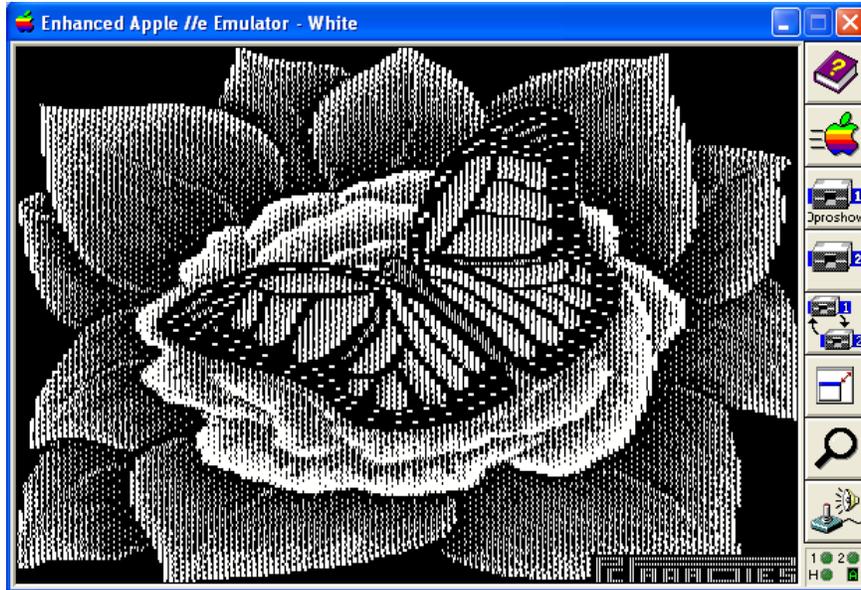


**Capturing and Recycling AppleWin Double Hi-Res (DHGR) Screens**  
A Tutorial by Bill Buckels, March 2013

**Summary - Capturing an AppleWin DHGR Screen**



The Apple II DHGR image above is displayed in [AppleWin](#)'s Monochrome White Screen Mode with "50% Scan lines" turned-off!

Capturing 560 x 384 screens (by pressing the [Print Screen] key) while displaying a DHGR image in AppleWin's Monochrome White Screen Mode, provides a "true" black and white bitmap of color DHGR screen memory that can be perfectly and accurately converted to a Windows Color BMP file (and Apple II DHGR binary files), using a program like my Apple II Full Color BMP (A2FCBMP) Conversion Utility, without "dragging along" AppleWin's "special color display effects".



Monochrome BMP converted to color by A2FCBMP

## **Table of Contents**

<a href="#">Capturing and Recycling AppleWin Double Hi-Res (DHGR) Screens.....</a>	<a href="#">1</a>
<a href="#">Summary - Capturing an AppleWin DHGR Screen.....</a>	<a href="#">1</a>
<a href="#">Table of Contents.....</a>	<a href="#">2</a>
<a href="#">Introduction.....</a>	<a href="#">3</a>
<a href="#">Footnote: Some Applications for Managing Apple II ProDOS Disk Images .....</a>	<a href="#">4</a>
<a href="#">Getting Started – Capturing Apple II 16-Color DHGR Screens.....</a>	<a href="#">4</a>
<a href="#">How to Tell if a DHGR Screen is Currently Displayed in AppleWin.....</a>	<a href="#">4</a>
<a href="#">AppleWin Screen Mode - Monochrome (White).....</a>	<a href="#">5</a>
<a href="#">Selecting a Folder (Directory) for AppleWin Captures.....</a>	<a href="#">5</a>
<a href="#">Keyboard Capturing Keys.....</a>	<a href="#">6</a>
<a href="#">The Pause Key – Stop or Restart a Running Apple II Program.....</a>	<a href="#">6</a>
<a href="#">PrintScreen – Capture a 560 x 384 SVGA BMP .....</a>	<a href="#">6</a>
<a href="#">File Names of AppleWin Screen Captures.....</a>	<a href="#">7</a>
<a href="#">Use Short File Names.....</a>	<a href="#">7</a>
<a href="#">Getting Started – Editing Converted Apple II DHGR Screens.....</a>	<a href="#">7</a>
<a href="#">A2FCBMP Specifics.....</a>	<a href="#">7</a>
<a href="#">AppleWin’s DHGR Monochrome White Screen Captures.....</a>	<a href="#">8</a>
<a href="#">AppleWin Changes between Versions .....</a>	<a href="#">8</a>
<a href="#">A2FCBMP’s DHGR Colors and Editing .....</a>	<a href="#">8</a>
<a href="#">Edit in Pixel Pairs.....</a>	<a href="#">9</a>
<a href="#">Consider Your Editing Layout.....</a>	<a href="#">9</a>
<a href="#">Preparing Text and Line Graphics for Discrete Color Conversion.....</a>	<a href="#">10</a>
<a href="#">Masking and Layering.....</a>	<a href="#">10</a>
<a href="#">Avoiding Anti-Aliasing of Text and Primitives.....</a>	<a href="#">10</a>
<a href="#">How to Create a “Masked Layer” in Windows Paint.....</a>	<a href="#">10</a>
<a href="#">The Down-Side of Avoiding Anti-Aliasing .....</a>	<a href="#">11</a>
<a href="#">Best Practices for Editing.....</a>	<a href="#">11</a>
<a href="#">Colors in an A2FCBMP and BMPA2FC “DHGR BMP” Image.....</a>	<a href="#">11</a>
<a href="#">Table of A2FCBMP and BMPA2FC “DHGR BMP” RGB Color Values.....</a>	<a href="#">12</a>
<a href="#">Getting Started - Converting Apple II “DHGR BMPs” to Apple II Format.....</a>	<a href="#">13</a>
<a href="#">Starting A2FCBMP, and BMPA2FC.....</a>	<a href="#">15</a>
<a href="#">Licence Agreement.....</a>	<a href="#">15</a>
<a href="#">Closing Remarks.....</a>	<a href="#">16</a>
<a href="#">Appendix A - Apple II DHGR Binary Patterns.....</a>	<a href="#">18</a>

**Disclaimer:** See the [Licence Agreement](#) in this tutorial for more details.

Note: The DHGR images in this tutorial document were produced either by converting AppleWin DHGR Monochrome Screen Captures, or converted from BMP files of IBM-PC images saved and edited in Windows Paint, or a by a combination of both methods, using the utilities A2FCBMP and BMPA2FC discussed in the pages that follow.

