**Shanu Nested DataGridView**

In my previous article I have explained about how to create a DatagGridView helper class using c# click this link to view my previous article.

“http://www.codeproject.com/Articles/841755/DataGridView-Helper-Class"

I have extended the DatagGridView Helper class to create a Nested DatagGridView. Few days before one of codeproject member asked a question on how to create a Nested or Master Detail or Hierarchical DataGridView for winforms. I started searching in Google but badly I didn’t get any result for nested DataGridView .I started working on creating a Nested DataGridView sample program which should be useful to all. As a result I have created a Nested DataGridView and you can find the source code from this article. My aim is to create a simple and easy program for users. User can download the code and can customize depend on their requirement.

**Why we need Nested or Hierarchical DataGridView**

In real time projects like Order Management, Production management and etc. we need to display the data in the hierarchical result.

For example let’s take Order Management project for a restaurant. Let’s consider four peoples going to a restaurant to have their lunch. The waiter from the restaurant will give a menu card to select the item to place an order. Now in a Table total 4 peoples are sitting. In a restaurant management usually for all tables there will be a unique Table Id or name. All 4 peoples will select their item need from menu card and place the order to serve their food. In restaurant management for each order we will create a unique Id in an Order Master table and all the item details related to the order in Order Detail table. Let’s see an example structure of the order.

**Why we need Master and Detail Table?** To avoid the duplicate data we can use the master Detail table relation to store our data. For example for every order there will be one waiter and one Table so if we didn’t use the Master Detail table relation the output will be like this below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Order No | Table ID | Item name | Price | Qty | Notes | Waiter | Order Date |
| Ord\_001 | Table\_001 | Burger | 150 | 4 | Served quickly | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Coffee | 60 | 2 | Need More sugar | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Chicken Fry | 450 | 1 | Served quickly | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Ice cream | 75 | 1 | Served quickly | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Fruit Salad | 50 | 1 | Served quickly | Shanu | 2014-11-28 |

Here we can see that Order No, Table ID, Waiter Name and order Date has been repeated. To avoid this duplicate data we will create a Master and Detail relation tables .See the below table for Master and Details.

**Order Master Table**

Here we can see the duplicate data has been stored in separate table as a Order Master Table.

|  |  |  |  |
| --- | --- | --- | --- |
| Order No | Table ID | Waiter | Order Date |
| Ord\_001 | Table\_001 | Shanu | 2014-11-28 |

**Order Detail Table**

Here we can see all the item details of order in separate table. But in the detail table we have used the **Order NO** for a relation to the Master table. Using the relation we can combine both the table and produce the out output.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order Detail No | Order No | Item name | Price | Qty | Notes |
| Ord\_Dtl\_001 | Ord\_001 | Burger | 150 | 4 | Served quickly |
| Ord\_Dtl\_002 | Ord\_001 | Coffee | 60 | 2 | Need More sugar |
| Ord\_Dtl\_003 | Ord\_001 | Chicken Fry | 450 | 1 | Served quickly |
| Ord\_Dtl\_004 | Ord\_001 | Ice cream | 75 | 1 | Served quickly |
| Ord\_Dtl\_005 | Ord\_001 | Fruit Salad | 50 | 1 | Served quickly |

**Normal grid result:**

The result can be shown without using the Hierarchal grid output .But we have to display the duplicate result as below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Order No | Table ID | Item name | Price | Qty | Notes | Waiter | Order Date |
| Ord\_001 | Table\_001 | Burger | 150 | 4 | Served quickly | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Coffee | 60 | 2 | Need More sugar | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Chicken Fry | 450 | 1 | Served quickly | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Ice cream | 75 | 1 | Served quickly | Shanu | 2014-11-28 |
| Ord\_001 | Table\_001 | Fruit Salad | 50 | 1 | Served quickly | Shanu | 2014-11-28 |

