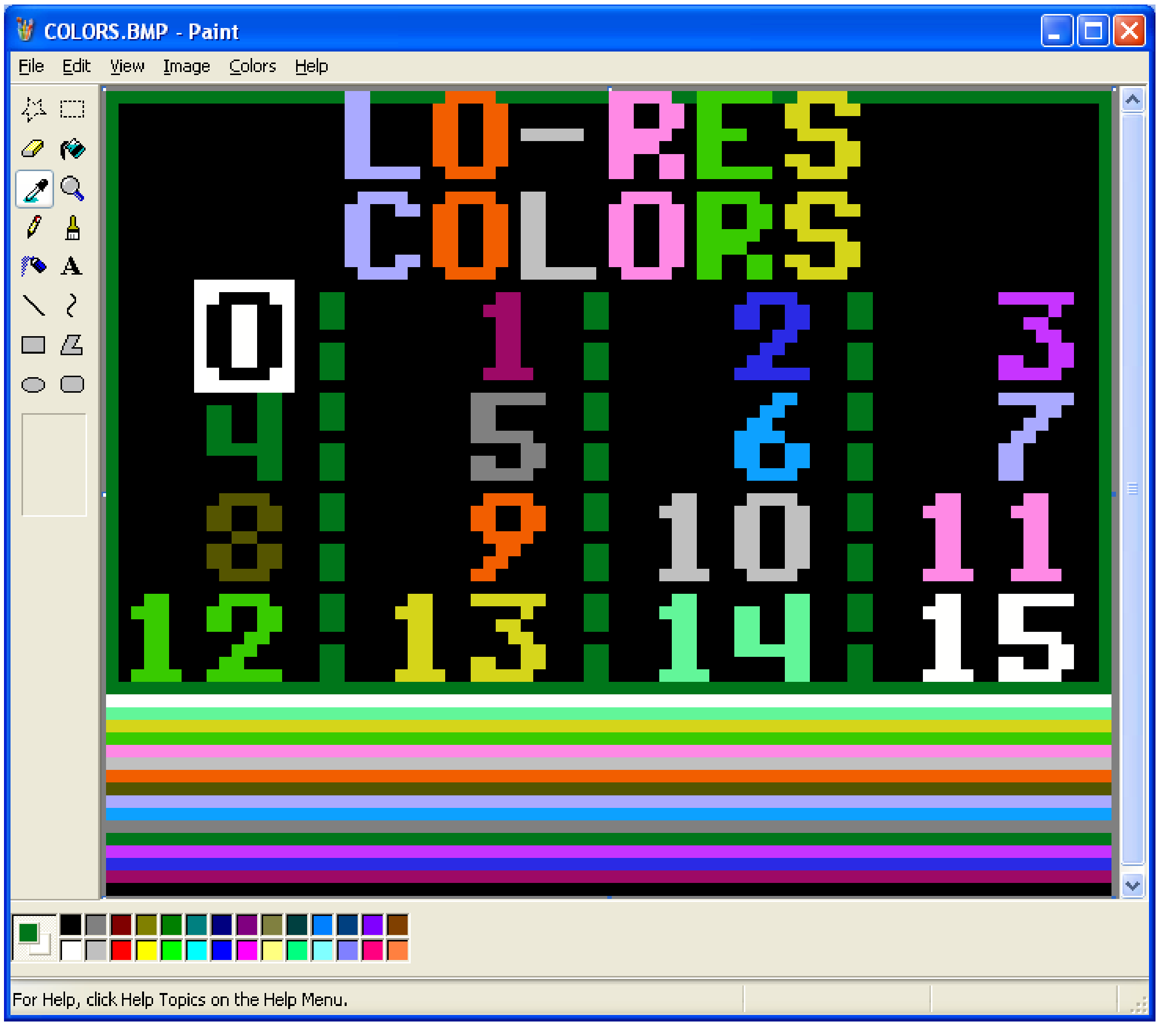
<u>Using AWINDLO and Text Files to Create Lo-Res and Double Lo-Res Screens</u>