## **Introduction**

1. The [AppleWin Emulator](#) can be used to capture monochrome white mode DHGR screens from a running Apple II program and save to a Windows BMP file. AppleWin has “hot keys” built-in to do this.
2. These monochrome mode screen captures can then be converted to Color Screens with Precise Pixel Locations using A2FCBMP, and the converted BMPs edited in Windows Paint and “retouched”. (Or they can just be left alone “as-is”.)



3. After editing, converted AppleWin Screen Captured bmp's can be converted back to Apple II Graphics in a variety of formats, including Apple II BSaved images (BIN AUX file pairs), and A2FC (uncompressed Apple II Full-Color files).
4. These “recycled” graphics can be placed back on an Apple II disk image again using utilities like [CiderPress](#), and subsequently used in your own programs or slide-shows (or for whatever purpose you see fit).

This document is a brief tutorial about capturing, editing, and otherwise recycling Apple II DHGR (Double Hi-Res) screens using AppleWin and Windows Paint, and 2 utilities that I have written for the occasion: A2FCBMP and BMPA2FC.

These programs (briefly described in this document) have their own documentation. I also provide commented source code with my utilities. Please consult the documentation for AppleWin, Windows Paint, CiderPress, and the documentation and source code for my own utilities for additional information not covered in this tutorial.

This tutorial will also provide additional information to help you on your way. Although this tutorial's primary focus is on the use of the AppleWin emulator as a DHGR graphics source, and Windows Paint as a graphics editor, and CiderPress as a disk manager, much of the general information it provides can be applied elsewhere and expanded upon.

So without further ado, let's put-on our Apple II "eye-patches" and get started.

### **Footnote: Some Applications for Managing Apple II ProDOS Disk Images**

- [CiderPress](#) - No other Apple II Disk Manager has as many features or supports as many formats. Actively developed and supports CF (compact flash) storage etc.
- [Apple II Oasis Disk Manager](#) – Has "Batch Commands" and great support for DOS 3.3 disk images as well as ProDOS, but not actively developed.
- [AppleCommander](#) – Cross-Platform Support, written in Java and actively developed. Targeted towards Apple II Developers with support for the cc65 compiler, but can be used by casual Apple II enthusiasts as well.

### **Getting Started – Capturing Apple II 16-Color DHGR Screens**

The AppleWin Application comes with a Windows Help File. Please consult AppleWin Help for attributions and AppleWin information not included here.

### **How to Tell if a DHGR Screen is Currently Displayed in AppleWin**

1. Pause AppleWin by Pressing the Pause Key
2. Open the AppleWin Debugger by Clicking On the Magnifying Glass Icon
3. The Bottom Right side of the screen displays the Apple II Graphics Mode
4. Close the AppleWin Debugger by Clicking On the Magnifying Glass Icon
5. Press the Pause Key Again to Resume AppleWin

If the current screen is Hi-Res (HGR) and \*NOT\* Double Hi-Res DHGR use one of my other utilities to convert it after reading the documentation below:

- [Capturing and Recycling HGR Screens](#)
- [BIN2RAG](#)
- [HMONSTER](#)
- [BMP2RAG](#)

A Hi-Res (HGR) AppleWin Screen Capture \*CAN'T\* be converted using the methods described in this tutorial, nor can a Lo-Res or Double Lo-Res AppleWin Screen Capture.

### AppleWin Screen Mode - Monochrome (White)



To Capture a DHGR Color Screen for conversion, select Monochrome (White) from AppleWin's configuration dialog which is accessed through the joystick-speaker icon on AppleWin's toolbar. **Do not select Color!** Turn-off 50% Scan lines!

Shortcut Function Keys F9 and Shift+F9 will also cycle through AppleWin's display modes in forward and reverse order respectively.

### Selecting a Folder (Directory) for AppleWin Captures

- Use the AppleWin Toolbar to select a disk image for drive 1 from the directory you wish to save to. Clicking-on a disk image (File Associations) will \*NOT\* work to select your AppleWin Capture Directory!
- After you select your AppleWin Capture Directory do not use the AppleWin Toolbar to load disk images for capture unless they are in the capture directory. Click-on disk images (Use File Associations) to load subsequent disk images from other directories. That way, "screen-grabs" from subsequent disk images will also be saved in the AppleWin capture directory that you previously selected.
- Keep Windows Explorer Open in your capture directory at all times. That way you will always see AppleWin's captured BMP's at the end of your file-list. If you need to "boot-around" to other directories to "click-on" (load) subsequent AppleWin disk-images containing DHGR screens you want to capture, use another copy of Windows Explorer to "boot-around".

## Keyboard Capturing Keys

### **The Pause Key – Stop or Restart a Running Apple II Program**

Use the Pause Key to Pause/Resume the Apple II Program you are capturing screens from. That way, you can pause a program to “grab” the title screen, or in fact any screen, then resume (by pressing Pause again) to capture the next screen.

