

Capturing and Recycling AppleWin HGR “Screen-Grabs” – A Tutorial
By Bill Buckels, March 2013



The Apple II HGR image above was originally an Apple II game screen. It was produced from the AppleWin Screen Capture below using the techniques in this tutorial.



Table of Contents

Capturing and Recycling AppleWin HGR “Screen-Grabs” – A Tutorial.....	1
Table of Contents.....	2
Introduction.....	3
Footnote: Some Applications for Managing Apple II ProDOS Disk Images	3
Getting Started – Capturing Apple II 6-Color HGR Screens.....	4
AppleWin Screen Mode - Color (standard).....	4
Selecting a Folder (Directory) for AppleWin Captures.....	4
Keyboard Capturing Keys.....	4
The Pause Key – Stop or Restart a Running Apple II Program.....	4
Shift+PrintScreen – Capture a 280 x 192 SVGA BMP	4
PrintScreen – Capture a 560 x 384 SVGA BMP	5
File Names of AppleWin Screen Captures.....	5
Use Short File Names.....	5
Getting Started – Editing Captured Apple II HGR Screens.....	6
AppleWin Captured BMP Specifics.....	6
AppleWin’s HGR Colors and Editing	7
AppleWin Changes between Versions	7
Consider Your Editing Layout.....	7
Best Practices for Editing.....	7
Colors in an AppleWin Captured HGR Image.....	8
Table of AppleWin HGR RGB Color Values.....	8
Getting Started – Converting Captured Apple II HGR Screens to Apple II Format.....	9
About HMONSTER and BIN2RAG, and BMP2RAG.....	9
Licence Agreement.....	10
HMONSTER – A “Round Trip” Utility.....	10
BMP2RAG – a “One-Way” Ticket	11
BIN2RAG – HGR Graphics Converter and Clipper for ProDOS 8.....	12
Closing Remarks.....	15
Appendix A - Apple II HGR Byte Pair Diagrams.....	16
Figure 1 - Black - Palette 0.....	16
Figure 2 - White - Palette 1.....	16
Figure 3 - Green - Palette 0.....	16
Figure 4 - Violet - Palette 0.....	17
Figure 5 - Orange - Palette 1.....	17
Figure 6 - Blue - Palette 1.....	17

Disclaimer and Curse: I am not responsible for anything. This cavalier and arrogant lack of responsibility extends to the AppleWin Emulator, Ciderpress, Windows Paint, none of which I had anything to do with, and just about anything else you are about to read or attempt as a result of reading any of this. See the [Licence Agreement](#) in this tutorial for more details. If you disagree and read further, may your Apple II fry forever!

Introduction

1. The [AppleWin Emulator](#) can be used to capture screens from a running Apple II program and saved to a Windows BMP file. AppleWin has “hot keys” built-in to do this.
2. These screen captures can then be edited in Windows Paint and “retouched”. (Or they can just be left alone “as-is”.)
3. After editing, AppleWin Screen Captured bmp’s can be converted back to Apple II Graphics in a variety of formats, including Apple II BSAVED images, raster images, image fragments and embedded graphics for Apple II C Programs.
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 HGR (Hi-Res) screens using AppleWin and Windows Paint, and 3 utilities that I have written for the occasion: HMONSTER, BMP2RAG, and BIN2RAG.

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 briefly “touch” on other methods to acquire Apple II HGR screens as well as additional information to help you on your way. Although this tutorial’s primary focus is on the use of the AppleWin emulator as an HGR 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 6-Color HGR Screens

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

AppleWin Screen Mode - Color (standard)

Select color (standard) from AppleWin's configuration dialog which is accessed through the joystick-speaker icon on AppleWin's toolbar. Do not select monochrome.

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, their "screen-grabs" will still 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 AppleWin disk-mages, 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.

Shift+PrintScreen – Capture a 280 x 192 SVGA BMP

1. Make sure you have Windows Explorer open in your capture directory.
2. Pause AppleWin.
3. While holding the Shift Key down, 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.

Note: If you intend to create Apple II HGR image fragments (“sprites”, etc) using my MS-DOS/Windows Utility BMP2RAG make sure you save to 280 x 192 (this size). If you just want to capture full-screens either size is OK (280 x 192 or 560 x 384 – see below).