We can also merge the same data and show the result like this below table. But the output is not much good and not easy to view and understand.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Order No | Table ID | Item name | Price | Qty | Notes | Waiter | Order Date |
| Ord\_001 | Table\_001 | Burger | 150 | 4 | Served quickly | Shanu | 2014-11-28 |
| Coffee | 60 | 2 | Need More sugar |
| Chicken Fry | 450 | 1 | Served quickly |
| Ice cream | 75 | 1 | Served quickly |
| Fruit Salad | 50 | 1 | Served quickly |

Let’s see now the hierarchicaloutput of the same result.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Order No | Table ID | Waiter | Order Date |
| + | Ord\_001 | Table\_001 | Shanu | 2014-11-28 |
|  | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Order Detail No | Order No | Item name | Price | Qty | Notes | | Ord\_Dtl\_001 | Ord\_001 | Burger | 150 | 4 | Served quickly | | Ord\_Dtl\_002 | Ord\_001 | Coffee | 60 | 2 | Need More sugar | | Ord\_Dtl\_003 | Ord\_001 | Chicken Fry | 450 | 1 | Served quickly | | Ord\_Dtl\_004 | Ord\_001 | Ice cream | 75 | 1 | Served quickly | | Ord\_Dtl\_005 | Ord\_001 | Fruit Salad | 50 | 1 | Served quickly | | | | |

Now this final result looks much better then all. It will be easy to view the master and detail of all records.

Here is my sample output of hierarchical Datagridview.



Same like Order Management in Restaurant projects we will have Bill Master and Detail, Account Master and Detail. Item Master and Detail, Inventory Master and Detail. In production projects we will have Production Order Master and Oder Detail, Finished Good Receipt Master and Detail, Finished Goods Issue Master and Detail and etc. Same like this in our all real time projects we will use the Master and detail relation to display our data.

**Using the Code**

As I told you in this article that I have used and extended my DataGridView helper class to create a Nested DataGridView. You can view my DataGridView helper class details from my article “Link”.

In my DGVhelper class I have added the below functionality to create the nested grid.

* ImageCoulmn
* DGVMasterGridClickEvents
* DGVDetailGridClickEvents

User can use all events like CellClick, CellContentClick and etc for both Master and Detail grid.

I have created two separate list class to populate the master and detail result. In form load called the method to add the details to each list class.

I have created both Master and Detail Datagridview programmatically (Dynamically) Using my ShanuDGVHelper Class.

**Master Grid Setting:** In Form load I have called this method to create a master DataGridView at runtime.

In my code I add the comments before each line to explain its use.

// to generate Master Datagridview with your coding

public void MasterGrid\_Initialize()

{

//First generate the grid Layout Design

Helper.ShanuDGVHelper.Layouts(Master\_shanuDGV, Color.LightSteelBlue, Color.AliceBlue, Color.WhiteSmoke, false, Color.SteelBlue, false, false, false);

//Set Height,width and add panel to your selected control

Helper.ShanuDGVHelper.Generategrid(Master\_shanuDGV, pnlShanuGrid, 1000, 600, 10, 10);

// Image Column creation

Helper.ShanuDGVHelper.Templatecolumn(Master\_shanuDGV, ShanuControlTypes.ImageColumn, "img", "", "", true, 26, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleCenter, DataGridViewContentAlignment.MiddleRight, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Master\_shanuDGV, ShanuControlTypes.BoundColumn, "Order\_No", "Order NO", "Order NO", true, 90, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleLeft, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Master\_shanuDGV, ShanuControlTypes.BoundColumn, "Table\_ID", "Table ID", "Table ID", true, 80, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleLeft, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Master\_shanuDGV, ShanuControlTypes.BoundColumn, "Description", "Description", "Description", true, 320, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleLeft, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Master\_shanuDGV, ShanuControlTypes.BoundColumn, "Order\_DATE", "Order DATE", "Order DATE", true, 140, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleCenter, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Master\_shanuDGV, ShanuControlTypes.BoundColumn, "Waiter\_ID", "Waiter\_ID", "Waiter\_ID", true, 120, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleLeft, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

//Convert the List to DataTable

DataTable detailTableList = ListtoDataTable(DataClass.OrderDetailBindClass.objDetailDGVBind);

// Image Colum Click Event - In this method we create an event for cell click and we will display the Detail grid with result.

objshanudgvHelper.DGVMasterGridClickEvents(Master\_shanuDGV, Detail\_shanuDGV, Master\_shanuDGV.Columns["img"].Index, ShanuEventTypes.cellContentClick, ShanuControlTypes.ImageColumn, detailTableList, "Order\_No");

// Bind data to DGV.