### **PrintScreen – Capture a 560 x 384 SVGA BMP**

Always capture to this size for a DHGR image to be converted. Always capture DHGR screens in AppleWin’s Monochrome White Screen Mode, and always capture any screen to be converted using my utilities with 50% Scan lines \*OFF\*!

1. Make sure you have Windows Explorer open in your capture directory.
2. Pause AppleWin.
3. Press and release the Print Screen Key.
4. Wait for AppleWin to finish saving. Before you resume AppleWin, look at the bottom of the file-list to make sure you got a “grab” and a new bmp exists.
5. Verify - Click-on the new bmp file that AppleWin created to make sure you got what you wanted before resuming AppleWin.
6. Resume AppleWin.



## **File Names of AppleWin Screen Captures**

When you follow the steps above, AppleWin saves screen captures to bmp files using automatic naming convention. Bmp file names begin with the disk image name followed by a sequential number. For example, captures from a disk image called “Winter\_Games.dsk”, will result in bmp files named “Winter\_Games\_000000000.bmp, Winter\_Games\_000000001.bmp, Winter\_Games\_000000002.bmp, etc”.

### **Use Short File Names**

Rename the bmps that you wish to convert to an MS-DOS 8.3 naming convention. For example, “Winter1.bmp, Winter2.bmp, Winter3.bmp, etc.” are OK. “Winter\_Games.bmp” is too long a name, because “Winter\_Games” is more than 8 characters in length.

My MS-DOS/Windows utilities **\*DO NOT\*** support long filenames, and they use the bmp file’s “base name” to create Apple II files from bmp files. For example, “Winter1.bmp” will become “WINTER1.BIN”, etc.

These MS-DOS/Windows utilities are written for distribution with the Aztec C65 MS-DOS cross-compiler for the Apple II. The Aztec C65 Apple II compiler will also run on other platforms like Linux (or Windows 7) in an MS-DOS emulator (like DOSBox), so I have created these utilities as MS-DOS programs to work outside of versions of Windows (like Windows XP) that do not necessarily need an emulator like DOSBox .

To make a long story short, make your long filenames short!

## **Getting Started – Editing Converted Apple II DHGR Screens**

**You can skip editing if you don’t need to make any changes. A2FCMP directly creates Apple II Native Files (BIN AUX file pairs and A2FC files) when it converts from a captured Black and White AppleWin DHGR screen.**

### **A2FCBMP Specifics**

You need to know a little about the way AppleWin represents the HGR Monochrome Screen in its captured BMP’s and a little about the 24 bit BMP’s that A2FCBMP converts these to before you go “hog-wild” in Windows Paint and make changes that cannot be converted back to Apple II DHGR Graphics files.

You also need to know a little about the way the DHGR screen works. The program source code and the documentation for A2FCBMP and BMPA2FC may also be of some help.

## **AppleWin's DHGR Monochrome White Screen Captures**

DHGR colors on an Apple II computer are unlike HGR; every single “bit” of a DHGR color is visible on the monochrome display. Therefore it is a very simple matter for a conversion utility like A2FCBMP to “reconstruct” the “true” color of an AppleWin Monochrome White DHGR screen capture. See [Appendix A](#) for DHGR Binary Color Patterns.

Unlike HGR images which cannot be accurately converted from monochrome because they use the “invisible hi-bit” in a monochrome display to set “Green-Violet” or “Orange-Blue”, DHGR images do not use the “invisible hi-bit”. Because the entire “bit-mask” is visible, the exact DHGR color “bit pattern” is available to converters like A2FCBMP, so correct color mapping occurs, provided of course that AppleWin does not decide to sacrifice its “spatial integrity” altogether in pursuit of the notion of simulated analog in some future version.

## **AppleWin Changes between Versions**

For the purposes of this document, and testing the utilities discussed herein, I used AppleWin versions 1.16 and 1.22

When I wrote A2FCBMP, if I had tried to convert based on AppleWin's Color Rendering, conversion results would have been inaccurate, because AppleWin “embellishes” the DHGR display to please its notion of trying to imitate the “effects” of a real Apple II, and “munches” the “true colors” of the digital image as a result. I would also have needed to remain “in-synch” with DHGR color changes between AppleWin versions. With a monochrome DHGR “bit mask” I do not have this problem, so consequently you will get accurate DHGR conversion results when you follow the steps in this tutorial.

Also you can always temporarily install one of the above noted older versions of AppleWin, as a “work around” to do monochrome DHGR screen captures if discrete DHGR monochrome bitmasking “breaks” in future versions

## **A2FCBMP's DHGR Colors and Editing**

There are 16 – Colors on the Apple II DHGR screen. Each one has its own 4-bit value which is spread alternately across Auxiliary and Main Memory (Banks 1 and 0) in increments of 7 bits; 1 byte of 7 bits in Bank 1, 1 byte of 7 bits in Bank 0, etc. starting from the left edge of the DHGR screen. Each horizontal scan-line is 80 bytes, with 40 bytes in Bank 0 alternating with 40 bytes in Bank 1

