

Get repainted regions



```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Drawing.Imaging;

public class Form1 : System.Windows.Forms.Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private void InitializeComponent()
    {
        this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
        this.ClientSize = new System.Drawing.Size(292, 273);
        this.Text = "";
        this.Resize += new System.EventHandler(this.Form1_Resize);
        this.Paint += new System.Windows.Forms.PaintEventHandler(this.Form1_Paint);
    }
    static void Main()
    {
        Application.Run(new Form1());
    }

    private void Form1_Paint(object sender,
        System.Windows.Forms.PaintEventArgs e)
    {

```

```

Graphics g = e.Graphics;

g.FillRectangle(Brushes.White, this.ClientRectangle);

SizeF sizeF = g.MeasureString("AA", this.Font);
StringFormat sf = new StringFormat();
sf.LineAlignment = StringAlignment.Center;

// Set properties of the grid
int cellHeight = (int)sizeF.Height + 200;
int cellWidth = 200;
int nbrColumns = 50;
int nbrRows = 50;

// Output general info to console
Console.WriteLine("-----");
Console.WriteLine("e.ClipRectangle = " + e.ClipRectangle);
Console.WriteLine("The following cells need to be redrawn " +
"(in whole or in part):");
// Draw the cells and the output to console
for (int row = 0; row < nbrRows; ++row) { for (int col = 0;
col < nbrColumns; ++col) { Point cellLocation = new Point(col
* cellWidth, row * cellHeight); Rectangle cellRect = new
Rectangle(cellLocation.X, cellLocation.Y, cellWidth,
cellHeight); if (cellRect.Intersects(e.ClipRectangle)) {
Console.WriteLine("Row:{0} Col:{1}", row, col);
g.FillRectangle(Brushes.White, cellRect);
g.DrawRectangle(Pens.Black, cellRect); String s =
String.Format("{0},{1}", col, row); g.DrawString(s, this.Font,
Brushes.Black, cellRect, sf); } } } } private void
Form1_Resize(object sender, EventArgs e) {
Invalidate(); } } [/csharp]

```