Master\_shanuDGV.DataSource = DataClass.OrderMasterBindClass.objMasterDGVBind;

}

**Cell Click Event:** I have called this above methodto create a Cell click event for master DataGridView.

objshanudgvHelper.DGVMasterGridClickEvents(Master\_shanuDGV, Detail\_shanuDGV, Master\_shanuDGV.Columns["img"].Index, ShanuEventTypes.cellContentClick, ShanuControlTypes.ImageColumn, detailTableList, "Order\_No");

This event will be used for the master grid Image click event. I have passed master DataGridView, detail DataGridView, Img column Index, Detail grid data table and detail table filter column name.

public void DGVMasterGridClickEvents(DataGridView ShanuMasterDGV, DataGridView ShanuDetailDGV, int columnIndexs, ShanuEventTypes eventtype, ShanuControlTypes types,DataTable DetailTable,String FilterColumn)

{

MasterDGVs = ShanuMasterDGV;

DetailDGVs = ShanuDetailDGV;

gridColumnIndex = columnIndexs;

DetailgridDT = DetailTable;

FilterColumnName = FilterColumn;

MasterDGVs.CellContentClick += new DataGridViewCellEventHandler(masterDGVs\_CellContentClick\_Event);

}

private void masterDGVs\_CellContentClick\_Event(object sender, DataGridViewCellEventArgs e)

{

DataGridViewImageColumn cols = (DataGridViewImageColumn)MasterDGVs.Columns[0];

// cols.Image = Image.FromFile(ImageName);

MasterDGVs.Rows[e.RowIndex].Cells[0].Value = Image.FromFile("expand.png");

if (e.ColumnIndex == gridColumnIndex)

{

if (ImageName == "expand.png")

{

DetailDGVs.Visible = true;

ImageName = "toggle.png";

// cols.Image = Image.FromFile(ImageName);

MasterDGVs.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = Image.FromFile(ImageName);

String Filterexpression = MasterDGVs.Rows[e.RowIndex].Cells[FilterColumnName].Value.ToString();

MasterDGVs.Controls.Add(DetailDGVs);

Rectangle dgvRectangle = MasterDGVs.GetCellDisplayRectangle(1, e.RowIndex, true);

DetailDGVs.Size = new Size(MasterDGVs.Width - 200, 200);

DetailDGVs.Location = new Point(dgvRectangle.X, dgvRectangle.Y + 20);

DataView detailView = new DataView(DetailgridDT);

detailView.RowFilter = FilterColumnName + " = '" + Filterexpression + "'";

if (detailView.Count <= 0)

{

MessageBox.Show("No Details Found");

}

DetailDGVs.DataSource = detailView;

}

else

{

ImageName = "expand.png";

// cols.Image = Image.FromFile(ImageName);

MasterDGVs.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = Image.FromFile(ImageName);

DetailDGVs.Visible = false;

}

}

else

{

DetailDGVs.Visible = false;

}

}

In cell click event if the image column is clicked. I will change the image to Expand and Collapse depends on the selected image name.

If the image is selected to Expand then I will make visible of the detail DataGridView.

In cell click event I will get for the current selected Order No. This order No will be used in “DataView” to filter only the selected order result. The Final result will be bind to the Detail DataGridView.

**Detail Grid Setting:** In Form load I have called this method to create a Detail DataGridView at runtime.

In my code I add the comments before each line to explain its use.