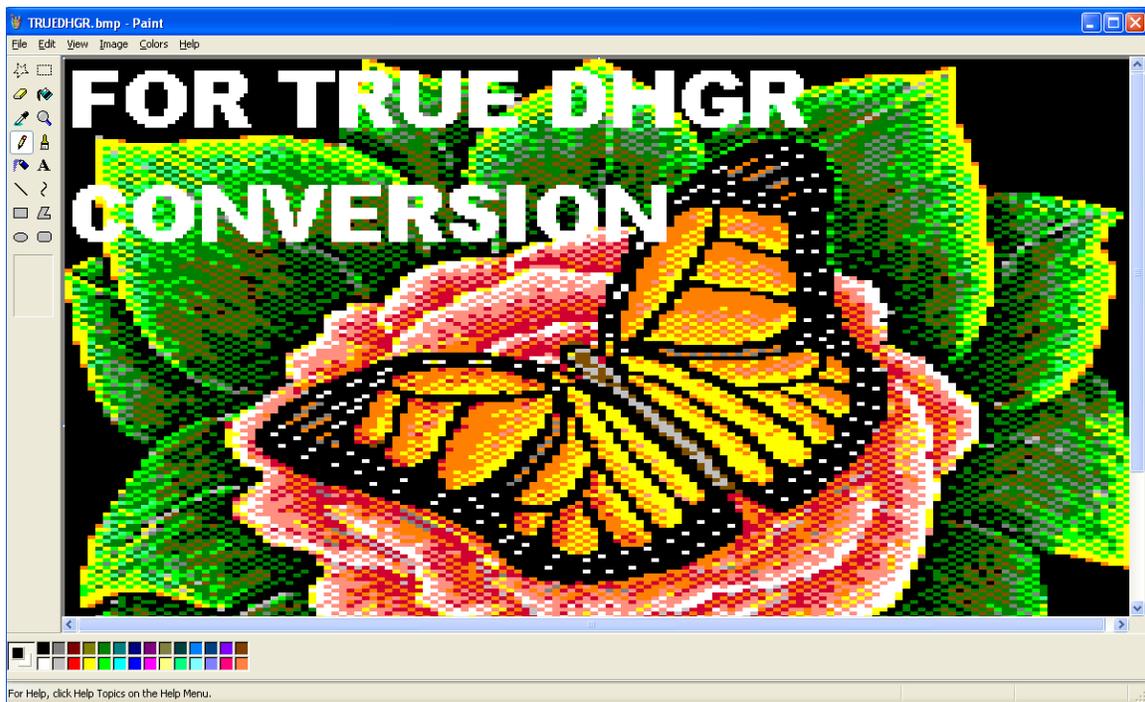
This equals 140 pixels of color interleaved with 7 pixels in each 4 byte block interleaved across Bank 1 and Bank 0 in DHGR screen memory.

A converted BMP resolution of 140 x 384 would have looked “really stupidly skinny” so when A2FCBMP converts DHGR, it doubles each pixel to preserve the horizontal “aspect ratio”.

A converted BMP resolution of 280 x 384 would have looked “stupidly skinny”, and AppleWin doubles the 192 scan-lines on the Apple II display, so it is “safe” for A2FCBMP to “throw away” every “second line”, to provide a 280 x 192 BMP that is “color-position-precise” and does not look “skinny”.

### **Edit in Pixel Pairs**

For the reasons noted above, edit in “pixel pairs” if you intend to convert back to Apple II DHGR using BMPA2FC because “on the way back” to the Apple II, every second pixel will be “tossed”!



### **Consider Your Editing Layout**

When you edit a “DHGR bmp” for Conversion to Apple II DHGR you need to consider the positioning of your pixels.

You also need to consider your colors. Only the 16 colors that A2FCBMP and BMPA2FC use to convert DHGR images to BMPs can be guaranteed to convert properly back to the Apple II when you use these utilities.

## **Preparing Text and Line Graphics for Discrete Color Conversion**

### **Masking and Layering**

The concept of “masking” and “layering” is used here. Windows Paint supports a transparent paste, a flood fill, a color eraser, and an “eye-dropper” tool, and a color viewing and adjustment dialog. These are all essential features that you should be familiar with, for proper editing of any bitmapped graphics. You can consult Windows Paint Help for more information about how to use these.

### **Avoiding Anti-Aliasing of Text and Primitives**

When you create text or a line in a 24 bit image Windows paint will “smooth” (anti-alias) the edges of curved or diagonal line. This creates convoluted extra colors that will not convert properly back to the 16 colors you are allowed by A2FCBMP or BMPA2FC.

### **How to Create a “Masked Layer” in Windows Paint**

An old concept that I borrowed from my days of titling analog video and using digital overlays over a composite signal is the idea of a bitmask. Without getting too far into how masking is used to “toast” an analog television display with digital graphics, cvbs “pass-through,” and so forth, the idea is still the same; do all your text and line graphics in some kind of monochrome... do not do them in 24 bits...

1. You need to create a 2 color mask to transparently paste “primitives” over your 24 bit image in a color that does not map to one of the 16 DHGR BMP color values supported by A2FCBMP and BMPA2FC for 24-bit BMPs.
2. After you have created your black lines or text on a white background in a 280 x 192 monochrome bmp (keeping in mind the 2 pixel width rule), save your monochrome bmp as a 16 color bmp, and flood fill your black lines with a bright color you can see like Magenta (basically “shocking pink”).
3. Then copy the resulting “Magenta Masked” text or line elements to the Windows ClipBoard and transparently paste over top of your “candidate” 24-bit “DHGR BMP”.
4. As a final step, set your supported “target” color for the text or lines that you have pasted, using the eye-dropper if the color you want exists on the image, or set the rgb value directly in the color dialog. See below for the rgb values for this purpose. Use the flood-fill, color eraser, or pixel tool to “re-colorate” your “Magenta Mask” to one of the 16 supported “DHGR BMP target colors”.

Text Tip – Use a thin font for text. Afterwards using the Windows Paint Clipping Box surround your text, and skew the horizontal aspect 200%.to double your pixels. This works every time! Your text will always be legible.

