

# アイコン

## アイコンの描画

アイコンとは \*.ico 拡張子であらわされる小さなビットマップのことです  
主に、アプリケーションアイコンなど、GUI の部品に用いられ  
16 色の 32 × 32 ピクセルのアイコンが多く用いられています

ウィンドウの左上にあるアイコンや、最小化したときのアイコンもちろんこれです  
.NET では、アイコンを **System.Drawing.Icon** クラスで表現します

```
System.Object
  System.MarshalByRefObject
    System.Drawing.Icon

[Serializable]
[ComVisible(false)]
public sealed class Icon : MarshalByRefObject,
  ISerializable, ICloneable, IDisposable
```

このクラスは、アイコンの情報や基本的な操作をパッケージ化しています  
基本的には、Bitmap クラスのようにファイル名を指定するだけで \*.ico ファイルを読み込みます  
このクラスのコンストラクタは、次のものが定義されています

```
[Serializable]
[ComVisible(false)]
public Icon(Stream stream);

[Serializable]
[ComVisible(false)]
public Icon(string fileName);

[Serializable]
[ComVisible(false)]
public Icon(Icon original , Size size);

[Serializable]
[ComVisible(false)]
public Icon(Type type , string resource);

[Serializable]
[ComVisible(false)]
public Icon(Icon original , int width , int height);

[Serializable]
[ComVisible(false)]
public Icon(Stream stream , int width , int height);
```

stream には、アイコンの読み込み元を示すデータストリームを指定します  
fileName は、Icon オブジェクトの読み込み元となるファイル名を指定します  
original は新たに作成する Icon オブジェクトの元になる Icon オブジェクトを指定します

size は新しいアイコンのサイズを示す Size オブジェクトを  
width には横幅、height には高さを指定します

type はリソースを検索するアセンブリを指定する Type オブジェクトを  
resource には読み込むリソース名を指定します

Icon オブジェクトは、**Graphics.DrawIcon()** メソッドを使うことで描画できます  
または、スケーリングせずに描画する場合は  
**Graphics.DrawIconUnstretched()** メソッドを使うこともできます

```
[ComVisible(false)]
public void DrawIcon(Icon icon , Rectangle targetRect);

[ComVisible(false)]
public void DrawIcon(Icon icon , int x , int y);

[ComVisible(false)]
public void DrawIconUnstretched(Icon icon , Rectangle targetRect);
```

icon には描画する Icon オブジェクトを、targetRect には描画する長方形を指定します  
x には描画するアイコンの左上の位置を示す X 座標、y には Y 座標を指定します  
DrawIconUnstretched() メソッドの場合は、常にもとのサイズが保持されます

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

class WinMain : Form {
  static Icon ico;
  public static void Main(string[] args) {
    ico = new Icon(args[0]);
    Application.Run(new WinMain());
  }

  override protected void OnPaint(PaintEventArgs e) {
    Graphics g = e.Graphics;
    Rectangle rect = new Rectangle(0 , 0 , 100 , 100);
    g.DrawIcon(ico , rect);

    rect.X = 100;
    g.DrawIconUnstretched(ico , rect);
  }
}
```

```
}
```



このプログラムは、コマンドラインで指定したアイコンファイルを読み込み  
Graphics.DrawIcon() メソッドを使って 100 × 100 にスケーリングして描画し  
さらに、Graphics.DrawIconUnstretched() メソッドでデフォルトサイズでも描画します

Icon クラスには、アイコンのサイズを取得するためのプロパティが公開されています  
横幅は **Icon.Width** 高さは **Icon.Height** で取得できます  
また、これらを Size オブジェクトとして **Icon.Size** で得ることもできます

```
[Serializable]  
[ComVisible(false)]  
public int Width {get;}
```

```
[Serializable]  
[ComVisible(false)]  
public int Height {get;}
```

```
[Serializable]  
[ComVisible(false)]  
public Size Size {get;}
```

これらのプロパティを使えば、アイコンのサイズを取得することができます

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    static Icon ico;  
    public static void Main(string[] args) {  
        ico = new Icon(args[0]);  
        Application.Run(new WinMain());  
    }  
  
    override protected void OnPaint(PaintEventArgs e) {  
        Graphics g = e.Graphics;  
        Rectangle rect = new Rectangle(0, 0, ico.Width, ico.Height);  
        g.DrawIcon(ico, rect);  
    }  
}
```

このプログラムは、単純にコマンドラインから指定したアイコンを描画します  
このとき、描画範囲を長方形で指定していますが  
アイコンのサイズをプロパティで取得しているため、正しいサイズで描画されます

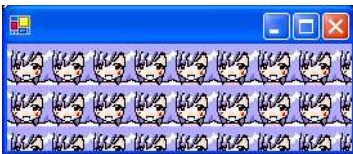
## ビットマップへの変換

Icon クラスは、アイコンをビットマップイメージに変換するメソッドを提供しています  
**Icon.ToBitmap()** メソッドを使えば、Icon オブジェクトのイメージを  
そのまま Bitmap オブジェクトに変換して、複雑な処理をさせることができます

```
[Serializable]  
[ComVisible(false)]  
public Bitmap ToBitmap();
```

Bitmap オブジェクトに変換すれば、GDI+ のさまざまな恩恵を受けられます

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    public static void Main(string[] args) {  
        Icon ico = new Icon(args[0]);  
        WinMain win = new WinMain();  
        win.BackgroundImage = ico.ToBitmap();  
        Application.Run(win);  
    }  
}
```



このプログラムは、コマンドラインで指定したアイコンを Bitmap に変換し  
コントロールの背景イメージとして設定しています

## ウィンドウアイコン

Windows の一般的なアプリケーションのフレームウィンドウには

左上にアイコンが表示され、タスクバーにも同じアイコンが表示されます  
これを設定するには **Form.Icon** プロパティにアイコンを設定します

```
public Icon Icon {get; set;}
```

このプロパティは、ウィンドウのアイコンを取得したり、設定したりすることができます  
これに先ほどの Icon オブジェクトを設定することで、ウィンドウのアイコンを変更できます

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

class WinMain : Form {
    public static void Main(string[] args) {
        WinMain win = new WinMain();
        win.Icon = new Icon(args[0]);
        Application.Run(win);
    }
}
```



このプログラムは、コマンドラインで指定したアイコンファイルをウィンドウに設定します  
図を見てわかるように、いつもとはタイトルバーのアイコンが違いますね

---

[前のページへ](#)

[戻る](#)

[次のページへ](#)