



the iconfactory

Quality Freeware Icons for your Mac

A Mac User's Guide to Making Windows Icons

by Craig Hockenberry, Chief Typist, The Iconfactory

Introduction

As a Mac user, you probably haven't had much exposure to the Windows environment. This guide will help you to understand how icons work on the PC platform.

Once you understand how the icons are utilized, you'll be able to create better-looking icons for the Windows platform. As you'll see while reading this guide, it's not easy.

At the end of this document, we'll give you some specific recommendations that will help you build the best possible icon for the platform.

The instructions in this guide assume that you will be using the Iconfactory's IconBuilder Pro software to create your icons. The guide also describes how to use our Windows application, ICONTEST.EXE, for testing your icons on a PC or PC emulator.

Forget What You Already Know

In many ways, the Mac OS and Windows offer very different desktop environments. The way an icon is displayed on the PC differs greatly from the way it is displayed on a Mac.

There is a lot to learn. Much of what you know about the Mac cannot be used to help you understand what's going on with Windows. That's why we're writing this guide.

One of the things that will help you to understand Windows icons is the ICONTEST application. Originally, we wrote this application to figure out what Windows was doing. As we go through this guide, we'll be using ICONTEST to demonstrate some of the "peculiarities" of Windows.

A Mac User's Guide to Making Windows Icons

Page 2

The ICO File

Windows uses a file with the ICO extension to store icon resources. The file typically contains large and small icons with different versions for each color depth.

When programmers build an application, they take these ICO files and put them into the application's resource data. If you are delivering icons to a client who is going to put them into an application, the preferred method of delivery is the ICO format.

Internet users also use ICO files. These files are often put into a ZIP archive that can be downloaded (similar to what Mac users do with folders and Stuffit archives.) After download, the ICO file can be used to customize the user's desktop. Any Windows shortcut (similar to an alias on the Mac) can be customized with an ICO file.

Finally, when you are using the Windows desktop or explorer, an ICO file will automatically display its icon in the list of files. The ICO format is an excellent exchange mechanism between Macs and PCs.

Now we'll talk about what goes into the ICO file.

Sizes

An ICO file can contain icons with the following sizes:

Small: 16 x 16 pixels
Large: 32 x 32 pixels
Huge: 48 x 48 pixels

The small and large icons are common both in user interfaces and in the Windows desktop and explorer.

The Windows desktop also supports huge icons for folders, applications, and documents. This use of huge icons is optional and is set using the Display control panel (see the sidebar on page 5 for more information.)

Many applications and documents do not include huge icons. When Windows displays huge icons, and one is not available in the icon resources, it scales up a large icon. In the process, the icon can become blurry due to the scaling (see Figure 1).

If you are developing an icon for an application, it is a good idea to create the huge version of the icon so that people with poor vision have a fighting chance.



Figure 1. Some huge icons on the desktop. The Internet Options icon has a huge version. The other icons are blocky and blurry because they have been scaled up from either a 16- or 256-color version.

A Mac User's Guide to Making Windows Icons

Page 3

Colors

Each of the icon sizes can contain the following bit depths:

- 1-bit: black and white
- 4-bit: 16 colors
- 8-bit: 256 colors
- 32-bit: millions of colors

All of the color depths use a 1-bit mask to define the transparent and opaque areas of the icon.

These depths are similar to what is found on the Mac. The big differences are in the color tables used for each depth.

Windows XP uses an 8-bit mask on the 32-bit icon and supports various levels of transparency (just like the Mac.) Other versions of Windows use a 1-bit mask.

The question now is what colors can you use to create the icon?

Black & White

Windows supports black and white icons. The problem is that no one ever uses them. If you are creating icons for an application, you can leave them out. No one will notice.

16 colors

16-color icons were originally designed for the CGA (Color Graphics Adapter) standard. This standard used 7 colors at full intensity, 7 colors at half intensity, white and a 75% gray. Figure 2. shows the colors.

To be blunt, these colors suck. They were designed to make the hardware cheap, not to please the eye. The result is a collection of bright, fully-saturated colors and their counterparts, which are dark and hard to read. Also, half of the grayscale colors have special display properties (as a background or when the icon is disabled.)