PrintScreen – Capture a 560 x 384 SVGA BMP

Note: If you intend to create Apple II HGR image fragments (“sprites”, etc) using my MS-DOS/Windows Utility BMP2RAG make sure you ***DO NOT*** save to this size (560 x 384). However if you intend to create Apple II image fragments (“sprites”, etc) using my “native mode” Apple II/ProDOS Utility, BIN2RAG, or if you just want to capture full-screens, either size is OK (280 x 192 or 560 x 384 - see notes above). Keep in mind that a 560 x 384 BMP takes 4 times the disk space as the equivalent 280 x 192 BMP.

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_00000000.bmp, Winter_Games_00000001.bmp, Winter_Games_00000002.bmp, etc”.

Use Short File Names

Rename the bmp’s 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! Or recompile the source code for my MS-DOS utilities using gcc (or some other compiler) and port your finished program (after long-name and other changes, and testing of course) to the platform of your choice. In either case use at your own risk! (I say this kindly but keep in mind that my utilities over-write files without asking, so govern yourselves accordingly).

Getting Started – Editing Captured Apple II HGR Screens



An AppleWin Captured HGR Screen is shown above being edited in Windows Paint for XP in “fat bits” mode (View Large Size).

You can skip this step if you don’t need to make any changes in the BMP that AppleWin captured for you, and simply use one of my utilities (HMONSTER or BMP2RAG) to convert your AppleWin bmp to an Apple II HGR file.

AppleWin Captured BMP Specifics

You need to know a little about the way AppleWin represents the HGR Screen in its captured BMP’s before you go “hog-wild” in Windows Paint and make changes that are too detailed to be converted to Apple II Graphics.

You also need to know a little about the way the HGR screen works when it comes to color on the Apple II. [Appendix A](#) gives you some of this information. You should read [Appendix A](#) now if you are not sure how the HGR screen works. The program source

code and the documentation for DMONSTER, BMP2RAG, and BIN2RAG may also be of some help.

AppleWin's HGR Colors and Editing

There are 6 – Colors on the Apple II HGR screen. Black and White can occur at 280 positions. Green and Violet or Orange and Blue can only occur at 140 positions.

AppleWin Changes between Versions

AppleWin recently changed its colors and the way it saves its screens (along with many other very good changes). Since AppleWin is actively developed, it may also change again. Some support for future changes is already in BMP2RAG and HMONSTER.

For the purposes of this document, and testing the utilities discussed herein, I used AppleWin versions 1.16 and 1.20

In an AppleWin 1.16 captured image (like on a real Apple II generally speaking) Violet or Blue (“odd pixels”) are a “half-shift” (1 pixel) to the left of Green and Orange (“even pixels”). AppleWin version 1.20 uses a slightly different scheme which is more or less equivalent.

Consider Your Editing Layout

When you edit an AppleWin captured bmp (or any Graphic for Conversion to Apple II HGR) you need to consider the positioning of your pixels.

You also need to consider your “color grouping”. On a 280 x 192 bmp (just like on a real Apple II HGR display) Green and Violet cannot exist in the same 7 pixel block as Orange and Blue, and when every even “block” of 7 pixels joins every “odd block”, 1 “colored” pixel will be split between blocks. Therefore, you are best to consider thinking and grouping in 14 pixel blocks when editing these images (if you can).

When a Left Blue Pixel or a Left Violet Pixel meets a Right Orange pixel or a Right Green Pixel, a black gap occurs. When a Left Orange Pixel or a Left Green Pixel meets a Right Blue Pixel or a Right Violet Pixel a white gap occurs.

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 14 pixel blocks if you can.
- Work in only Green and Violet or Blue and Orange if you can.
- Group only Green and Violet or Blue and Orange in a 7 or a 14 pixel block.

- Don't expect perfect results, especially the first time.
- Don't shoot me. I am only the messenger (optional).

Note: The point of origin of these pixel blocks is the left edge of the image (or the HGR screen).

Since we are just getting started, that is enough about layout for now. We also need to know about colors.

