

ラスタオペレーション

ピクセルの描画は、描画される前に**ある二つの値が比較**されます
それは、**描画される色** (ソース)と**画面の色** (ディスティネーション)です

簡単に考えて、白を0黒を1として比較するとします
描画しようとしている色と画面の色を比較して、その結果で描画命令を実行するとすれば
ピクセルレベルの操作を全体的に、しかも簡易に行うことができます

このようなピクセルレベルのビットの操作を**ラスタオペレーション** (ROP)と呼ばれ
2つのピクセルを操作することを**バイナリラスタオペレーション** (ROP2)と呼びます
2つのピクセルとは、描画する色と画面の色の二つです

さて、このラスタオペレーションで一体何ができるのでしょうか
例えば、画面の色が白の時は黒を出力し、画面の色が黒の時は白を出力するというような
排他的な論理演算による操作などは、このラスタオペレーションで行うことができます

この操作を実現するのが **System.Drawing.Region** クラスです
このクラスは、リージョン(範囲)情報を保持し
さらに、他のリージョンやパス、長方形などと論理演算して ROP を実現します

```
Object
  MarshalByRefObject
    Region

public sealed class Region : MarshalByRefObject, IDisposable

コンストラクタは、以下のものが定義されています
```

```
public Region();
public Region(GraphicsPath path);
public Region(Rectangle rect);
public Region(RectangleF rect);
public Region(RegionData rgnData);
```

path には、リージョンとなるパスを表す GraphicsPath オブジェクトを
rect には、長方形を示す構造体オブジェクトを指定します
rgnData は、リージョンのデータを示す RegionData オブジェクトを指定します

リージョンは、指定したパスや長方形などで初期化され、インスタンスが作られます
後は、このリージョンに対して指定した矩形と論理演算を行えます

単純に、指定された領域と現在の領域を組み合わせるには **Region.Union()** を
重なる部分を反転させるには **Region.Xor()** メソッドを使います

```
public void Union(GraphicsPath path);
public void Union(Rectangle rect);
public void Union(RectangleF rect);
public void Union(Region region);
```

```
public void Xor(GraphicsPath path);
public void Xor(Rectangle rect);
public void Xor(RectangleF rect);
public void Xor(Region region);
```

path には、現在のリージョンと比較するパスを表す GraphicsPath オブジェクトを
rect には、長方形をあらわす構造体オブジェクトを
region には、リージョンを表す Region オブジェクトを指定することができます

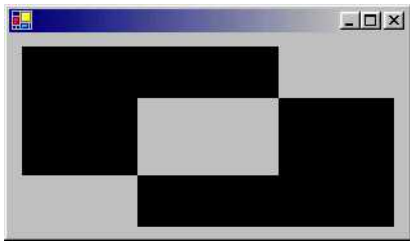
Region オブジェクトは **Graphics.FillRegion()** メソッドで描画できます

```
public void FillRegion(Brush brush, Region region);
```

brush には塗りつぶしに用いる論理ブラシを
region には描画する Region オブジェクトを指定します

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

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }
    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        Region rg = new Region(new Rectangle(10, 10, 200, 100));
        rg.Xor(new Rectangle(100, 50, 200, 100));
        g.FillRegion(Brushes.Black, rg);
    }
}
```



このプログラムは、指定した長方形で Region インスタンスを作成し
さらに、Xor() メソッドで、指定した長方形を排他的に追加しています
その結果、Region は交わった部分を除外して更新されます

この他にも、指定した領域のうち現在の領域から
交わらない部分だけで更新する **Region.Complement()** メソッドや
逆に、現在の領域のうち指定した領域に交わらない部分だけで更新する
Region.Exclude() メソッドなども存在します
これらのパラメータは Union() や Xor() メソッドと同じなので、説明は省略します

[前のページへ](#)

[戻る](#)

[次のページへ](#)