

# Clipping Combinations

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

class ClippingCombinations: Form
{
    string strCaption = "CombineMode = ";
    MenuItem miCombineMode;

    public static void Main()
    {
        Application.Run(new ClippingCombinations());
    }
    protected override void OnPaint(PaintEventArgs pea)
    {
        DoPage(pea.Graphics,          ForeColor, ClientSize.Width,
        ClientSize.Height);
    }
    public ClippingCombinations()
    {
        ResizeRedraw = true;

        Menu = new MainMenu();
        Menu.MenuItems.Add("&CombineMode");

        EventHandler          ehClick          =          new
        EventHandler(MenuCombineModeOnClick);

        for (int i = 0; i < 6; i++) { MenuItem mi = new MenuItem("&" +
        (CombineMode)i); mi.Click += ehClick; mi.RadioCheck = true;
        Menu.MenuItems[0].MenuItems.Add(mi); } miCombineMode =
        Menu.MenuItems[0].MenuItems[0]; miCombineMode.Checked = true;
    } void MenuCombineModeOnClick(object obj, EventArgs ea) {
        miCombineMode.Checked = false; miCombineMode = (MenuItem) obj;
```

```
miCombineMode.Checked = true; Text = strCaption +  
(CombineMode)miCombineMode.Index; Invalidate(); } protected  
void DoPage(Graphics grfx, Color clr, int cx, int cy) {  
GraphicsPath path = new GraphicsPath(); path.AddEllipse(0, 0,  
2 * cx / 3, cy); grfx.SetClip(path); path.Reset();  
path.AddEllipse(cx / 3, 0, 2 * cx / 3, cy); grfx.SetClip(path,  
(CombineMode)miCombineMode.Index);  
grfx.FillRectangle(Brushes.Red, 0, 0, cx, cy); } } [/csharp]
```