Colors in an AppleWin Captured HGR Image

You cannot just use any color you please and expect it to map properly to the Apple II. In an AppleWin Captured BMP Image, you must do any editing with AppleWin Colors 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 AppleWin colors with AppleWin 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 new colors can be defined from there.

The table below gives the RGB values and the mapping of AppleWin's HGR Colors, and for the more technically inclined, the palette position in AppleWin's SVGA (256 color) bmp header. The AppleWin bmp image data follows the header. Like all 8-bit 256 color uncompressed bmp's, the image data in a bmp is simply an upside-down raster-oriented “dump of bytes” which are indexes into the RGB values in the header.

Table of AppleWin HGR RGB Color Values

Note: RGB Values in decimal (required by Paint) in AppleWin Index Order:

Version	1.16			1.20			Index
	Red	Green	Blue	Red	Green	Blue	16-21
BLACK	0	0	0	0	0	0	0x10
WHITE	255	255	254	255	255	254	0x11
BLUE	0	128	255	13	161	255	0x12
ORANGE	240	80	0	242	94	0	0x13
GREEN	32	192	0	56	203	0	0x14
VIOLET	160	0	255	199	52	255	0x15

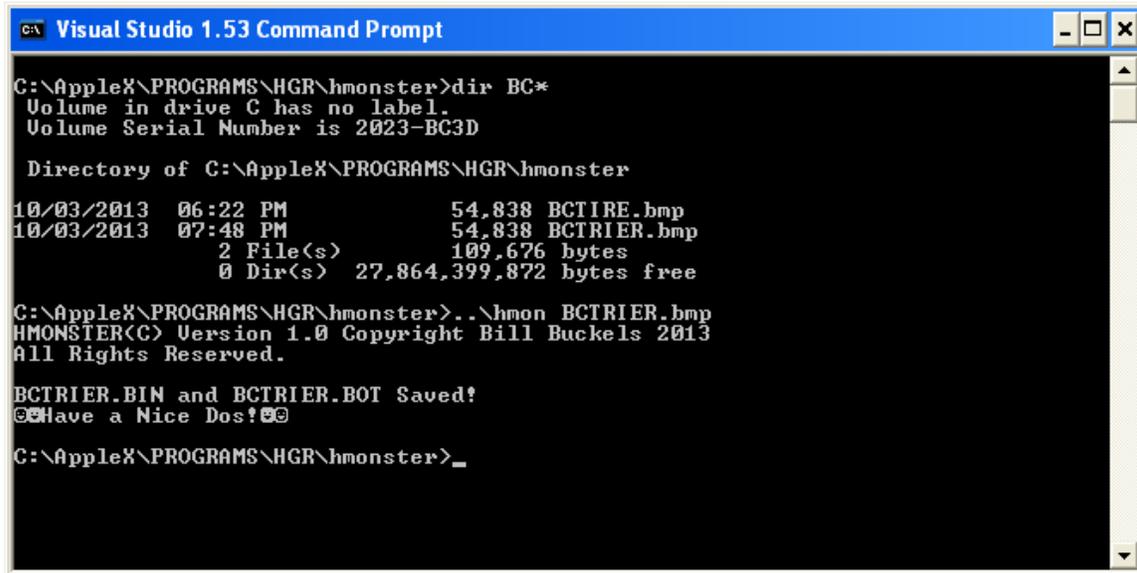
This tutorial is not meant to teach you how to use Windows Paint to edit bitmapped images. The Windows Paint help file and practice and patience will do that.

Using Windows Paint to create a “clipped” image fragment for BMP2RAG is also beyond the scope of this tutorial. The BMP2RAG documentation and source code and your own experience will teach you that. Conversely, you can create clipped images

(HGR RAG files) on the Apple II itself using BIN2RAG. Again, read the BIN2RAG documentation, source code and use BIN2RAG yourself to learn more.

Since we are just getting started, that is enough about Windows Paint for now!

