using System;

using System.Collections.Generic;

using System.Windows.Forms;

using System.Drawing;

using System.Collections;

using System.Data;

using System.Text;

namespace PrintDataGrid

{

class PrintDGV

{

private static StringFormat StrFormat; // Holds content of a TextBox Cell to write by DrawString

private static StringFormat StrFormatComboBox; // Holds content of a Boolean Cell to write by DrawImage

private static Button CellButton; // Holds the Contents of Button Cell

private static CheckBox CellCheckBox; // Holds the Contents of CheckBox Cell

private static ComboBox CellComboBox; // Holds the Contents of ComboBox Cell

private static int TotalWidth; // Summation of Columns widths

private static int RowPos; // Position of currently printing row

private static bool NewPage; // Indicates if a new page reached

private static int PageNo; // Number of pages to print

private static ArrayList ColumnLefts = new ArrayList(); // Left Coordinate of Columns

private static ArrayList ColumnWidths = new ArrayList(); // Width of Columns

private static ArrayList ColumnTypes = new ArrayList(); // DataType of Columns

private static int CellHeight; // Height of DataGrid Cell

private static int RowsPerPage; // Number of Rows per Page

private static System.Drawing.Printing.PrintDocument printDoc =

new System.Drawing.Printing.PrintDocument(); // PrintDocumnet Object used for printing

private static string PrintTitle = ""; // Header of pages

private static DataGridView dgv; // Holds DataGridView Object to print its contents

private static List<string> SelectedColumns = new List<string>(); // The Columns Selected by user to print.

private static List<string> AvailableColumns = new List<string>(); // All Columns avaiable in DataGrid

private static bool PrintAllRows = true; // True = print all rows, False = print selected rows

private static bool FitToPageWidth = true; // True = Fits selected columns to page width , False = Print columns as showed

private static int HeaderHeight = 0;

public static void Print\_DataGridView(DataGridView dgv1)

{

PrintPreviewDialog ppvw;

try

{

// Getting DataGridView object to print

dgv = dgv1;

// Getting all Coulmns Names in the DataGridView

AvailableColumns.Clear();

foreach (DataGridViewColumn c in dgv.Columns)

{

if (!c.Visible) continue;

AvailableColumns.Add(c.HeaderText);

}

// Showing the PrintOption Form

PrintOptions dlg = new PrintOptions(AvailableColumns);

if (dlg.ShowDialog() != DialogResult.OK) return;

PrintTitle = dlg.PrintTitle;

PrintAllRows = dlg.PrintAllRows;

FitToPageWidth = dlg.FitToPageWidth;

SelectedColumns = dlg.GetSelectedColumns();

RowsPerPage = 0;

ppvw = new PrintPreviewDialog();

ppvw.Document = printDoc;

// Showing the Print Preview Page

printDoc.BeginPrint +=new System.Drawing.Printing.PrintEventHandler(PrintDoc\_BeginPrint);

printDoc.PrintPage +=new System.Drawing.Printing.PrintPageEventHandler(PrintDoc\_PrintPage);

if (ppvw.ShowDialog() != DialogResult.OK)

{

printDoc.BeginPrint -= new System.Drawing.Printing.PrintEventHandler(PrintDoc\_BeginPrint);

printDoc.PrintPage -= new System.Drawing.Printing.PrintPageEventHandler(PrintDoc\_PrintPage);

return;

}

// Printing the Documnet

printDoc.Print();

printDoc.BeginPrint -= new System.Drawing.Printing.PrintEventHandler(PrintDoc\_BeginPrint);

printDoc.PrintPage -= new System.Drawing.Printing.PrintPageEventHandler(PrintDoc\_PrintPage);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

}

}

private static void PrintDoc\_BeginPrint(object sender,

System.Drawing.Printing.PrintEventArgs e)

{

try

{

// Formatting the Content of Text Cell to print

StrFormat = new StringFormat();

StrFormat.Alignment = StringAlignment.Near;

StrFormat.LineAlignment = StringAlignment.Center;

StrFormat.Trimming = StringTrimming.EllipsisCharacter;

// Formatting the Content of Combo Cells to print

StrFormatComboBox = new StringFormat();

StrFormatComboBox.LineAlignment = StringAlignment.Center;

StrFormatComboBox.FormatFlags = StringFormatFlags.NoWrap;

StrFormatComboBox.Trimming = StringTrimming.EllipsisCharacter;

ColumnLefts.Clear();

ColumnWidths.Clear();

ColumnTypes.Clear();

CellHeight = 0;

RowsPerPage = 0;

// For various column types

CellButton = new Button();

CellCheckBox = new CheckBox();

CellComboBox = new ComboBox();