## **The Down-Side of Avoiding Anti-Aliasing**

The “down-side” of creating aliased line elements is “jaggies”, but unless you are lucky enough to convolve your transition filter between the 16 DHGR colors evenly, the effects of doing so will appear horrendously ugly, malformed, distorted, and otherwise uneven on the less-capable Apple II display. Consider that a simple rgb convolution of 4 different intersecting pixels “triple-cubes” each your rgb gun values, and generally creates a “brand new” color, which generally reduces poorly to the “nearest color” in only 4 bits of color depth. Such techniques give the illusion of “analog” detail in more capable displays, and can be combined with light source (grayscale) masks to make a darned nice job that is less “expensive” than ray-tracing in real-time, but on the DHGR these techniques will usually (if not always) make a mess. I apologize profusely for ending this discussion here, and opting for a simplistic overview without providing a lot of hard facts, but you can do the math yourself.

## **Best Practices for Editing**

I realize this is quite a lot of information if you are hearing it for the first time so here are some best practices to follow:

- Work in pixel pairs if you can.
- Work in the 16 colors provided for you.
- Use a “Magenta Mask” to preserve discrete colors (see above).
- Don’t expect perfect results the first time.

## **Colors in an A2FCBMP and BMPA2FC “DHGR BMP” Image**

You cannot just use any color you please and expect it to map properly to the Apple II. In a “DHGR BMP” Image. You must do any editing with the 16 colors supported by A2FCBMP and BMPA2FC if you want to properly convert to the Apple II after editing.

In Windows Paint, you use the “eyedropper” tool to select an existing color, or set the “DHGR BMP” colors with “DHGR BMP” RGB (Red, Green, Blue) values in the “colors” drop-down menu in Windows XP Paint, or the “edit colors” drop-down menu in Windows 7 Paint. The RGB values can be reviewed, and “DHGR BMP” colors can be defined from there.

The table below gives the required RGB values and the Apple II mapping of the 16 “DHGR BMP” colors.

### Table of A2FCBMP and BMPA2FC “DHGR BMP” RGB Color Values

Note: RGB Values (in “LO Res” Index Order) in decimal (required by Paint):

Red	Green	Blue	Apple II Color
0	0	0	black
208	0	48	red
0	0	128	dkblue
255	0	255	purple
0	128	0	dkgreen
128	128	128	gray
0	0	255	medblue
96	160	255	ltblue
128	80	0	brown
255	128	0	orange
192	192	192	grey
255	144	128	pink
0	255	0	ltgreen
255	255	0	yellow
64	255	144	aqua
255	255	255	white

This tutorial is not meant to teach you how to use Windows Paint to edit bitmapped images. The Windows Paint help file and practice and patience will do that, and this is enough about Windows Paint for now!



