



# Iconizer

Created by: Edward Willemsen, 2007

Icon Library by: Castor Tiu

**Table of Contents**

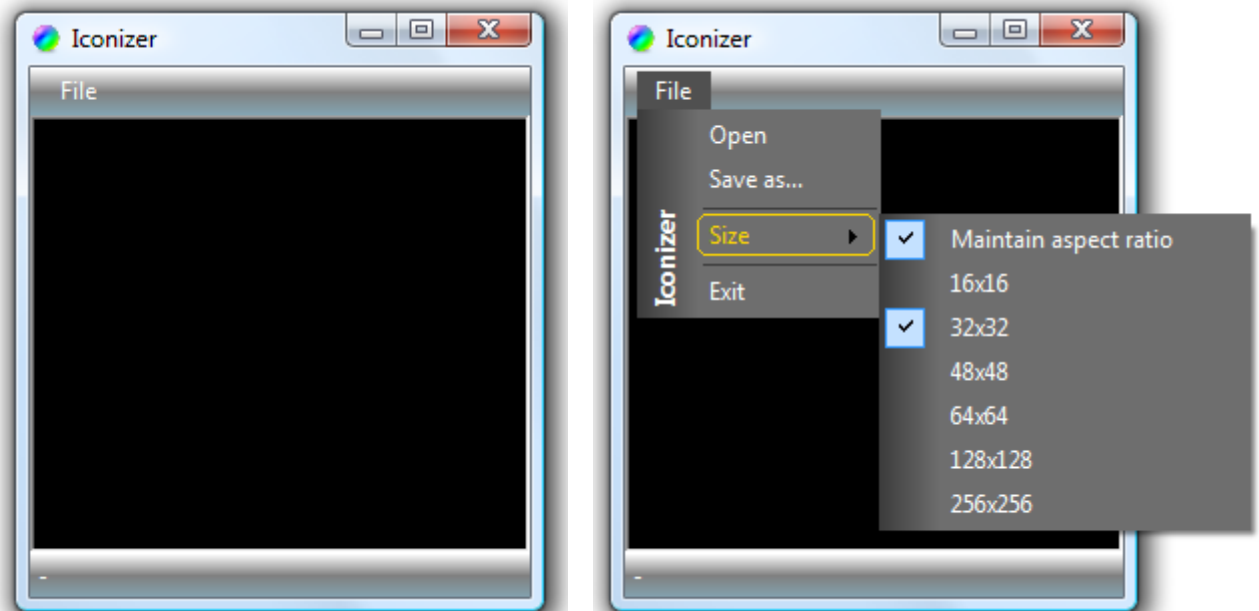
- 1. Introduction..... 3
- 2. Interface ..... 3
  - 2.1. Main Menu ..... 4
    - 2.1.1. Open ..... 4
    - 2.1.2. Save as... ..... 4
    - 2.1.3. Size..... 4
    - 2.1.4. Exit..... 5
  - 2.2. Prerequisites..... 5
- 3. Techno-Talk ..... 5
- 4. License ..... 5
- 5. History ..... 6

## 1. Introduction

Have you ever been in the situation that you created a small application for some colleagues which they probably require for just a few weeks or months? If so, when you were in that situation did you actually give the application identity by including an icon? I've been in that situation more than once and creating, or misusing an existing icon, can still be rather time consuming. Especially for those applications I created a small program which can simply turn a photo into an icon which can be used in your application or as shortcut-icon right away.

## 2. Interface

When the program is started it will look like the image on the left and when exploring the programs main menu, the program will look like the picture on the right:



As you can see, the theme of the application is not according the chosen theme within Windows. The program contains a class called 'FancyToolStrip' which changes the look and feel of the program.

The program remembers its last position on the desktop, so it will show up as where you expect it to show up. Furthermore, settings like 'Maintain aspect ratio' and target icon size will be saved as well.

## 2.1. Main Menu

The items found within the main menu are the following:

- File
  - Open
  - Save as...
  - Size
  - Exit

### 2.1.1. Open

The following photo or image formats can be read by the program:

JPG file(s);

Joint Photographic Experts Group, with .JPG as file extension.

JPEG file(s);

Joint Photographic Experts Group, with .JPEG as file extension.

PNG file(s);

Portable Network Graphics, with .PNG as file extension.

Bitmap file(s);

Windows bitmap, with .BMP as file extension.

GIF file(s);

Graphics Interchange Format, with .GIF as file extension. These images have limited color depth.

When the photo or image is opened, it will be displayed in its image area. You can resize the application and the image content will resize as well.

### 2.1.2. Save as...

Using this option the save dialog will be shown and you can pick-out a name and folder to store the icon. When the icon is saved, the program's icon will be changed and the new created icon is used as temporary program icon.

### 2.1.3. Size

The width and height of a photo is seldom the same. To remain the ratio between the width and the height, called the aspect ratio, check the Maintain aspect ratio option. For a landscape image, the width is larger than the height; this will result in image within the scope of the selected size. For example, a photo with the dimensions 1024x768, using size 32x32 maintaining aspect ratio, will end

up with dimensions 32x24. When you load the photo again and save it within maintaining the aspect ratio, the size will be 32x32, but the content will be changed a little.

#### 2.1.4. Exit

Well, as you might expect, this option will close the program. Keep in mind that any unsaved icon will be lost, so if you want to keep it, save it first!

## 2.2. Prerequisites

This program is written for the Microsoft .NET Framework 2.0.

## 3. Techno-Talk

The program is using the 'GetThumbnailImage' routine provided by the .NET Framework. This routine is fast and creates rather sharp thumbnails. The routine will take the required image size and the image load abort handler as parameters.

```
Image.GetThumbnailImageAbort cb =  
    new Image.GetThumbnailImageAbort(this.ImageLoadAborted);  
  
Bitmap bmp = new  
    Bitmap(src.GetThumbnailImage(iW, iH, null, IntPtr.Zero));
```

Now that we have a bitmap thumbnail of the selected image, we have to put it into the real icon constructor. We create a SingleIcon object called icon and give it a name. Next we add the thumbnailed bitmap into the icon and save it using the input obtained from the SaveFileDialog.

```
SingleIcon icon = new SingleIcon("Iconizer");  
icon.Add(bmp, Color.Transparent);  
icon.Save(dlgSave.FileName);
```

Since we've added an image load abort handler, we'll have to write it as well:

```
private bool ImageLoadAborted()  
{  
    return true;  
}
```

## 4. License

Be aware that the use of IconLib falls under the following license agreement:



<http://creativecommons.org/licenses/by-sa/2.5/>

Other parts of the source code can freely be used or modified in any desired way.

## 5. History

The following table explains the version history of this program.

Version	Date	Comment
<b>V1.0</b>	December 2007	The first release of the program.