Getting Started – Converting Captured Apple II HGR Screens to Apple II Format



```
C:\AppleII\PROGRAMS\HGR\hmonster>dir BC*
Volume in drive C has no label.
Volume Serial Number is 2023-BC3D

Directory of C:\AppleII\PROGRAMS\HGR\hmonster

10/03/2013  06:22 PM                54,838 BCTIIE.bmp
10/03/2013  07:48 PM                54,838 BCTRIER.bmp
             2 File(s)                109,676 bytes
             0 Dir(s) 27,864,399,872 bytes free

C:\AppleII\PROGRAMS\HGR\hmonster>..\hmon BCTRIER.bmp
HMONSTER(C) Version 1.0 Copyright Bill Buckels 2013
All Rights Reserved.

BCTRIER.BIN and BCTRIER.BOT Saved!
Have a Nice Dos!

C:\AppleII\PROGRAMS\HGR\hmonster>
```

The screen above “says it all”. When your AppleWin screen-captured bmp is ready to be converted into an Apple II HGR file, from MS-DOS (or DosBOX), or from a Windows cmd Window, with HMONSTER “on path” simply type “hmon” followed by the bmp filename. After conversion, place the converted HGR file on an Apple II disk (or whatever your intention is). By the way, if you use BMP2RAG the process for converting from the command line is the same.

You can also launch either utility by clicking-on the program (hmon.exe or bmp2rag.exe) in Windows Explorer or from the “DOS prompt” without a bmp filename, and you will be prompted to enter the filename.

This more-or-less ends the “unofficial Getting Started tutorial”. But if you are still interested in hearing about my utilities and some other related stuff, a little more follows.

About HMONSTER and BIN2RAG, and BMP2RAG

You have the option of using your choice of 2 of my utilities to convert AppleWin HGR bmp files back to Apple II HGR Formats:

- HMONSTER (hmon.exe) – Full-Screen a2 HGR Output in 280 x192 or 280 x 160 only. Monochrome and Color BMP’s are supported for input.
- BMP2RAG (bmp2rag.exe) – Full-Screen or “Clipped” a2 HGR Output including embedded and compressed output. Monochrome BMP’s are not supported.

These are both 16-bit MS-DOS command-line programs (console applications). They provide more features than converting AppleWin screens. Some of the options they provide are quite technical and beyond the scope of this tutorial. The differences between how they handle conversion makes them both independently useful but when it comes to simply converting from an AppleWin captured fullscreen, the results are identical.

I should note that all of my utilities overwrite files without asking for permission. Work on a copy and of course, use at your own risk.

Licence Agreement

You have a royalty-free right to use, modify, reproduce and distribute HMONSTER, BIN2RAG, and BMP2RAG, 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, and I hope they will.

HMONSTER – A “Round Trip” Utility



Hmon.exe started life innocently enough... in the beginning it was called hmono.exe and only worked with 2 color bmp files. AppleWin support was not added yet. One day while struggling with some “fancy features” in BMP2RAG, I decided to take a break and add one-way support for old IBM-PC GWBASIC 320 x 200 x 4 color BSAVED images (now-obsolete) to hmono.

I made discovery after discovery during my testing, most which had more to do with color conversion than monochrome conversion. As HMONSTER grew, I added support for 16-color BMP conversion, then 24-bit BMP support for Windows Clipboard grabs of AppleWin Windows, (completely ignoring the fact that AppleWin itself saves its own screens). After finally adding AppleWin support for 256 color bmps's and after renaming hmono to HMONSTER (for obvious reasons), I revisited BMP2RAG and added AppleWin Support for 280 x 192 AppleWin Screen Captures.

HMONSTER (hmon.exe) was supposed to be a simple utility, and BMP2RAG was (and still is) a little more advanced and harder to use for most, since it manipulates color through algorithmic patterns that are a little more complex.

Read the documentation and source code for HMONSTER to discover more. This tutorial is about simple conversion of AppleWin screen captures, so now we need to move-on while our “eye-patches” are still firmly in place. Talk of advanced subjects like lunar landings needs to wait for another document.

BMP2RAG – a “One-Way” Ticket



BMP2RAG is a MS-DOS command line graphics file converter which produces Apple II full-screen HGR files for Graphics-Only (and Mixed Text and Graphics) in 4 different formats from 16 color, 256 color, and 24-bit Windows BMP files.

It also produces partial screen image fragments in 3 different formats (sorry AppleSoft Programmers, no fragments for you, but you can optionally output a BSaved full screen with the fragment in the top-left corner).

