1. Providing tight integration with XML
2. Providing seamless integration with the .NET Framework (e.g., compatibility with the base class library's type system).
3. Providing a comprehensive disconnected data-access model, which is crucial to the Web environment

ADO / ADO.Net Comparisons

Feature	ADO	ADO.Net
In memory data storage	Recordset object Mimics single table	Dataset object Contains DataTables
Data Reads	Sequential	Sequential or non-sequential
Data Sources	OLE/DB via the Connection object	Managed provider calls the SQL APIs

ADO / ADO.Net Comparisons

Feature	ADO	ADO.Net
Disconnected data	Limited support, suitable for R/O	Strong support, with updating
Passing datasets	COM marshalling	DataSet support for XML passing
Connection Mode	Recordset object operated in a fully connected state.	Disconnected Mode,

.Net Data Providers

- What is Data Providers ?
- why required in ADO.NET
- A .NET managed data provider is a collection of classes designed to allow you to communicate with a particular type of data store.

.Net Data Providers



The OleDb data provider is for connecting to any OLE-DB compliant data store (Access, Excel, Microsoft SQL Server, Oracle, etc., essentially any modern data store).

- The SqlClient data provider is specifically for Microsoft's SQL Server, versions 7.0 and up.
- The SqlClient data provider provides much faster access to SQL Server than the OleDb equivalent;

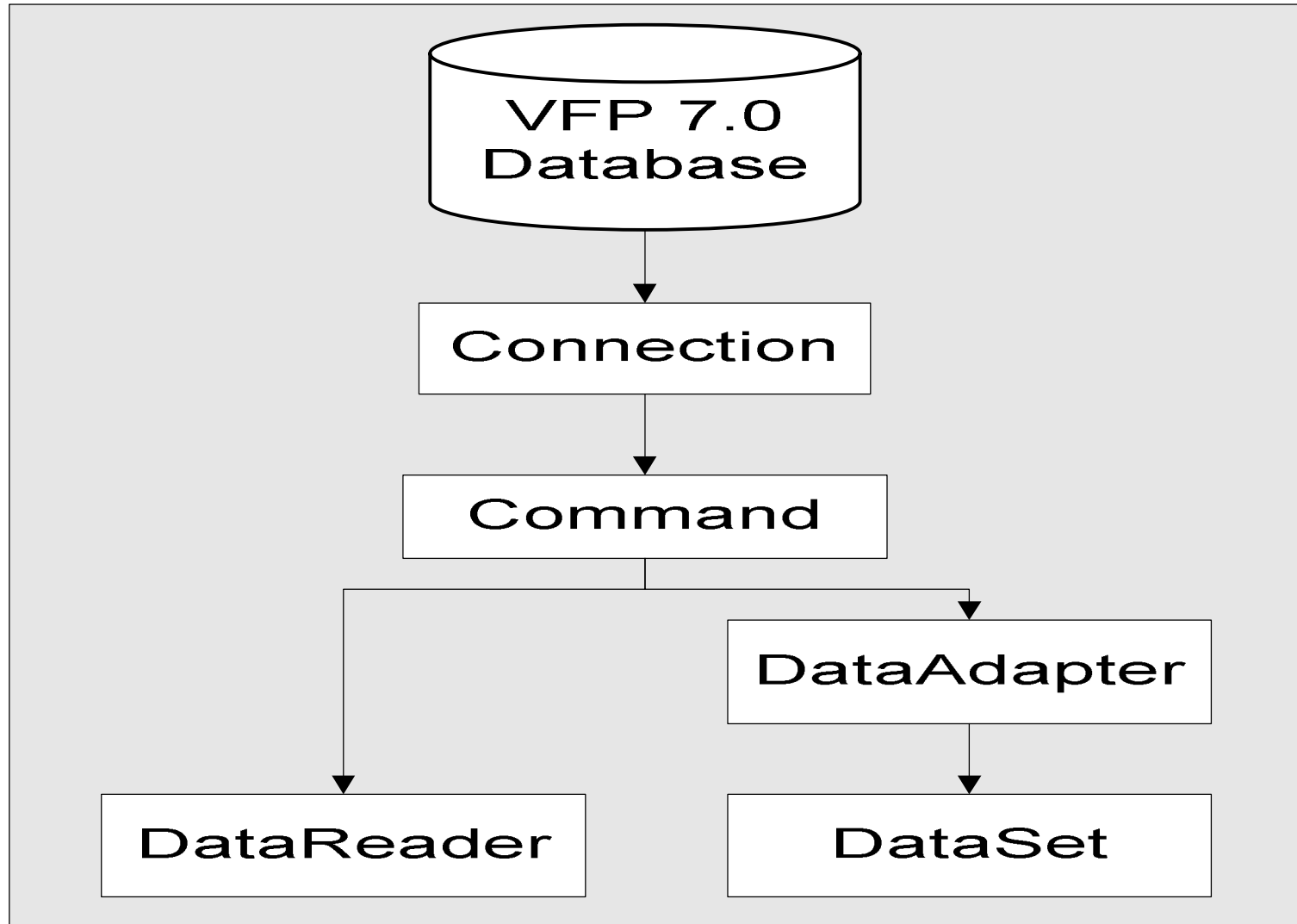
.Net Data Providers

- Each .NET data provider implements the same base classes for Connection, Command, DataReader, Parameter, and Transaction – although their actual names depend on the provider.
- Example : For Connection Object,
- SqlClient Data Providers -> SqlConnection
- OLEDB Data Providers -> OleDbConnection.
- Both will have same basic features