// to generate Detail Datagridview with your coding

public void DetailGrid\_Initialize()

{

//First generate the grid Layout Design

Helper.ShanuDGVHelper.Layouts(Detail\_shanuDGV, Color.Peru, Color.Wheat, Color.Tan, false, Color.Sienna, false, false, false);

//Set Height,width and add panel to your selected control

Helper.ShanuDGVHelper.Generategrid(Detail\_shanuDGV, pnlShanuGrid, 800, 200, 10, 10);

// Color Dialog Column creation

Helper.ShanuDGVHelper.Templatecolumn(Detail\_shanuDGV, ShanuControlTypes.BoundColumn, "Order\_Detail\_No", "Detail No", "Order Detail No", true, 90, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleCenter, DataGridViewContentAlignment.MiddleRight, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Detail\_shanuDGV, ShanuControlTypes.BoundColumn, "Order\_No", "Order NO", "Order NO", true, 80, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleLeft, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Detail\_shanuDGV, ShanuControlTypes.BoundColumn, "Item\_Name", "Item\_Name", "Item\_Name", true,160, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleLeft, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Detail\_shanuDGV, ShanuControlTypes.BoundColumn, "Notes", "Notes", "Notes", true, 260, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleLeft, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Detail\_shanuDGV, ShanuControlTypes.BoundColumn, "Price", "Price", "Price", true, 70, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleRight, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

// BoundColumn creation

Helper.ShanuDGVHelper.Templatecolumn(Detail\_shanuDGV, ShanuControlTypes.BoundColumn, "QTY", "QTY", "QTY", true, 40, DataGridViewTriState.True, DataGridViewContentAlignment.MiddleRight, DataGridViewContentAlignment.MiddleCenter, Color.Transparent, null, "", "", Color.Black);

objshanudgvHelper.DGVDetailGridClickEvents(Detail\_shanuDGV);

}

**Cell Click Event:** I have called this above methodto create a Cell click event for Detail DataGridView.

objshanudgvHelper.DGVDetailGridClickEvents(Detail\_shanuDGV);

This event will be used for the Detail grid Cell Click Event. I have passed Detail DataGridView, , Using the messagebox I will display the cell text from cell click event.

public void DGVDetailGridClickEvents(DataGridView ShanuDetailDGV)

{

DetailDGVs = ShanuDetailDGV;

DetailDGVs.CellContentClick += new DataGridViewCellEventHandler(detailDGVs\_CellContentClick\_Event);

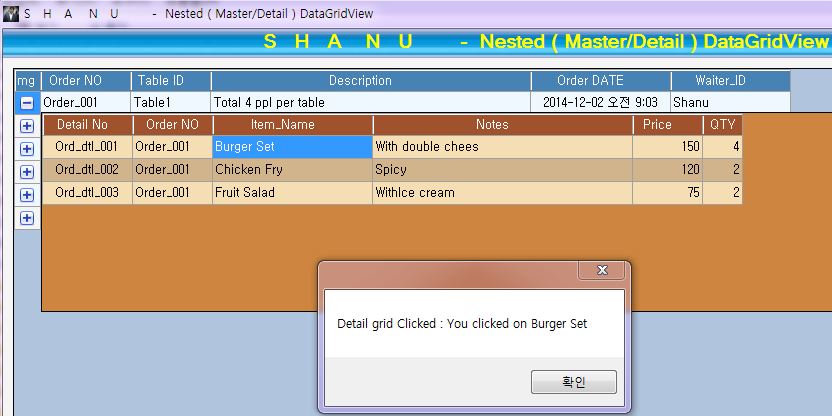
}

private void detailDGVs\_CellContentClick\_Event(object sender, DataGridViewCellEventArgs e)

{

MessageBox.Show("Detail grid Clicked : You clicked on " + DetailDGVs.Rows[e.RowIndex].Cells[e.ColumnIndex].Value);

}



This program explains the basic functions to create single level HierarchicalDataGridView. This same functionality can be used to create multi level HierarchicalDataGridView.