Quite a number of features are available and of particular interest for the C programmer is the optional embedded graphics output... which can be inserted directly into an Apple II C program as a byte array.

BMP2RAG does not accept input of 560 x 384 AppleWin screen captures. It is limited to a maximum size input file of 280 x 192. Remember when capturing screens from AppleWin to convert using BMP2RAG, you must hold the Shift key down while pressing and releasing the Print Screen key to produce this size.

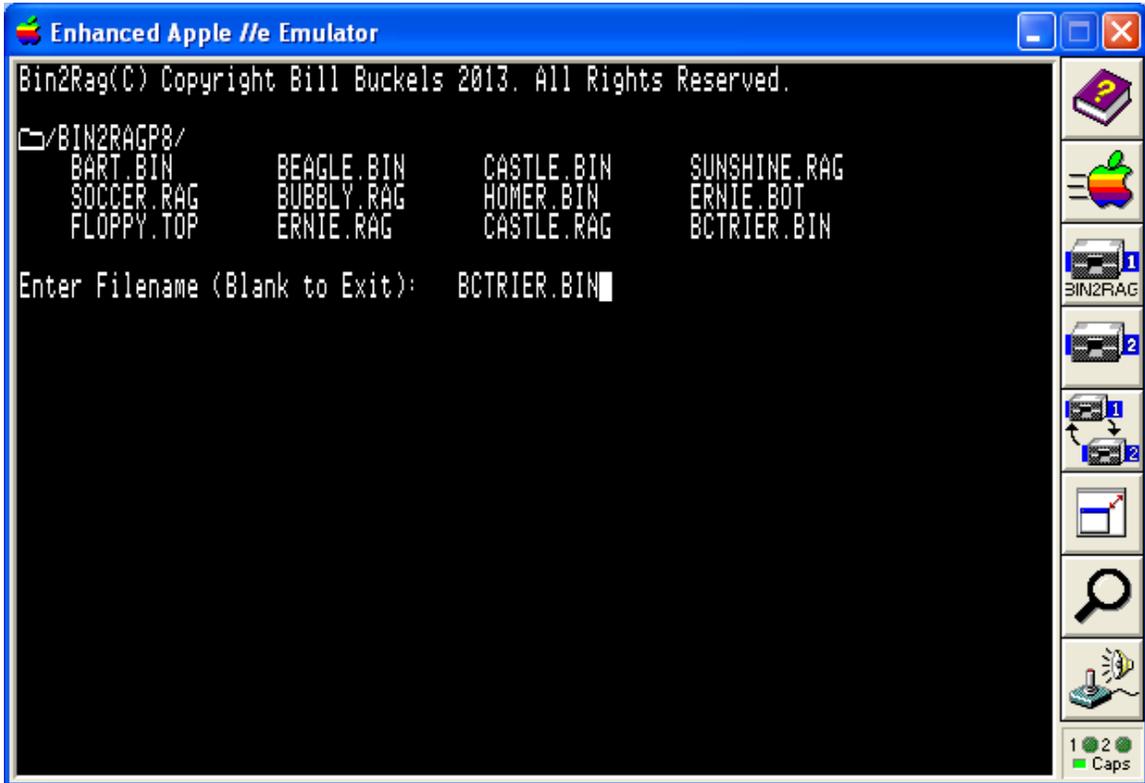
BIN2RAG – HGR Graphics Converter and Clipper for ProDOS 8



I am not going to say much about BIN2RAG except that it runs on the Apple II under ProDOS, and if command-line utilities are not “your thing” when it comes to chunking-up an HGR image that has been created from an AppleWin Screen Capture, or (more likely) you just like to play with HGR bitmapped graphics on the Apple II, this little SYS program may be fun for you to use.

But wait, there's more and if a picture is worth a thousand words, we only have a few more thousand words to go!