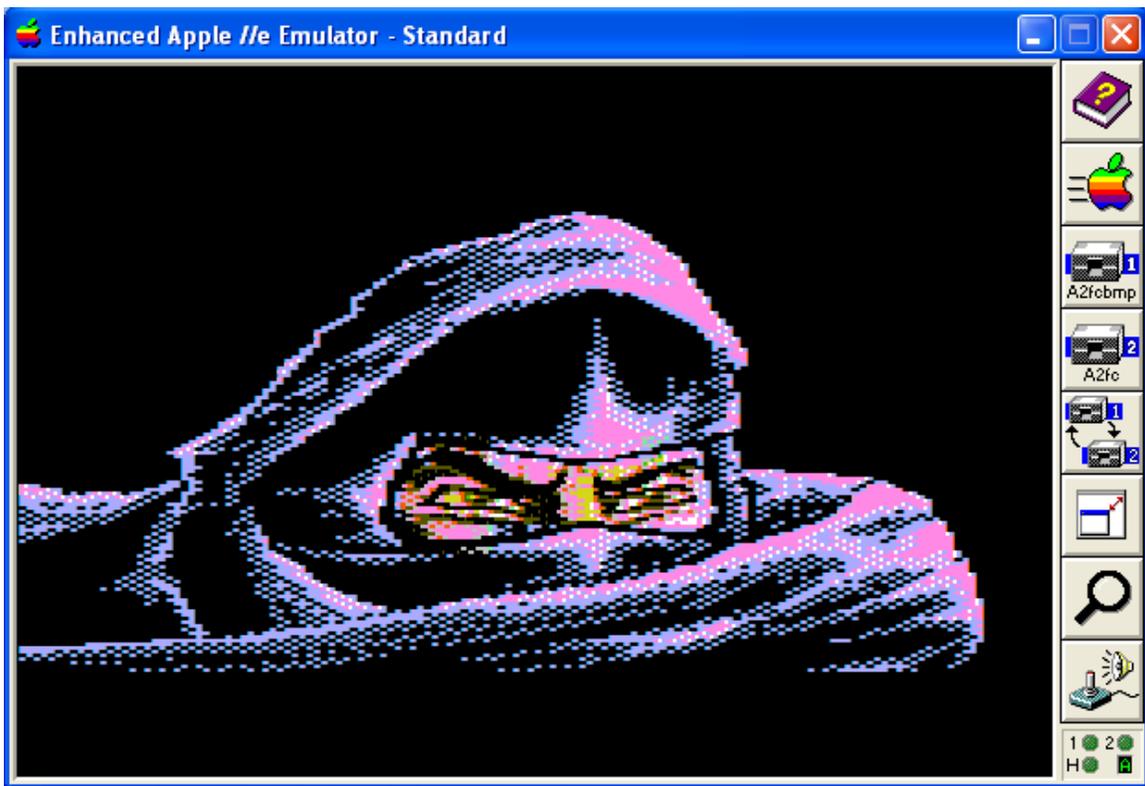
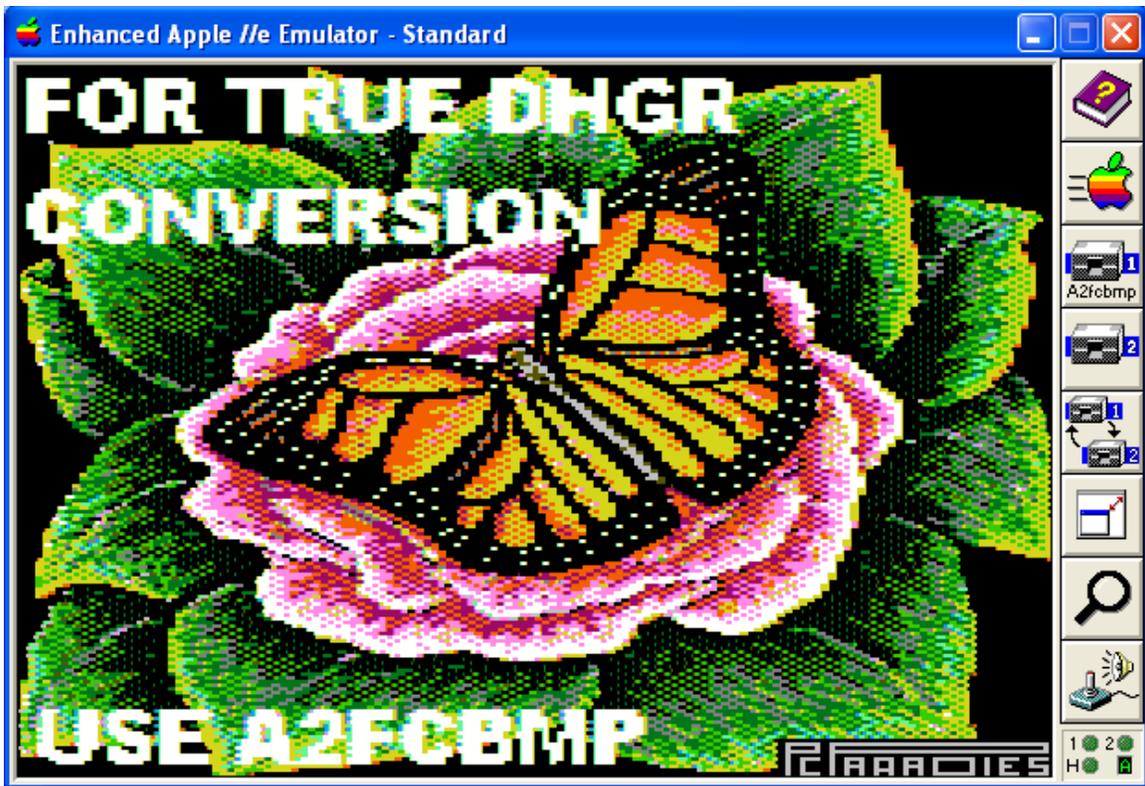
## Getting Started - Converting Apple II “DHGR BMPs” to Apple II Format



The screen above “says it all”. When your DHGR bmp is ready to be converted into an Apple II DHGR file, from MS-DOS (or DosBOX), or from a Windows cmd Window, with Bmpa2fc “on path”, simply type “Bmpa2fc” followed by the bmp filename.



After conversion, place the converted DHGR file on an Apple II disk (or whatever your intention is).



