Grid Printing

```
/*
Professional Windows GUI Programming Using C#
by Jay Glynn, Csaba Torok, Richard Conway, Wahid Choudhury,
Zach Greenvoss, Shripad Kulkarni, Neil Whitlow

Publisher: Peer Information
ISBN: 1861007663
*/
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

using System.Data.SqlClient;
using System.Drawing.Printing;

namespace GridPrinting
{
///

/// Summary description for Form1.
///
public class GridPrintingDemo : System.Windows.Forms.Form
{
//Stored Page Settings object
private PageSettings oPageSettings;

private System.Windows.Forms.MenuItem menuItem1;
private System.Windows.Forms.MenuItem menuItem2;
private System.ComponentModel.Container components = null;
private System.Windows.Forms.MainMenu mnuMain;
private System.Windows.Forms.PrintPreviewDialog ppDialog;
```

```
private System.Windows.Forms.MenuItem menuItem3;
private System.Windows.Forms.MenuItem menuItem4;
private System.Windows.Forms.PrintDialog oPrintDialog;
private System.Windows.Forms.MenuItem menuItem5;
private System.Windows.Forms.MenuItem menuItem6;
private System.Windows.Forms.PageSetupDialog oPageSetup;
private System.Windows.Forms.DataGrid dGrid;

public GridPrintingDemo()
{
//
// Required for Windows Form Designer support
//
InitializeComponent();

oPageSettings = new PageSettings();
}

///

/// Clean up any resources being used.
///
protected override void Dispose( bool disposing )
{
if( disposing )
{
if (components != null)
{
components.Dispose();
}
}
base.Dispose( disposing );
}

#region Windows Form Designer generated code
///

/// Required method for Designer support – do not modify
/// the contents of this method with the code editor.
```

```
///
private void InitializeComponent()
{
//  System.Resources.ResourceManager  resources  =  new
System.Resources.ResourceManager(typeof(GridPrintingDemo));
this.dGrid = new System.Windows.Forms.DataGrid();
this.mnuMain = new System.Windows.Forms.MainMenu();
this.menuItem1 = new System.Windows.Forms.MenuItem();
this.menuItem3 = new System.Windows.Forms.MenuItem();
this.menuItem4 = new System.Windows.Forms.MenuItem();
this.menuItem2 = new System.Windows.Forms.MenuItem();
this.menuItem5 = new System.Windows.Forms.MenuItem();
this.menuItem6 = new System.Windows.Forms.MenuItem();
this.ppDialog = new System.Windows.Forms.PrintPreviewDialog();
this.oPrintDialog = new System.Windows.Forms.PrintDialog();
this.oPageSetup = new System.Windows.Forms.PageSetupDialog();
((System.ComponentModel.ISupportInitialize)(this.dGrid)).Begin
Init();
this.SuspendLayout();
//
// dGrid
//
this.dGrid.DataMember = “”;
this.dGrid.Dock = System.Windows.Forms.DockStyle.Fill;
this.dGrid.HeaderForeColor                                    =
System.Drawing.SystemColors.ControlText;
this.dGrid.Name = “dGrid”;
this.dGrid.Size = new System.Drawing.Size(292, 266);
this.dGrid.TabIndex = 0;
//
// mnuMain
//
this.mnuMain.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
this.menuItem1});
//
// menuItem1
```

```
//
this.menuItem1.Index = 0;
this.menuItem1.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
this.menuItem3,
this.menuItem4,
this.menuItem2,
this.menuItem5,
this.menuItem6});
this.menuItem1.Text = “&File”;
//
// menuItem3
//
this.menuItem3.Index = 0;
this.menuItem3.Text = “Page Setup…”;
this.menuItem3.Click                    +=                    new
System.EventHandler(this.menuItem3_Click);
//
// menuItem4
//
this.menuItem4.Index = 1;
this.menuItem4.Text = “Print…”;
this.menuItem4.Click                    +=                    new
System.EventHandler(this.menuItem4_Click);
//
// menuItem2
//
this.menuItem2.Index = 2;
this.menuItem2.Text = “Print Preview…”;
this.menuItem2.Click                    +=                    new
System.EventHandler(this.menuItem2_Click);
//
// menuItem5
//
this.menuItem5.Index = 3;
this.menuItem5.Text = “-“;
//
```

```
// menuItem6
//
this.menuItem6.Index = 4;
this.menuItem6.Text = “Exit”;
//
// ppDialog
//
this.ppDialog.AutoScrollMargin = new System.Drawing.Size(0,
0);
this.ppDialog.AutoScrollMinSize = new System.Drawing.Size(0,
0);
this.ppDialog.ClientSize = new System.Drawing.Size(400, 300);
this.ppDialog.Enabled = true;
//                 this.ppDialog.Icon                 =
((System.Drawing.Icon)(resources.GetObject(“ppDialog.Icon”)));
this.ppDialog.Location = new System.Drawing.Point(121, 21);
this.ppDialog.MaximumSize = new System.Drawing.Size(0, 0);
this.ppDialog.Name = “ppDialog”;
this.ppDialog.Opacity = 1;
this.ppDialog.TransparencyKey = System.Drawing.Color.Empty;
this.ppDialog.Visible = false;
//
// oPrintDialog
//
this.oPrintDialog.AllowSomePages = true;
//
// GridPrintingDemo
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(292, 266);
this.Controls.AddRange(new System.Windows.Forms.Control[] {
this.dGrid});
this.Menu = this.mnuMain;
this.Name = “GridPrintingDemo”;
this.Text = “Printing Grid Example”;
this.Load += new System.EventHandler(this.Form1_Load);
((System.ComponentModel.ISupportInitialize)(this.dGrid)).EndIn
```

```
it();
this.ResumeLayout(false);

}
#endregion

///

/// The main entry point for the application.
