

Get all known color

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

public class Form1 : System.Windows.Forms.Form {
    ArrayList knownColorList = new ArrayList();
    ArrayList knownColorNameList = new ArrayList();

    public Form1() {
        NonSystemColors(knownColorList, knownColorNameList);
    }
    [STAThread]
    static void Main() {
        Application.Run(new Form1());
    }
    private void NonSystemColors(ArrayList knownColorList,
        ArrayList knownColorNameList) {
        Array cA = Enum.GetValues(typeof(KnownColor));
        foreach (KnownColor knwnC in cA) {
            Color curC = Color.FromKnownColor(knwnC);
            if (!curC.IsSystemColor) {
                knownColorList.Add(curC);
                knownColorNameList.Add(curC.Name.ToString());
            }
        }
    }
    protected override void OnPaint(PaintEventArgs pea) {
        Graphics g = pea.Graphics;
        int wi = 70, hi = 12, rectNb = 8;
        int count = knownColorList.Count;
```

```

this.Width = (wi + 2) * rectNb + 9;
int y = (int)(count / rectNb);
this.Height = y * (2 + hi) + 60;

DisplayKnownColors(g, count, wi, hi, rectNb);
g.Dispose();
}
private void DisplayKnownColors(Graphics g, int count, int wi,
int hi, int rectNb) {
Rectangle rec;
Pen p = new Pen(this.ForeColor);
Brush b;

StringFormat strfmt = new StringFormat();
strfmt.LineAlignment = strfmt.Alignment =
StringAlignment.Near;

int x, y;
for (int i = 0; i < count; i++) { x = (int)(i % rectNb); y =
(int)(i / rectNb); rec = new Rectangle(1 + x * (2 + wi), 1 + y
* (2 + hi), wi, hi); g.DrawRectangle(p, rec); b = new
SolidBrush((Color)knownColorList[i]); g.FillRectangle(b, rec);
b = new SolidBrush(Color.Black);
g.DrawString((string)knownColorNameList[i], this.Font, b, rec,
strfmt); } } } [/csharp]

```