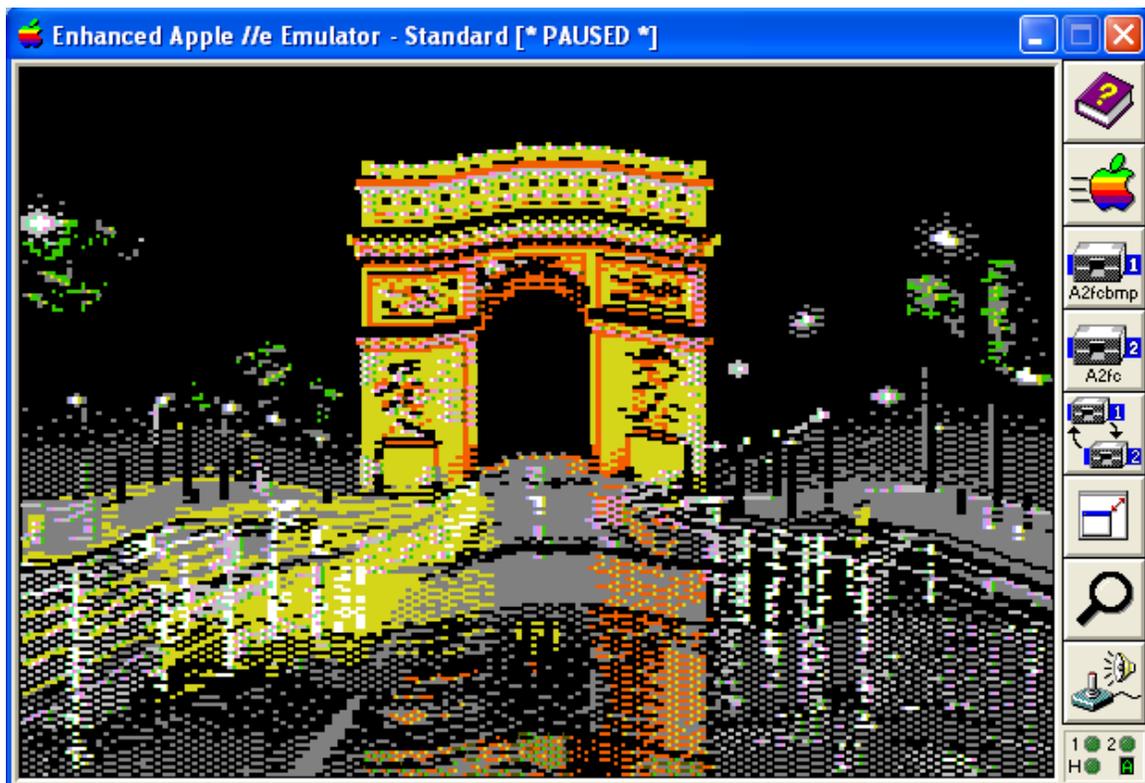
## **Starting A2FCBMP, and BMPA2FC**

Both A2FCBMP.exe and BMPA2FC.exe work in basically the same manner when launched from the command line; enter either A2FCBMP or BMPA2FC, followed by the name of the file to be converted.

You can also launch either program by clicking-on the program in Windows Explorer or from the “DOS prompt” without a bmp filename, and you will be prompted to enter the filename.

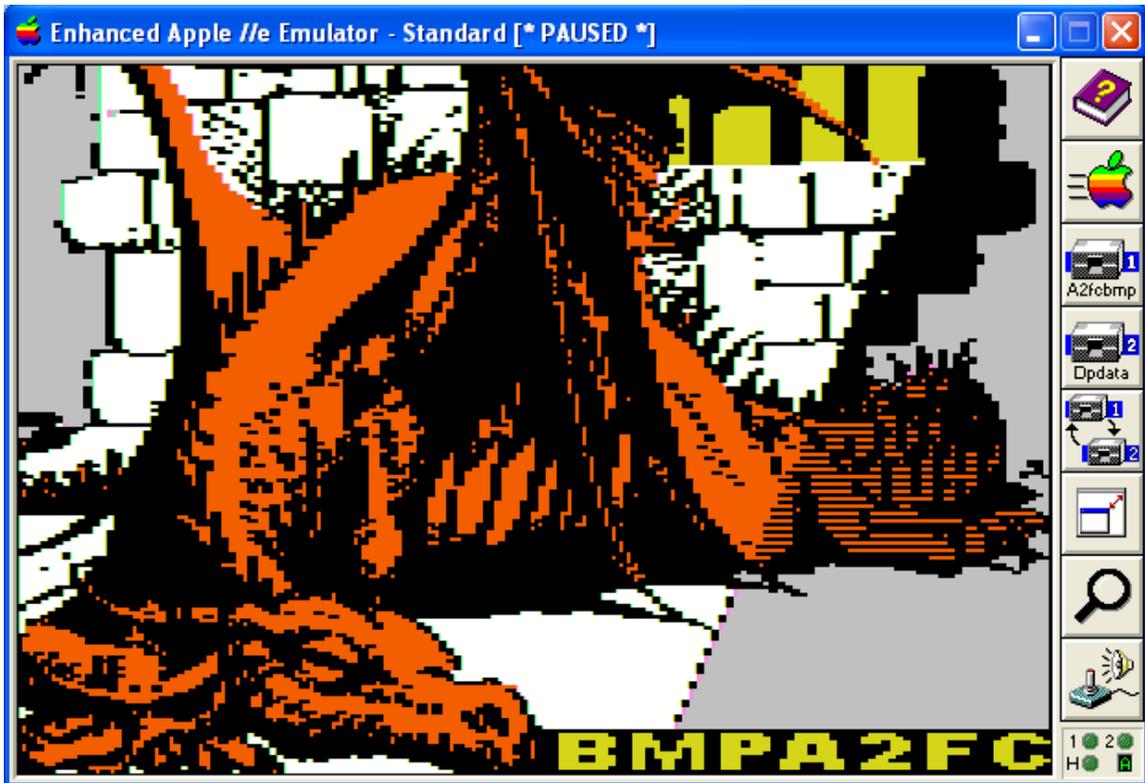
Again, read the individual documentation, source code, and use A2FCBMP, and BMPA2FC to see how all this works for you.

This ends the DHGR “Getting Started” tutorial.



## **Licence Agreement**

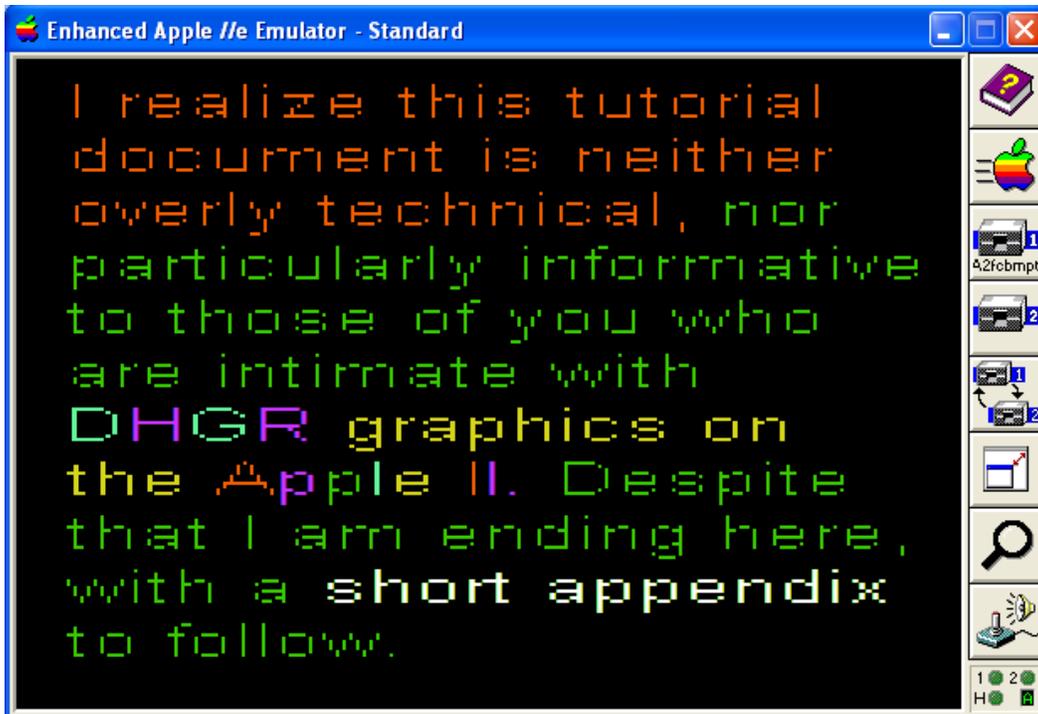
You have a royalty-free right to use, modify, reproduce and distribute A2FCBMP and BMPA2FC, related source code, and related demos and the other stuff they come with (“baggage”) including this tutorial document. You may use these programs and their “baggage” for whatever you wish, provided that you agree that I have no warranty obligations or liability resulting from said distribution in any way whatsoever. This seems fair enough since the rest of the planet can do exactly the same thing.