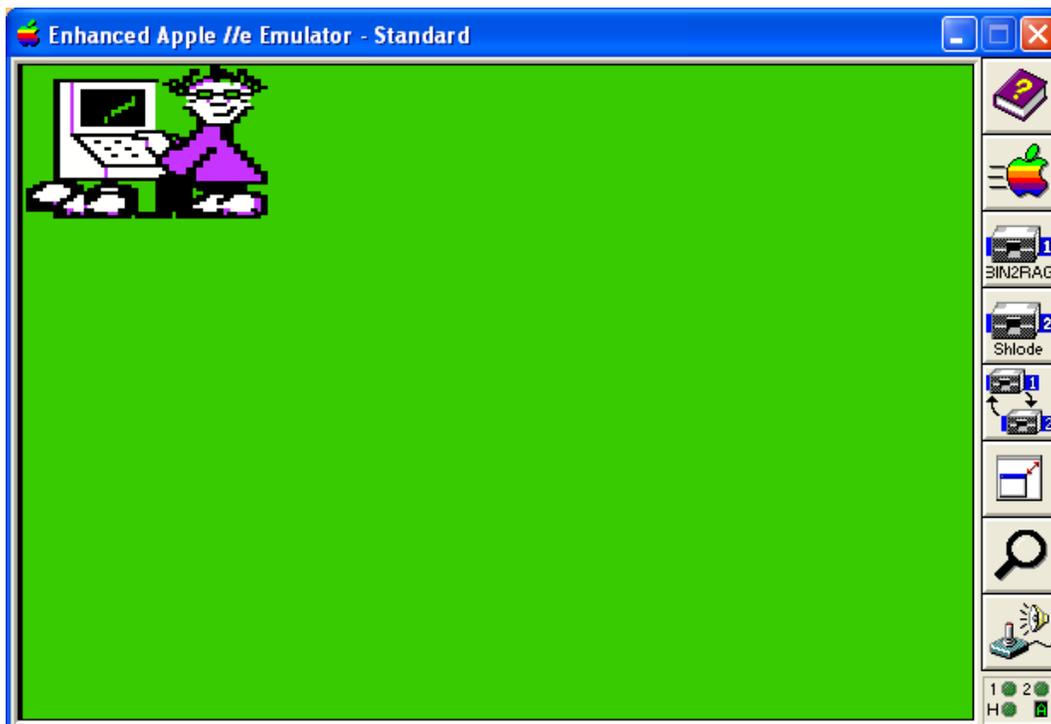
The screen below is BIN2RAG's file selection display. It displays the files in the current directory but also accepts prefixed pathnames from other drives for loading and saving.