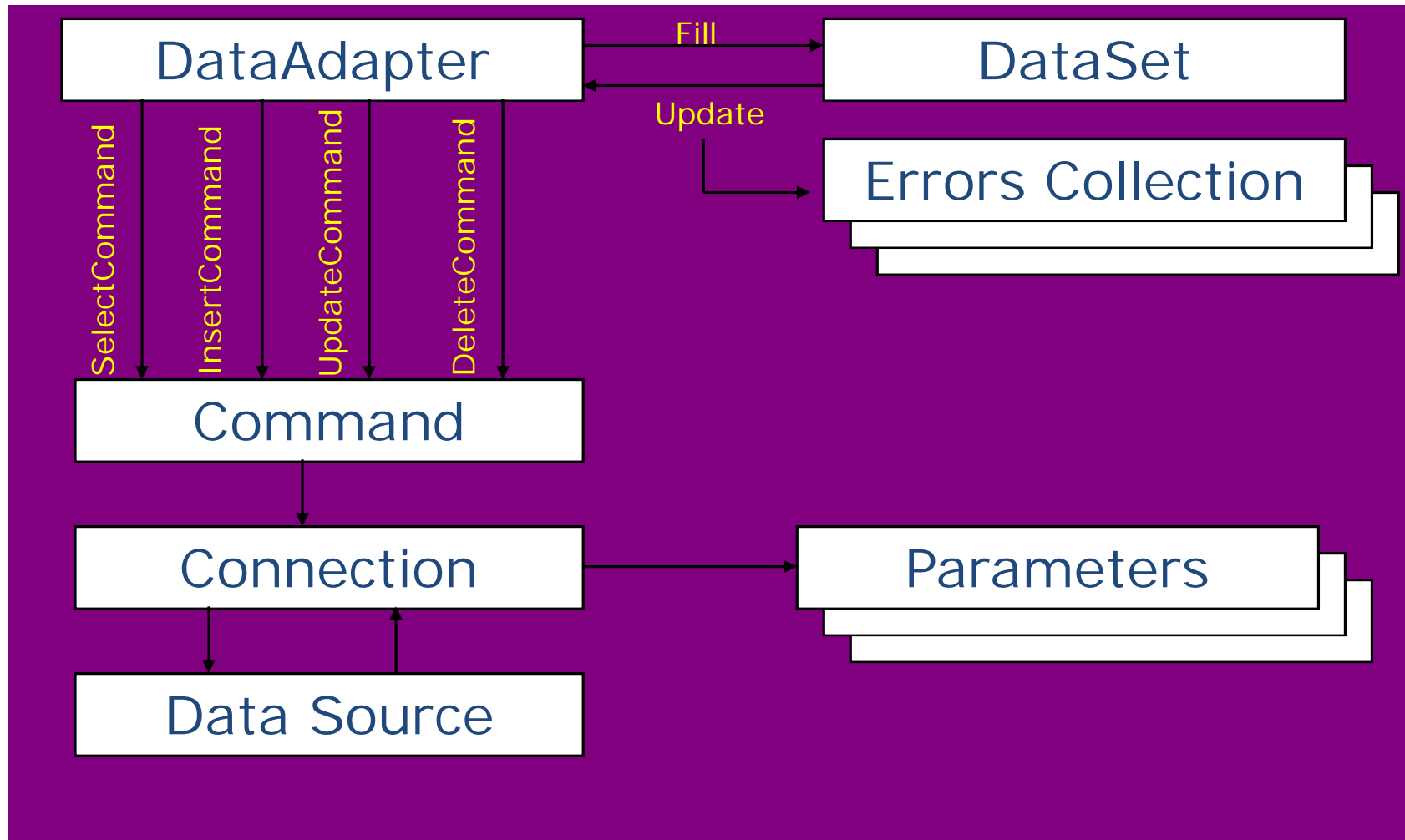
.Net Data Providers

- Each data Providers has its own Namespace
- SQLClient -----> System.Data.SqlClient
- OLEDB----->System.Data.OleDb
- ODBC ----->System.Data.odbc
- These all are subset of System.Data Namespace

ADO.Net Object OverView



ADO.Net object model



Namespaces

- System.Data & System.Data.Common
- System.Data.SqlClient & System.Data.OleDb
- System.Data.SqlTypes
- System.XML & System.XML.Schema

Using Namespaces

- VB.Net
Imports System.Data
Imports System.Data.SqlClient

```
Dim sqlAdp as SqlDataAdapter
```

- C#
using System.Data;
using System.Data.SqlClient;
- SqlDataAdapter sqlAdp= new
SqlDataAdapter();