### Closing Remarks

I hope those of you who have a use for converting from Windows BMP files to Apple II DHGR graphics files will explore this topic further and with enthusiasm, and will add BMPA2FC, and A2FCBMP to your toolbox. For those brave souls who wish to pursue all this further from the C Programming Side, at time of this writing I am preparing the 2013 release of the Aztec C65 AppleX Apple II ProDOS cross-compiler “bundle” for MS-DOS/Windows, which is also full of utilities, sample programs, and other goodies like these, and which will include A2FCBMP, BMPA2FC, and much more!

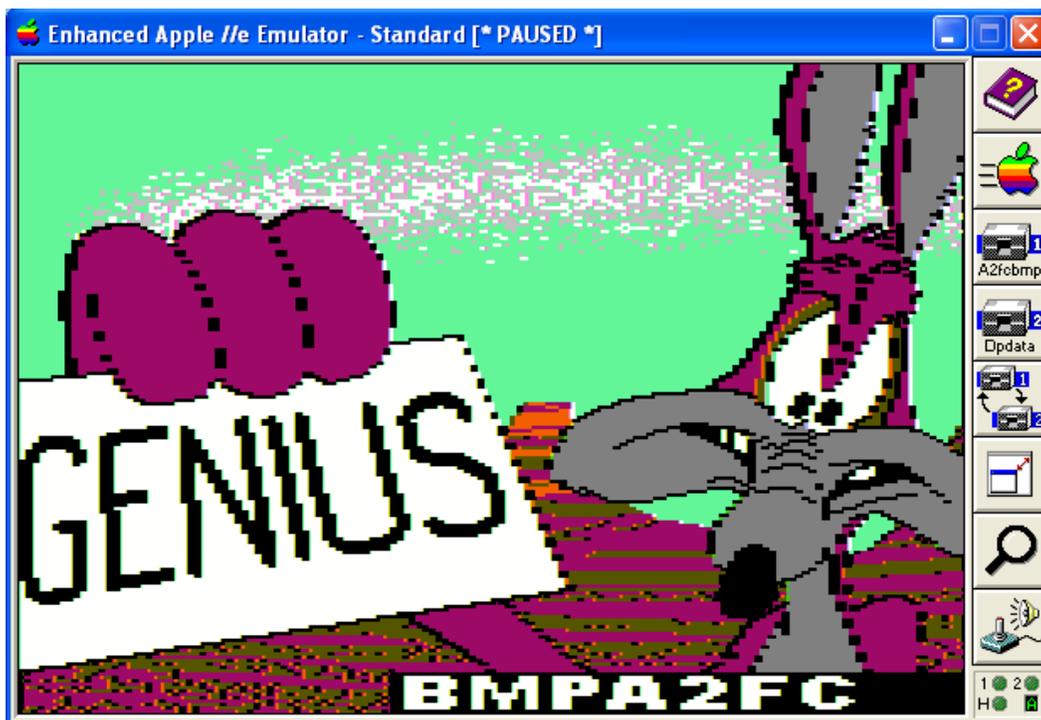
The [AppleX](http://www.aztec.com/applex/) 2013 Update will soon be available from the Aztec C Website’s compilers page at <http://www.aztec.com/applex/>



Have Fun! And All the Best!

Bill Buckels, March 24, 2013

[bbuckels@mts.net](mailto:bbuckels@mts.net)



## Appendix A - Apple II DHGR Binary Patterns

The following is logically reordered to match the Apple II lores color order...

Color	aux1	main1	aux2	main2	Repeated Binary Pattern
Black	00	00	00	00	0000
Magenta	08	11	22	44	0001
DarkBlue	11	22	44	08	1000
Violet	19	33	66	4C	1001
DarkGreen	22	44	08	11	0100
Grey1	2A	55	2A	55	0101
MediumBlue	33	66	4C	19	1100
LightBlue	3B	77	6E	5D	1101
Brown	44	08	11	22	0010
Orange	4C	19	33	66	0011
Grey2	55	2A	55	2A	1010
Pink	5D	3B	77	6E	1011
Green	66	4C	19	33	0110
Yellow	6E	5D	3B	77	0111
Aqua	77	6E	5D	3B	1110
White	7F	7F	7F	7F	1111