THE BOTTOM OF THIS PAGE LEFT UNINTENTIONALLY BLANK

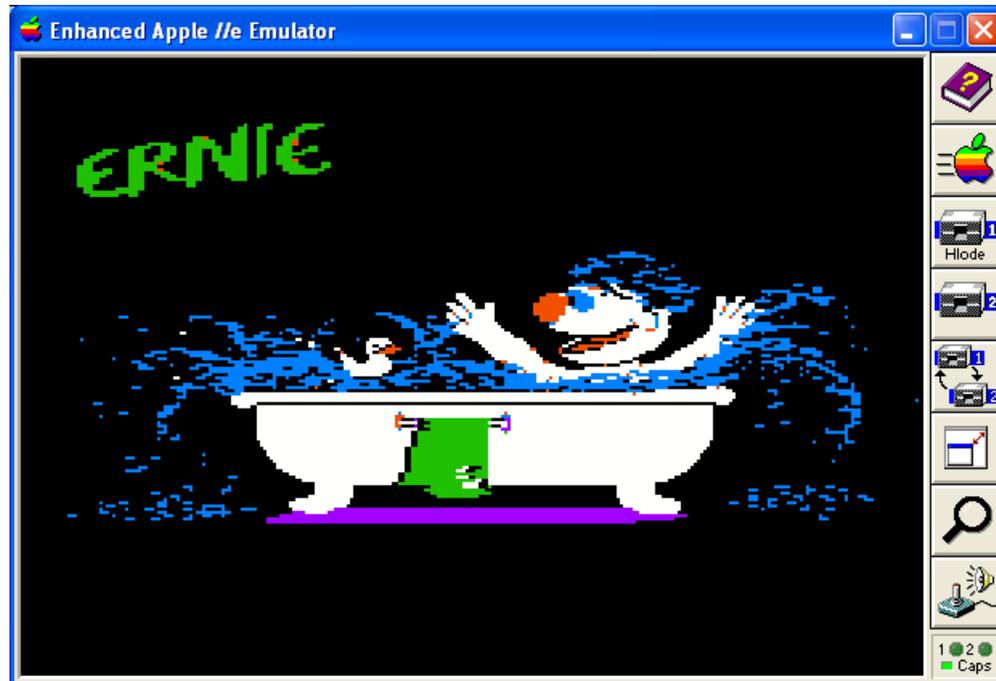


The screen above shows BIN2RAG's "clipboard" around the geek, and shows the menu (which is toggled on and off with the spacebar). The menu commands are obvious. After saving the "geek" he can be loaded back-in again for validation (or whatever). See below... the "geek" has loaded!



Again read the individual documentation, source code and use BIN2RAG, and HMONSTER and BMP2RAG to see how all this works for you.

Closing Remarks



I realize this tutorial document is neither overly technical, nor particularly informative to those of you who are intimate with HGR graphics on the Apple II. Despite that I am ending here, with a short appendix to follow.

I hope those of you who have a use for converting from Windows BMP files to Apple II HGR graphics files will explore this topic further and with enthusiasm, and will add BMP2RAG, HMONSTER, and BIN2RAG 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 BMP2RAG, HMONSTER, BIN2RAG and much more!

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

Have Fun! And All the Best!

Bill Buckels, March 12, 2013
bbuckels@mts.net

Appendix A - Apple II HGR Byte Pair Diagrams

Note - In Figures 1 and 2, the Palette Select Bit could have been "Don't Care", since Black or White are available in either palette by using the same Pixel Pair Pattern.