// Calculating Total Widths

TotalWidth = 0;

foreach (DataGridViewColumn GridCol in dgv.Columns)

{

if (!GridCol.Visible) continue;

if (!PrintDGV.SelectedColumns.Contains(GridCol.HeaderText)) continue;

TotalWidth += GridCol.Width;

}

PageNo = 1;

NewPage = true;

RowPos = 0;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private static void PrintDoc\_PrintPage(object sender,

System.Drawing.Printing.PrintPageEventArgs e)

{

int tmpWidth, i;

int tmpTop = e.MarginBounds.Top;

int tmpLeft = e.MarginBounds.Left;

try

{

// Before starting first page, it saves Width & Height of Headers and CoulmnType

if (PageNo == 1)

{

foreach (DataGridViewColumn GridCol in dgv.Columns)

{

if (!GridCol.Visible) continue;

// Skip if the current column not selected

if (!PrintDGV.SelectedColumns.Contains(GridCol.HeaderText)) continue;

// Detemining whether the columns are fitted to page or not.

if (FitToPageWidth)

tmpWidth = (int)(Math.Floor((double)((double)GridCol.Width /

(double)TotalWidth \* (double)TotalWidth \*

((double)e.MarginBounds.Width / (double)TotalWidth))));

else

tmpWidth = GridCol.Width;

HeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,

GridCol.InheritedStyle.Font, tmpWidth).Height) + 11;

// Save width & height of headres and ColumnType

ColumnLefts.Add(tmpLeft);

ColumnWidths.Add(tmpWidth);

ColumnTypes.Add(GridCol.GetType());

tmpLeft += tmpWidth;

}

}

// Printing Current Page, Row by Row

while (RowPos <= dgv.Rows.Count - 1)

{

DataGridViewRow GridRow = dgv.Rows[RowPos];

if (GridRow.IsNewRow || (!PrintAllRows && !GridRow.Selected))

{

RowPos++;

continue;

}

CellHeight = GridRow.Height;

if (tmpTop + CellHeight >= e.MarginBounds.Height + e.MarginBounds.Top)

{

DrawFooter(e, RowsPerPage);

NewPage = true;

PageNo++;

e.HasMorePages = true;

return;

}

else

{

if (NewPage)

{

// Draw Header

e.Graphics.DrawString(PrintTitle, new Font(dgv.Font, FontStyle.Bold),

Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -

e.Graphics.MeasureString(PrintTitle, new Font(dgv.Font,

FontStyle.Bold), e.MarginBounds.Width).Height - 13);

String s = DateTime.Now.ToLongDateString() + " " + DateTime.Now.ToShortTimeString();

e.Graphics.DrawString(s, new Font(dgv.Font, FontStyle.Bold),

Brushes.Black, e.MarginBounds.Left + (e.MarginBounds.Width -

e.Graphics.MeasureString(s, new Font(dgv.Font,

FontStyle.Bold), e.MarginBounds.Width).Width), e.MarginBounds.Top -

e.Graphics.MeasureString(PrintTitle, new Font(new Font(dgv.Font,

FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);

// Draw Columns

tmpTop = e.MarginBounds.Top;

i = 0;

foreach (DataGridViewColumn GridCol in dgv.Columns)

{

if (!GridCol.Visible) continue;

if (!PrintDGV.SelectedColumns.Contains(GridCol.HeaderText))

continue;

e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),

new Rectangle((int) ColumnLefts[i], tmpTop,

(int)ColumnWidths[i], HeaderHeight));

e.Graphics.DrawRectangle(Pens.Black,

new Rectangle((int) ColumnLefts[i], tmpTop,

(int)ColumnWidths[i], HeaderHeight));

e.Graphics.DrawString(GridCol.HeaderText, GridCol.InheritedStyle.Font,

new SolidBrush(GridCol.InheritedStyle.ForeColor),

new RectangleF((int)ColumnLefts[i], tmpTop,

(int)ColumnWidths[i], HeaderHeight), StrFormat);

i++;

}

NewPage = false;

tmpTop += HeaderHeight;

}

// Draw Columns Contents

i = 0;

foreach (DataGridViewCell Cel in GridRow.Cells)

{

if (!Cel.OwningColumn.Visible) continue;

if (!SelectedColumns.Contains(Cel.OwningColumn.HeaderText))

continue;

// For the TextBox Column

if (((Type) ColumnTypes[i]).Name == "DataGridViewTextBoxColumn" ||

((Type) ColumnTypes[i]).Name == "DataGridViewLinkColumn")

{

e.Graphics.DrawString(Cel.Value.ToString(), Cel.InheritedStyle.Font,

new SolidBrush(Cel.InheritedStyle.ForeColor),

new RectangleF((int)ColumnLefts[i], (float)tmpTop,

(int)ColumnWidths[i], (float)CellHeight), StrFormat);

