

# Hexagon Gradient Brush

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;

class HexagonGradientBrush: Form
{
    const float fSide = 50; // Side (also radius) of hexagon

    public static void Main()
    {
        Application.Run(new HexagonGradientBrush());
    }
    public HexagonGradientBrush()
    {
        ResizeRedraw = true;
    }
    protected override void OnPaint(PaintEventArgs pea)
    {
        DoPage(pea.Graphics,          ForeColor, ClientSize.Width,
        ClientSize.Height);
    }
    protected void DoPage(Graphics grfx, Color clr, int cx, int
    cy)
    {
        float fHalf = fSide * (float) Math.Sin(Math.PI / 3);

        PointF[] aptf = {new PointF( fSide, 0),
        new PointF( fSide * 1.5f, 0),
        new PointF( fSide, 0),
        new PointF( fSide / 2, -fHalf),
        new PointF(-fSide / 2, -fHalf),
        new PointF(-fSide, 0),
        new PointF(-fSide * 1.5f, 0),
```

```
new PointF(-fSide, 0),
new PointF(-fSide / 2, fHalf),
new PointF( fSide / 2, fHalf) };

PathGradientBrush pbrush1 = new PathGradientBrush(aptf,
WrapMode.Tile);

for (int i = 0; i < aptf.Length; i++) { aptf[i].X += fSide *
1.5f; aptf[i].Y += fHalf; } PathGradientBrush pbrush2 = new
PathGradientBrush(aptf, WrapMode.Tile);
grfx.FillRectangle(pbrush1, 0, 0, cx, cy);
grfx.FillRectangle(pbrush2, 0, 0, cx, cy); } } [/csharp]
```