The worst part is that you can't work around them. We'll talk more about this in the next section.



Figure 2. The 16 colors used for a standard Windows icon. The two colors on the right are special. White pixels are disabled differently than color pixels. The light gray pixel will change color depending on the background color of the window.



A Mac User's Guide to Making Windows Icons

256 colors

The big lie: there is a standard Windows color table for 256 color icons.

The "standard" color table that's used by many applications (including Photoshop and Debabelizer) is a standard color table from version 3.1 of Windows. Very few people use that version of Windows anymore.

With the advent of Internet browsers, the standard colors are no longer used. Why? Because the color table does not include all of the 216 web-safe colors used by the browser.

At the same time, the 216 web-safe colors are not guaranteed to be present. If the user has not started a browser, they won't be defined in the system color table.

The only colors that are guaranteed to be available at any given time are the 16 colors mentioned above. Learn to love them.

So what does that mean for the icon designer who's creating a 256-color icon? Use any icon colors you want. As you'll see in the next section, a 256-color icon can be displayed only when the user has a display set to thousands or millions of colors.

IconBuilder Pro lets you choose the colors you use to build the 8-bit (256 color) version of the icon. By default it uses the Mac system palette, but you can also choose to use the "standard" Windows colors (or a custom color table that you create in Photoshop).

We find that using a Mac system palette for your 256-color icon results in the most visual compatibility between platforms. The Mac palette also has a very even distribution of color, unlike the "standard" Windows colors.

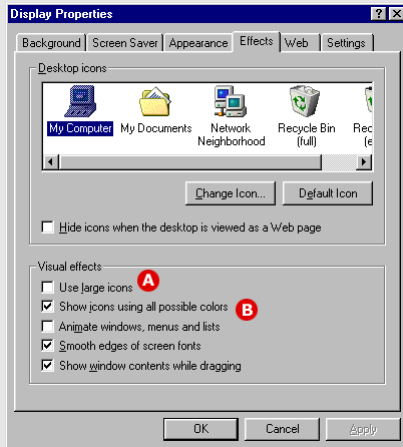
Millions of colors

You can use any color you want, because this version of the icon does not have a color table.

If you are making an icon that you want to display correctly on any color setting, create a 16-color version—and be careful to make the RGB version look similar.



A Mac User's Guide to Making Windows Icons



The Display Properties dialog is where Windows users change their display settings. Several of these settings affect how icons are drawn on the desktop.

To show the Display Properties dialog box, click on the Windows desktop with the right mouse button. Select "Properties" from the context menu.

Here are the main display properties that affect icons:

- A** When this checkbox is set, Windows uses 48 x 48 pixel icons instead of the standard 32 x 32 pixel icons.
- B** Allows a user to see more than 16-color icons.

Both of these options are disabled by default. Many users don't know how to change their display properties, so they will most likely see 32 x 32 pixels at 16 colors.

You can also use the Settings tab of this dialog to adjust your display depth. Most VGA compatible display cards support thousands of colors or more. Some older display cards will require you to reboot after you've changed the display settings.

When you are creating icons for Windows, it is very important to check your work with different display properties.

Icon Display

Windows has a clearly defined set of rules for choosing the icon to display. Unfortunately, these rules are complex and unintuitive. If this section starts to make your head spin, go onto the next section and run the exercises with ICONTEST.

When figuring out which icon to draw, Windows looks in the resource file for an icon that matches the size it needs. If you have only a small icon, and Windows needs the large version, the small icon will be scaled up to fit the requested size (either 32 x 32 pixels or 48 x 48 pixels). If you have only a large icon, it will be scaled down.

If the resource file contains more than one icon of the requested size, Windows will choose the one with the greatest color depth, without exceeding the color depth of the current display settings. When all icons exceed the color depth of the display, the one with the lowest color depth is chosen.

Now for the really strange rule: Windows treats all color depths greater than or equal to 256-colors as equals. If you have a 32-bit icon and a 256-color icon, Windows will choose whichever one it finds first in the resource file. Normally, this is the 256-color icon.

