

Graphics.CopyFromScreen Method (System.Drawing)

By dotnet-bot

Archived: 2026-04-05 19:17:42 UTC

CopyFromScreen(Int32, Int32, Int32, Int32, Size)

Source:

[Graphics.cs](#)

Source:

[Graphics.cs](#)

Source:

[Graphics.cs](#)

Source:

[Graphics.cs](#)

Source:

[Graphics.cs](#)

Source:

[Graphics.cs](#)

Source:

[Graphics.cs](#)

Source:

[Graphics.cs](#)

Performs a bit-block transfer of the color data, corresponding to a rectangle of pixels, from the screen to the drawing surface of the [Graphics](#).

```
public void CopyFromScreen(int sourceX, int sourceY, int destinationX, int destinationY, System.Drawing.Size b
```

Parameters

sourceX

[Int32](#)

The x-coordinate of the point at the upper-left corner of the source rectangle.

sourceY

[Int32](#)

The y-coordinate of the point at the upper-left corner of the source rectangle.

destinationX

[Int32](#)

The x-coordinate of the point at the upper-left corner of the destination rectangle.

destinationY

[Int32](#)

The y-coordinate of the point at the upper-left corner of the destination rectangle.

blockRegionSize

[Size](#)

The size of the area to be transferred.

Exceptions

Examples

The following code example demonstrates how to use the [CopyFromScreen](#) for printing a copy of the current form.

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

public class Form1 :
    Form
{
    private Button printButton = new Button();
    private PrintDocument printDocument1 = new PrintDocument();

    public Form1()
    {
        printButton.Text = "Print Form";
        printButton.Click += new EventHandler(printButton_Click);
        printDocument1.PrintPage += new PrintPageEventHandler(printDocument1_PrintPage);
        this.Controls.Add(printButton);
    }

    void printButton_Click(object sender, EventArgs e)
    {
```

```
        CaptureScreen();
        printDocument1.Print();
    }

    Bitmap memoryImage;

    private void CaptureScreen()
    {
        Graphics myGraphics = this.CreateGraphics();
        Size s = this.Size;
        memoryImage = new Bitmap(s.Width, s.Height, myGraphics);
        Graphics memoryGraphics = Graphics.FromImage(memoryImage);
        memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, s);
    }

    private void printDocument1_PrintPage(System.Object sender,
        System.Drawing.Printing.PrintPageEventArgs e)
    {
        e.Graphics.DrawImage(memoryImage, 0, 0);
    }

    public static void Main()
    {
        Application.Run(new Form1());
    }
}
```

Remarks

The [CopyFromScreen](#) methods are useful for layering one image on top of another. To specify how the source and destination colors are blended, use one of the [CopyFromScreen](#) methods that takes a [CopyPixelOperation](#) parameter.

See also

- [How to: Print a Windows Form](#)
- [How to: Copy Pixels For Reducing Flicker in Windows Forms](#)

Applies to

► .NET 11 (package-provided) and other versions