}

// For the Button Column

else if (((Type) ColumnTypes[i]).Name == "DataGridViewButtonColumn")

{

CellButton.Text = Cel.Value.ToString();

CellButton.Size = new Size((int)ColumnWidths[i], CellHeight);

Bitmap bmp =new Bitmap(CellButton.Width, CellButton.Height);

CellButton.DrawToBitmap(bmp, new Rectangle(0, 0,

bmp.Width, bmp.Height));

e.Graphics.DrawImage(bmp, new Point((int)ColumnLefts[i], tmpTop));

}

// For the CheckBox Column

else if (((Type) ColumnTypes[i]).Name == "DataGridViewCheckBoxColumn")

{

CellCheckBox.Size = new Size(14, 14);

CellCheckBox.Checked = (bool)Cel.Value;

Bitmap bmp = new Bitmap((int)ColumnWidths[i], CellHeight);

Graphics tmpGraphics = Graphics.FromImage(bmp);

tmpGraphics.FillRectangle(Brushes.White, new Rectangle(0, 0,

bmp.Width, bmp.Height));

CellCheckBox.DrawToBitmap(bmp,

new Rectangle((int)((bmp.Width - CellCheckBox.Width) / 2),

(int)((bmp.Height - CellCheckBox.Height) / 2),

CellCheckBox.Width, CellCheckBox.Height));

e.Graphics.DrawImage(bmp, new Point((int)ColumnLefts[i], tmpTop));

}

// For the ComboBox Column

else if (((Type) ColumnTypes[i]).Name == "DataGridViewComboBoxColumn")

{

CellComboBox.Size = new Size((int)ColumnWidths[i], CellHeight);

Bitmap bmp = new Bitmap(CellComboBox.Width, CellComboBox.Height);

CellComboBox.DrawToBitmap(bmp, new Rectangle(0, 0,

bmp.Width, bmp.Height));

e.Graphics.DrawImage(bmp, new Point((int)ColumnLefts[i], tmpTop));

e.Graphics.DrawString(Cel.Value.ToString(), Cel.InheritedStyle.Font,

new SolidBrush(Cel.InheritedStyle.ForeColor),

new RectangleF((int)ColumnLefts[i] + 1, tmpTop, (int)ColumnWidths[i]

- 16, CellHeight), StrFormatComboBox);

}

// For the Image Column

else if (((Type) ColumnTypes[i]).Name == "DataGridViewImageColumn")

{

Rectangle CelSize = new Rectangle((int)ColumnLefts[i],

tmpTop, (int)ColumnWidths[i], CellHeight);

Size ImgSize = ((Image)(Cel.FormattedValue)).Size;

e.Graphics.DrawImage((Image)Cel.FormattedValue,

new Rectangle((int)ColumnLefts[i] + (int)((CelSize.Width - ImgSize.Width) / 2),

tmpTop + (int)((CelSize.Height - ImgSize.Height) / 2),

((Image)(Cel.FormattedValue)).Width, ((Image)(Cel.FormattedValue)).Height));

}

// Drawing Cells Borders

e.Graphics.DrawRectangle(Pens.Black, new Rectangle((int)ColumnLefts[i],

tmpTop, (int)ColumnWidths[i], CellHeight));

i++;

}

tmpTop += CellHeight;

}

RowPos++;

// For the first page it calculates Rows per Page

if (PageNo == 1) RowsPerPage++;

}

if (RowsPerPage == 0) return;

// Write Footer (Page Number)

DrawFooter(e, RowsPerPage);

e.HasMorePages = false;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private static void DrawFooter(System.Drawing.Printing.PrintPageEventArgs e,

int RowsPerPage)

{

double cnt = 0;

// Detemining rows number to print

if (PrintAllRows)

{

if (dgv.Rows[dgv.Rows.Count - 1].IsNewRow)

cnt = dgv.Rows.Count - 2; // When the DataGridView doesn't allow adding rows

else

cnt = dgv.Rows.Count - 1; // When the DataGridView allows adding rows

}

else

cnt = dgv.SelectedRows.Count;

// Writing the Page Number on the Bottom of Page

string PageNum = PageNo.ToString() + " of " +

Math.Ceiling((double)(cnt / RowsPerPage)).ToString();

e.Graphics.DrawString(PageNum, dgv.Font, Brushes.Black,

e.MarginBounds.Left + (e.MarginBounds.Width -

e.Graphics.MeasureString(PageNum, dgv.Font,

e.MarginBounds.Width).Width) / 2, e.MarginBounds.Top +

e.MarginBounds.Height + 31);

}

}

}