If you are working on a 256-color display, Windows will not use a 256-color icon. Rather, it will choose a 16-color icon. If it cannot find a 16-color icon, it will automatically reduce the colors of any icon it finds to 16 colors (and this conversion process usually makes the icon look really bad).

Windows will also choose the 16-color icon if you are using a display that's capable of displaying millions of colors and you haven't set the option "Show icons using all available colors". In Windows 95, 98, and NT, this option is NOT enabled by default (and since many users don't know this option exists, they probably won't get to see your fancy 256 color icon.) In spite of the high-powered graphics cards in most PCs, most users see a plain old 16-color icon from the days of CGA video adapters.

There are also problems displaying some icons with millions of colors on Windows NT 4.0 (SP 5). The problem occurs when the icon is displayed in thousands of colors. Parts of the icon (usually the mask) are not displayed correctly. In Windows 2000 and XP, this problem has been fixed.

Finally, do not include a 16-color version of a 48 x 48 pixel icon. It will be displayed as a generic icon if you do. It's just one of those 64,000 bugs left in Windows. :-)

A Mac User's Guide to Making Windows Icons

Page 6

Exercises

These exercises are designed to show you how icons are displayed on Windows.

1 Copy ICONTEST.EXE file from your IconBuilder Pro folder onto a PC (or PC emulator). It doesn't matter where you put the file; we find that it's easiest to access from the desktop.

2 Set your display to 256 colors. You do this in the Settings tab of the Display Properties dialog. The fastest way to get there is by clicking on the desktop with the right mouse button. In the menu that is displayed, select "Properties". Then go to the "Settings" tab.

3 Double-click on the ICONTEST icon (the anvil) to start ICONTEST.

4 Click on the telephone icon (#5 in Preset Icon Resources frame) to see a display similar to Figure 3.

Notice the preset icons. Even though your display is set to 256 colors, all the icons are displayed in 16 colors.

5 Now take a look at preset #2. Why does it look so bad? There are several reasons.

The first is that there is no 16-color version of the icon. Windows picked the closest color depth (which was 256-colors as you'll see in the blurry text).

Another reason is that the 256-color icon was reduced to 16 colors automatically. The smooth gradients of red, green, and blue get posterized.

The final reason is that no large version of the icon was created. The huge icon has been scaled down to fit into the 32 x 32 space.

6 Now exit ICONTEST by clicking on the "X" in the upper right-hand corner of the window. Windows applications, unlike Mac applications, do not handle changes to display settings very well. The colors in the icons will often shift when you change your settings.

(continued)

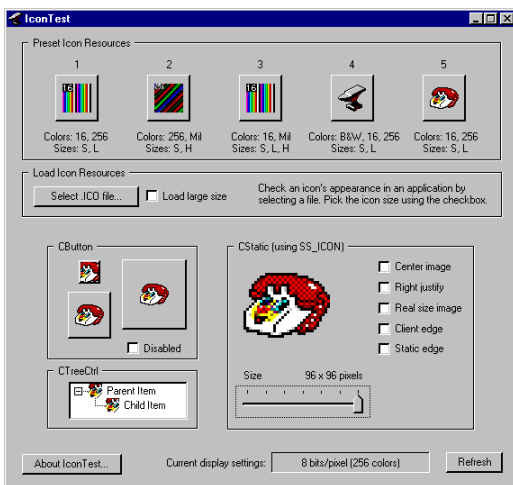


Figure 3. ICONTEST running with display settings at 256 colors.

A Mac User's Guide to Making Windows Icons

Page 7

7 Change your display settings to thousands or millions of colors (High Color or True Color.)

The fastest way is to click on the desktop with the right mouse button. In the menu that is displayed, select "Properties". Then go to the "Settings" tab.

While you're in the Display Properties dialog, go to the "Effects" tab and make sure that "Show icons using all possible colors" is turned off.

8 Restart ICONTEST, by double clicking on its icon, and you should see the same results as before. What's going on? Why don't you see the 256-color icons?

You are seeing the icons without "all possible colors." By default, Windows is setup this way.