///
[STAThread]
static void Main()
{
Application.Run(new GridPrintingDemo());
}

private void Form1_Load(object sender, System.EventArgs e)
{
try
{
//COnnection to database – get dataset
SqlConnection aConn = new SqlConnection("data source=HOBBIT;initial catalog=Northwind;integrated security=SSPI;");
SqlDataAdapter sqlAdapter = new SqlDataAdapter("Select CompanyName, Address, City, Region, PostalCode, Country from Customers",aConn);
DataSet aDataSet = new DataSet();
sqlAdapter.Fill(aDataSet);

//Assign dataset to Grid
dGrid.DataSource = aDataSet;
dGrid.SetDataBinding(aDataSet.Tables[0],"");
}
catch(Exception ex)
{
MessageBox.Show(ex.ToString(),"Error");
}
```

```
}

private void menuItem2_Click(object sender, System.EventArgs
e)
{
//Setup the document to print
PrintGridDocument aDoc = new PrintGridDocument(dGrid);
aDoc.Title = “Employee Report”;
oPrintDialog.Document = aDoc;
if (oPrintDialog.ShowDialog() == DialogResult.OK)
{
//Display the print preview dialog
aDoc.DefaultPageSettings = oPageSettings;
ppDialog.Document = aDoc;
ppDialog.ShowDialog();
}
}

private void menuItem3_Click(object sender, System.EventArgs
e)
{
oPageSetup.PageSettings = oPageSettings;
if(oPageSetup.ShowDialog() == DialogResult.OK)
{
oPageSettings = oPageSetup.PageSettings;
}
}

private void menuItem4_Click(object sender, System.EventArgs
e)
{
//Setup the document to print
PrintGridDocument aDoc = new PrintGridDocument(dGrid);
aDoc.Title = “Employee Report”;
oPrintDialog.Document = aDoc;
if (oPrintDialog.ShowDialog() == DialogResult.OK)
{
//Display the print preview dialog
```

```
aDoc.DefaultPageSettings = oPageSettings;
try
{
//Print the document
aDoc.Print();
}
catch(Exception ex)
{
//Display any errors
MessageBox.Show(ex.ToString());
}
}
}
}

public class PrintGridDocument : PrintDocument
{
//Data Members
private DataGrid m_oDataGrid;
private int m_nCurrPage;
private int m_nCurrRow;
private int m_nColumns;
private int m_nRows;
private bool m_bInitialized;
private int m_nLinesPerPage;
private int m_nTotalPages;
private int[] m_nColBounds;

//Properties
public Font PrintFont;
public string Title;

public PrintGridDocument(DataGrid aGrid) : base()
{
//Default Values
m_oDataGrid = aGrid;
m_nCurrPage = 0;
m_nCurrRow = 0;
```

```
m_bInitialized = false;

//Get total number of cols/rows in the data source
m_nColumns                                                    =
((DataTable)(m_oDataGrid.DataSource)).Columns.Count;
m_nRows = ((DataTable)(m_oDataGrid.DataSource)).Rows.Count;
}

//Override OnBeginPrint to set up the font we are going to use
protected override void OnBeginPrint(PrintEventArgs ev)
{
base.OnBeginPrint(ev);
//If client has not created a font, create a default font
// Note: an exception could be raised here, but it is
deliberately not
// being caught because there is nothing we could do at this
point!
