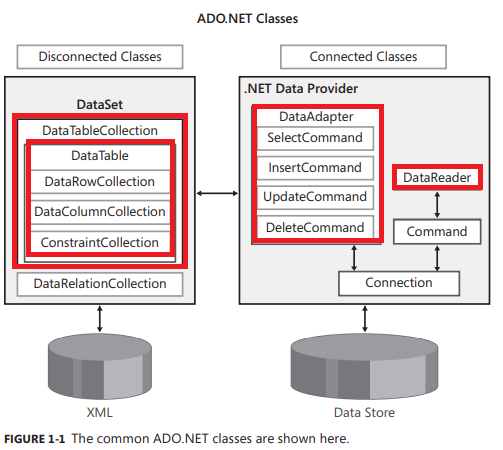
1. .NET desconectado
2. .NET conectado
3. LINQ
4. LINQ a SQL
5. LINQ a XML
6. ADO Entity Framework
7. WCF Data Services



The disconnected data access classes you instantiate in your applications are implemented in the **System.Data.dll** assembly from the .NET Framework. These classes are in the System.Data namespace.

DataTable cars = new DataTable ("Cars");

//Add the DataColumn using all properties

DataColumn vin = new DataColumn("Vin");

vin.DataType = typeof(string);

vin.MaxLength = 23;

vin.Unique = true;

vin.AllowDBNull = false;

vin.Caption = "VIN";

cars.Columns.Add(vin);

//Add the DataColumn using defaults

DataColumn make = new DataColumn("Make");

make.MaxLength = 35;

make.AllowDBNull = false;

cars.Columns.Add(make);

DataColumn year = new DataColumn("Year", typeof(int));

year.AllowDBNull = false;

cars.Columns.Add(year);

//Derived column using expression

DataColumn yearMake = new DataColumn("Year and Make");

yearMake.DataType = typeof(string);

yearMake.MaxLength = 70; yearMake.Expression = "Year + ' ' + Make";

cars.Columns.Add(yearMake);

DataColumns Properties

* DataType
* MaxLength
* Unique
* AllowDBNull
* Caption : default column name

//Set the Primary Key

cars.PrimaryKey = new DataColumn[] {vin};

**Autoincrement**

DataColumn col = dt.Columns.Add(CATEGORYID\_FIELD, typeof(System.Int32));

col.Autoincrement = true;

col.AutoincrementSeed = 1;

col.AutoIncrementStep = -1;

dt.PrimaryKey = new DataColumn[] {col};

A potential problem occurs when new rows are being inserted into an existing table for an identity field (in SQL Server) where the generated values conflict with existing values in the table because of, perhaps, new records added to the data source by other users. In this case, instead of being interpreted as new records by the data source, these records are incorrectly interpreted as updates of existing records.

The problem can be avoided by setting the AutoIncrementSeed value to -1 and the AutoIncrementStep value to -1 thereby generating a sequence of negative values that does not conflict with the values generated by the data source, as long as the data source does not generate negative values.

//Add New DataRow by creating the DataRow first

DataRow newCar = cars.**NewRow**();

newCar ["Vin"] = "123456789ABCD";

newCar ["Make"] = "Ford";

newCar ["Year"] = 2002;

cars.Rows.Add(newCar);

//Add New DataRow by simply adding the values

cars.Rows.**Add**("987654321XYZ", "Buick", 2001);

//Load DataRow, replacing existing contents, if existing

cars.LoadDataRow(new object[]

{ "987654321XYZ", "Jeep", 2002 },LoadOption.OverwriteChanges);