9 Exit ICONTEST.

10 Go back to the Display Properties dialog and turn on the "Show icons using all possible colors" option.

11 Restart ICONTEST and you'll finally see your 256-color icons (as shown in Figure 4).

Notice the millions of colors in preset #3 are being displayed correctly. The 256-color icons, with different color tables, are also correct.

12 Preset #2 is still looking a little bit sorry. What's wrong now?

The icon is still getting scaled down from the huge version. That explains the blurry text and some of the sharp edges on the gradients.

You've now had a quick taste of how Windows handles icons. Next, we'll explore how to use ICONTEST to check the icons you'll create with IconBuilder Pro.

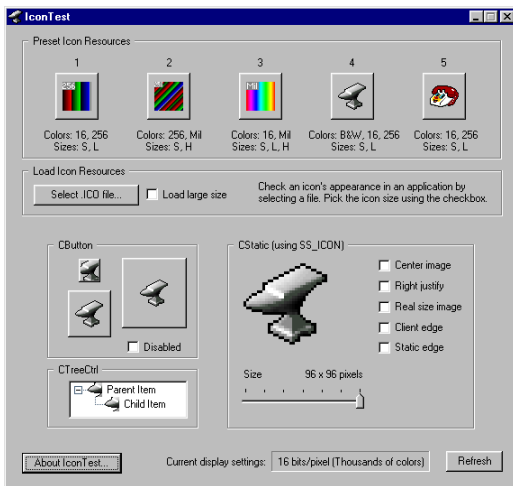


Figure 4. ICONTEST running in thousands of colors.

A Mac User's Guide to Making Windows Icons

Page 8

Testing Your Icon

In this section, we'll show you how to use ICONTEST to test your own icons. We'll look at each of the remaining frames in the interface and we'll describe how to set their various options and checkboxes (see Figure 5.)

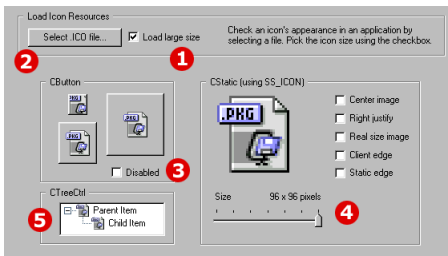


Figure 5. The ICONTEST user interface elements that allow you to test your own icons. The numbers show the steps in the Testing Your Icon section.

1 Select the size of the icon you want to test. Normally, this will be the large size (32 x 32 pixels). If the icons you are creating will be used in a Windows "tree control," use the small size (by clearing the checkbox).

2 Load the ICO file you've created into ICONTEST. Click on the button labeled "Select .ICO file". This will display a dialog box through which you can select the file to test.

3 Click the Disabled checkbox to see how your icon will look when it's disabled. The CButton frame shows how icons are displayed on buttons.

4 The CStatic frame shows how icons are displayed as a picture. You can use the slider to magnify the icon. The checkboxes can also be used to simulate the options a programmer can use; they control how the icon will be displayed in the picture frame.

5 The CTreeCtrl frame shows how your icon would look in a Windows tree control. Normally, you'll want to load the small size from the ICO file (see step 1.) If you don't, Windows will automatically scale down the large icon.

The current display settings are shown at the bottom of the ICONTEST interface. Make sure you also check your icons with your display set to 256 colors and thousands of colors (also called High Color or True Color).

The Refresh button causes all of the items in the user interface to be redisplayed. You won't need to use it while testing your icons.

A Mac User's Guide to Making Windows Icons

Page 9

Toolbars

If you've ever used a Windows application from Microsoft, you've probably encountered a "toolbar." A toolbar contains a palette of icons that you can move around in the application's main window. Sometimes toolbars are called "cool bars" (usually by people who have never used a Mac).

A good example of an application that uses toolbars is Internet Explorer; try grabbing one of the vertical bars on the left-hand side of the toolbar and moving it around. You can also click on the right mouse button to customize the toolbar.

Even though toolbars display iconic images, they are not created with ICO files. Instead, they are created with BMP files. A BMP file is a standard Windows format for bitmap data (in fact, BMP is a shortened version of "bitmap").