if(PrintFont == null)
PrintFont = new Font("Arial", 9);

}

//Override the OnPrintPage to provide the printing logic for
the document
protected override void OnPrintPage(PrintPageEventArgs e)
{
//Call base method
base.OnPrintPage(e);

//Get the margins
int nTextPosX = e.MarginBounds.Left;
int nTextPosY = e.MarginBounds.Top;

//Do first time initialization stuff
if(!m_bInitialized)
{
// Calculate the number of lines per page.
m_nLinesPerPage   =   (int)(e.MarginBounds.Height   /
PrintFont.GetHeight(e.Graphics));
```

```
m_nTotalPages = (int)Math.Ceiling((float)m_nRows / (float)m_nLinesPerPage);

//Create bounding box for columns
m_nColBounds = new int[m_nColumns];

//Calculate the correct spacing for the columns
for(int nCol = 0;nCol e.MarginBounds.Width / m_nColumns)
m_nColBounds[nCol] = e.MarginBounds.Width / m_nColumns;

//Can't be less than column width
if(m_nColBounds[nCol] < (int)Math.Round(e.Graphics.MeasureString(((DataTable)(m_oDataGrid.DataSource)).Columns[nCol].ColumnName, PrintFont).Width))
m_nColBounds[nCol] = (int)Math.Round(e.Graphics.MeasureString(((DataTable)(m_oDataGrid.DataSource)).Columns[nCol].ColumnName, PrintFont).Width);
} //Move to correct starting page
if(this.PrinterSettings.PrintRange == PrintRange.SomePages) {
while(m_nCurrPage < this.PrinterSettings.FromPage-1) { //Move to next page - advance data to next page as well m_nCurrRow += m_nLinesPerPage; m_nCurrPage++; if(m_nCurrRow > m_nRows)
return;
}

if(m_nCurrPage > this.PrinterSettings.ToPage)
{
//Don't print anything more
return;
}
}

//Set flag
m_bInitialized = true;
}

//Move to next page
m_nCurrPage++;
```

```
//Print Title if first page
if(m_nCurrPage == 1)
{
Font TitleFont = new Font(“Arial”,15);
int nXPos = (int)(((e.PageBounds.Right – e.PageBounds.Left) /2
) –
(e.Graphics.MeasureString(Title,TitleFont).Width / 2));
e.Graphics.DrawString(Title,TitleFont,Brushes.Black,nXPos,e.Ma
rginBounds.Top – TitleFont.GetHeight(e.Graphics) – 10);
}

//Draw page number
string strOutput = “Page ” + m_nCurrPage + ” of ” +
m_nTotalPages;
e.Graphics.DrawString(strOutput,PrintFont,Brushes.Black,e.Marg
inBounds.Right –
e.Graphics.MeasureString(strOutput,PrintFont).Width,
e.MarginBounds.Bottom);

//Utility rectangle – use for many drawing operations
Rectangle aRect = new Rectangle();

//Loop through data
for(int nRow=m_nCurrRow; nRow < m_nRows; nRow++) { //Draw the
current row within a shaded/unshaded box aRect.X =
e.MarginBounds.Left; aRect.Y = nTextPosY; aRect.Width =
e.MarginBounds.Width; aRect.Height =
(int)PrintFont.GetHeight(e.Graphics); //Draw the box if(nRow%2
== 0) e.Graphics.FillRectangle(Brushes.LightGray,aRect);
e.Graphics.DrawRectangle(Pens.Black,aRect); //Loop through
each column for(int nCol=0; nCol < m_nColumns; nCol++) { //Set
the rectangle to the correct position aRect.X = nTextPosX;
aRect.Y = nTextPosY; aRect.Width = m_nColBounds[nCol];
aRect.Height = (int)PrintFont.GetHeight(e.Graphics); //Print
the data
e.Graphics.DrawString(m_oDataGrid[nRow,nCol].ToString(),PrintF
ont,Brushes.Black,aRect); //Advance the x Position counter
nTextPosX += m_nColBounds[nCol]; } //Reassign the x position
```

counter nTextPosX = e.MarginBounds.Left; //Move the y position counter down a line nTextPosY += (int)PrintFont.GetHeight(e.Graphics); //Check to see if we have reached the line limit - move to a new page if so if(nRow - ((m_nCurrPage-1) * m_nLinesPerPage) == m_nLinesPerPage) { //Save the current row m_nCurrRow = ++nRow; e.HasMorePages = true; return; } } } } } [/csharp]