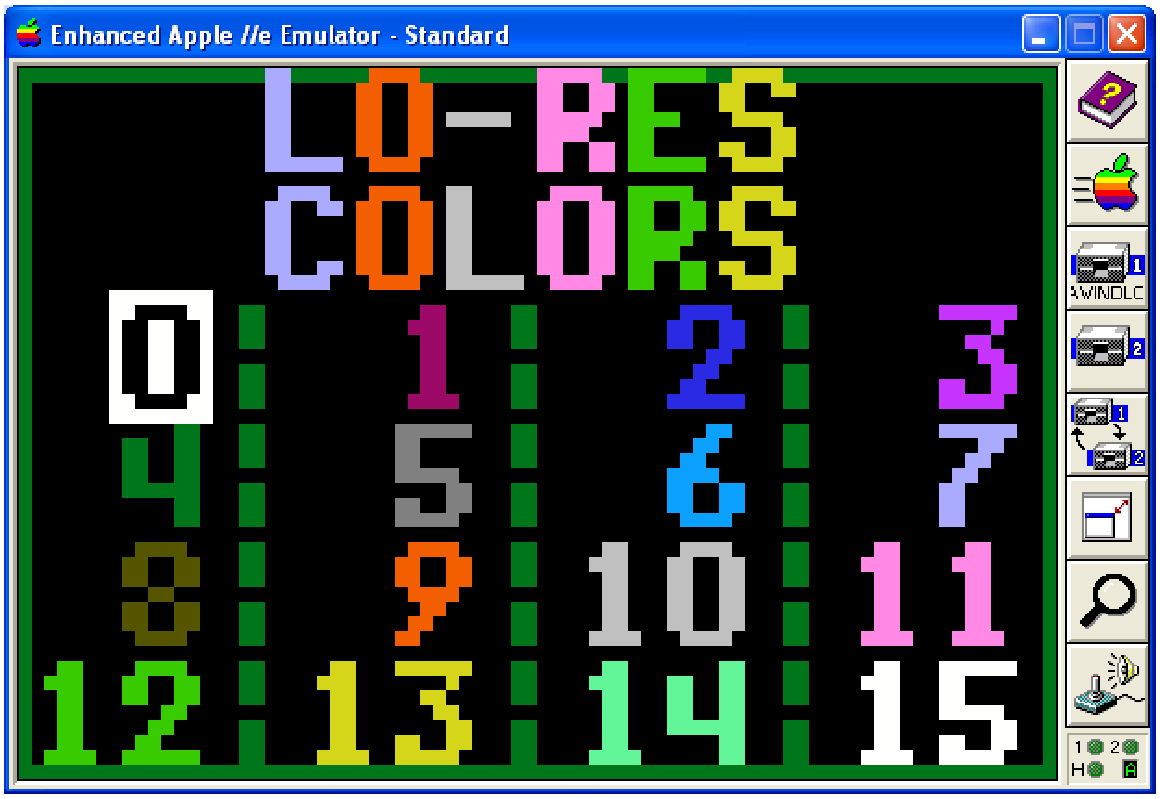
A Tutorial by Bill Buckels, April 2013


COLORS.BMP - Paint
File Edit View Image Colors Help
LO-RES
COLORS
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
For Help, click Help Topics on the Help Menu.

The screen above shows a 24 bit BMP that was originally created by AWINDLO from a 6-line text file. The numbers for each of the Lo-Res (LGR) (and Double Lo-Res (DLGR)) colors were then filled in Windows Paint, with the color corresponding to the Apple II Lo-Res color number. The colors were set to the proper fill-color by selecting from the color bar palette below the image using Windows Paint's eyedropper tool. The color bar palette is arranged in LGR/DLGR color order, starting with color 0 (black) at the bottom, and finishing with color 15 (white) at the top of the palette (directly below the image).

The BMP as shown is now ready to be converted to an Apple II DLGR screen file using BMP2LO. The Apple II screen file can then be placed on an Apple II disk for display in an Apple II program.