When you are creating icons for a toolbar, a client will often require the data to be delivered in BMP format. Photoshop can easily create a BMP file, but what colors should it contain and how are the icons placed within the file?

A toolbar BMP file should contain only 16 colors. Yes, these are the same 16 colors that are used when you are creating icons. In the IconBuilder Pro folder, we've included a color table with these 16 colors (16colors.act).

Standard Icons

Application Icons



Figure 6. A typical toolbar bitmap file with the standard Windows icons (open, save and print) and four icons specific to the application. The small version shown is 16 pixels high and 128 pixels wide (8 places at 16 pixels each.)

As far as the layout is concerned, the icons are placed side-by-side in the file. The height of the bitmap is usually 16 pixels for a small toolbar and 32 pixels for a large toolbar. A simple toolbar bitmap is shown in Figure 6.

When you are creating the file, make sure that the background color has an RGB value of 192, 192, 192. This color is automatically changed to match the window background color.

When the icon is disabled, all colors except white and the background will be used to emboss the icon. You can check the disabled state of the icon in ICONTEST (using the Disable button in the CButton frame).

When you create the toolbar, you won't need IconBuilder Pro because you are not creating ICO files. However, clients like to see work-in-progress and it is much easier to show them desktop icons in the ICO format. When they sign-off on the icons, you can then take the layers you used with IconBuilder and put them into a final BMP file for delivery.

Make sure that you select a depth of "4-bit" when you save the BMP file with Photoshop. If this depth is not available, either you're saving an image that is not indexed color or you haven't used the standard 16 colors for Windows.



A Mac User's Guide to Making Windows Icons

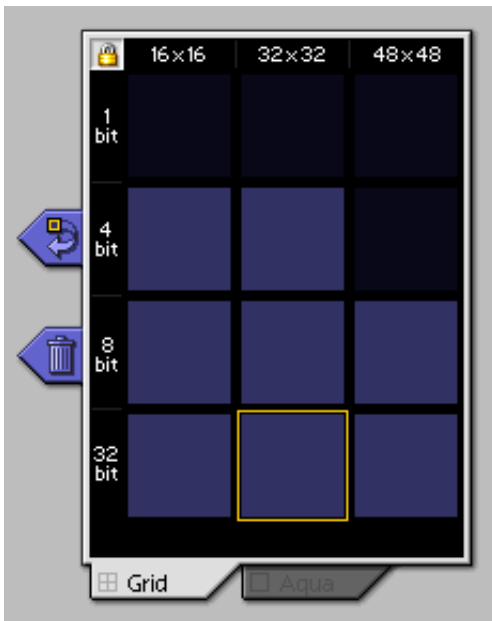


Figure 7. To create an icon that is compatible with all versions of Windows, fill in all the grid cells that are not darkened.

Recommendations

We believe that by following these guidelines, you will produce the best possible icon for the Windows user. Here are our recommendations:

- Always make, and include, a 16-color icon for 16 x 16 and 32 x 32 pixel icons. Make it look as good as you can because this is the version that will be seen by the majority of the people using the application.
- Never include a 16-color version of the 48 x 48 pixel icon. If you do, your icon will be displayed as a generic document. Go figure.
- If you are making icons that will be used on Windows NT, do not include 24-bit icons (at any size). Some colors will be displayed incorrectly. IconBuilder does not support 24-bit icons, but be careful if you use someone else's product.
- Use a Mac system palette if you are making a 256-color icon. This will give you the best cross-platform compatibility.
- Don't bother with a black & white version of the icon. No one will ever see it.
- Test your icon with ICONTEST using display settings at 256 colors and thousands of colors. You and your client will be happy you did.
- Use IconBuilder Pro to create your icons from within Photoshop. Creating different versions of the icons using layers makes your job much easier.
- IconBuilder Pro also has a QuickConvert feature that automatically creates a Windows icon from any Mac icon. Cross-platform development has never been easier.

© 1999-2001 The Iconfactory. All rights reserved. The Iconfactory and Iconfactory logo are registered trademarks. All other product names mentioned are used for identification purposes only and may be trademarks or registered trademarks of their respective holders.

