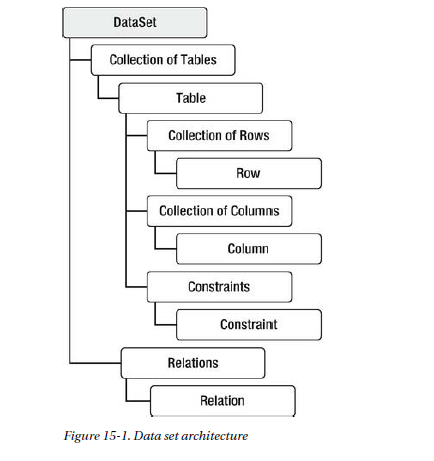
**Using Data Sets and Data Adapters**

Data sets are completely independent of and can be used either connected to or

disconnected from data sources. Their fundamental purpose is to provide a relational view of data

stored in an in-memory cache.



Sample Data Set Code

Members:

1. Connection String
2. Command String
3. Adapter Name
4. Data Set Name
5. View ( Data Grid View )

String Connectstring = “ DataSource = ; Initial Catalog = pub; Integrated Security = true; “

String SqlCommand = “ Select \* from Author”;

SqlConnection connection = new SqlConnection ( Connectionstring);

SqlAdapter Adapter = new SqlAdapter ( SqlCommand, connection );

Dataset ds = new Dataset ();

Connection.Open ();

Adapter.fill ( ds, “Author\_Table”);

Connection.Close();

Datagridview.DataSource = ds;

DatagridView.DataMembers = “Author\_Table”;

**Using DataGridView**

Shows data.

Members:

1. Data Source – Source of your data, either table or list
2. Columns – You can manipulate the visibility by adding .Visible property
3. AutoGenerateColumns - Generates Column

String Location = @”C:\Temp”

String[] FilesSearch = new String[] { “Process.xml” };

//Get All Files

DirectoryInfo folder = new DirectoryInfo (location);

FileInfo[] file = folder.Getfile (“\*”);

//Filter folder

List<FileInfo> ListOfFiles = new List<FileInfo> ();

ListOfFiles.AddRange (file.where( x => FileSearch.Contains (x.Name)));

//Show it into a DataGridView

dataGridView.DataSource = ListOfFiles;

//set All columns into invisible

foreach ( DataGridViewColumn col in ListOfFiles.Column)

{

Col.Visible = false;

}

//Show the desired column

ListOFFiles.Column[“Name”].Visible = True;