Figure 1 - Black - Palette 0

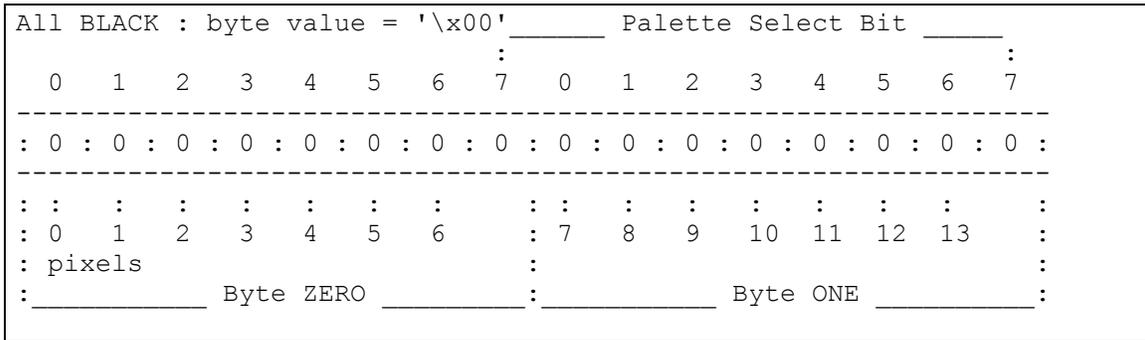


Figure 2 - White - Palette 1

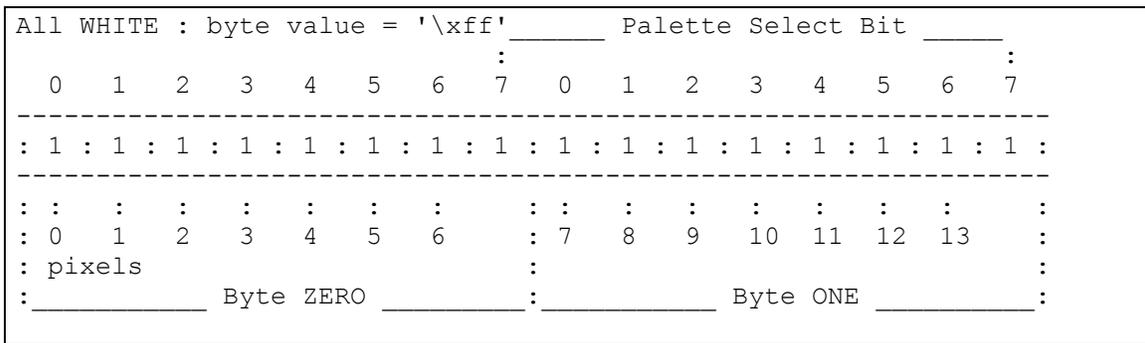


Figure 3 - Green - Palette 0

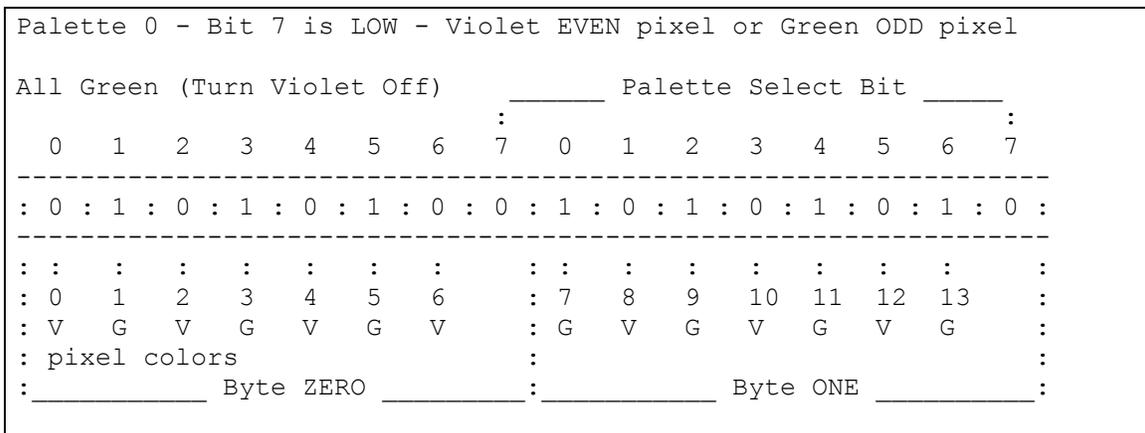


Figure 4 - Violet - Palette 0

Palette 0 - Bit 7 is LOW - Violet EVEN pixel or Green ODD pixel															
All Violet (Turn Green Off)							Palette Select Bit								
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
-----							-----								
: 1	: 0	: 1	: 0	: 1	: 0	: 1	: 0	: 0	: 1	: 0	: 1	: 0	: 1	: 0	: 0
-----							-----								
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
: 0	: 1	: 2	: 3	: 4	: 5	: 6	:	: 7	: 8	: 9	: 10	: 11	: 12	: 13	:
: V	: G	: V	: G	: V	: G	: V	:	: G	: V	: G	: V	: G	: V	: G	:
: pixel colors							:								
: _____ Byte ZERO _____							: _____ Byte ONE _____								

Figure 5 - Orange - Palette 1

Palette 1 - Bit 7 is HIGH - Blue EVEN pixel or Orange ODD pixel															
All Orange (Turn Blue Off)							Palette Select Bit								
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
-----							-----								
: 0	: 1	: 0	: 1	: 0	: 1	: 0	: 1	: 1	: 0	: 1	: 0	: 1	: 0	: 1	: 1
-----							-----								
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
: 0	: 1	: 2	: 3	: 4	: 5	: 6	:	: 7	: 8	: 9	: 10	: 11	: 12	: 13	:
: B	: O	: B	: O	: B	: O	: B	:	: O	: B	: O	: B	: O	: B	: O	:
: pixel colors							:								
: _____ Byte ZERO _____							: _____ Byte ONE _____								

Figure 6 - Blue - Palette 1

Palette 1 - Bit 7 is HIGH- Blue EVEN pixel or Orange ODD pixel															
All Blue (Turn Orange Off)							Palette Select Bit								
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
-----							-----								
: 1	: 0	: 1	: 0	: 1	: 0	: 1	: 1	: 0	: 1	: 0	: 1	: 0	: 1	: 0	: 1
-----							-----								
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
: 0	: 1	: 2	: 3	: 4	: 5	: 6	:	: 7	: 8	: 9	: 10	: 11	: 12	: 13	:
: B	: O	: B	: O	: B	: O	: B	:	: O	: B	: O	: B	: O	: B	: O	:
: pixel colors							:								
: _____ Byte ZERO _____							: _____ Byte ONE _____								