This tutorial will provide you with an overview of producing similar Apple II screen files from text files in the MS-DOS/Windows environment using the command-line utilities AWINDLO and BMP2LO in conjunction with Windows Paint.



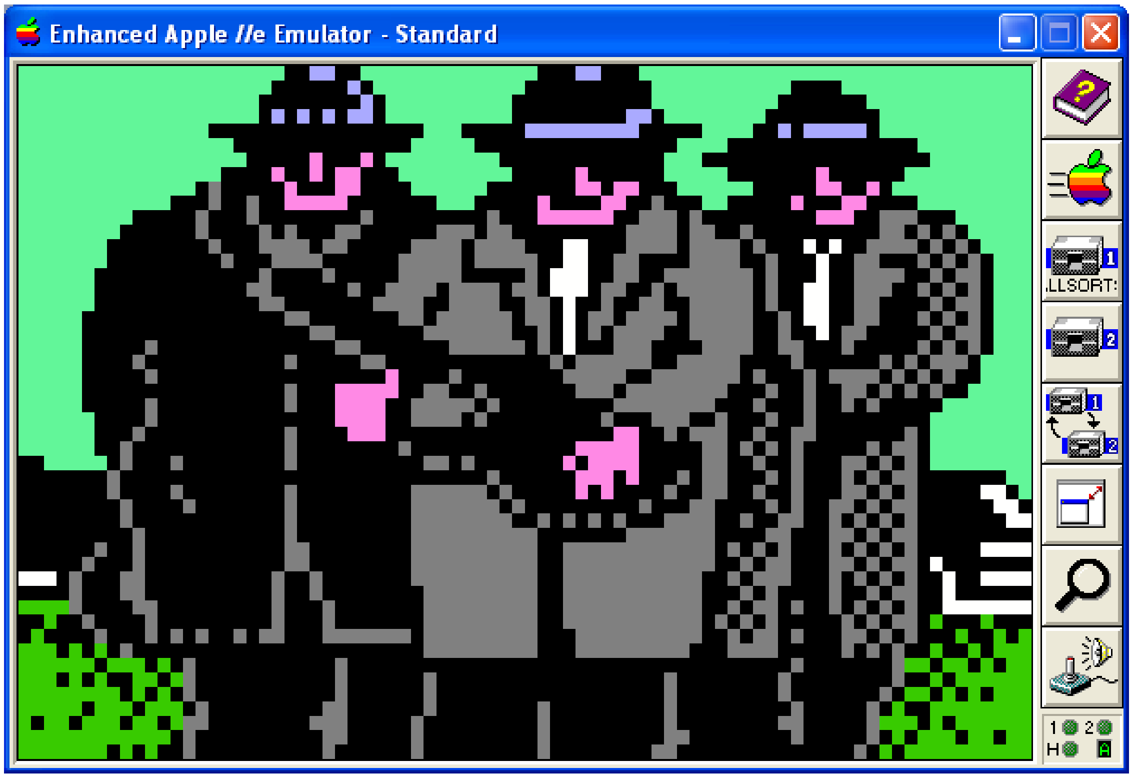
The DLGR screen above started life as a 6 line text file and was converted to its current glory using the utilities AWINDLO.exe and BMP2LO.exe and the methods described in this tutorial. Now it happily resides on the Apple II.

Table of Contents

Disclaimer: See the **Licence Agreement** in this tutorial for more details.

Writing a program to put letters on a DLGR or LGR picture is a strange undertaking. First off, like anything else for the Apple II, one never knows if anyone will ever use such a thing. Secondly, programming this contraption was analogous to being a rat in a maze (writing about it has been equally rewarding provided one enjoys a good maze game). And thirdly, I am sure glad it's done. I would like to thank the AppleWin Development Team whose excellence inspired me to finish this off, and not walk away from it, (like I did about 4 years back when my Double Lo-Res Aztec C65 code fell to pieces in ProDOS). It's back together now of course with much more to follow.

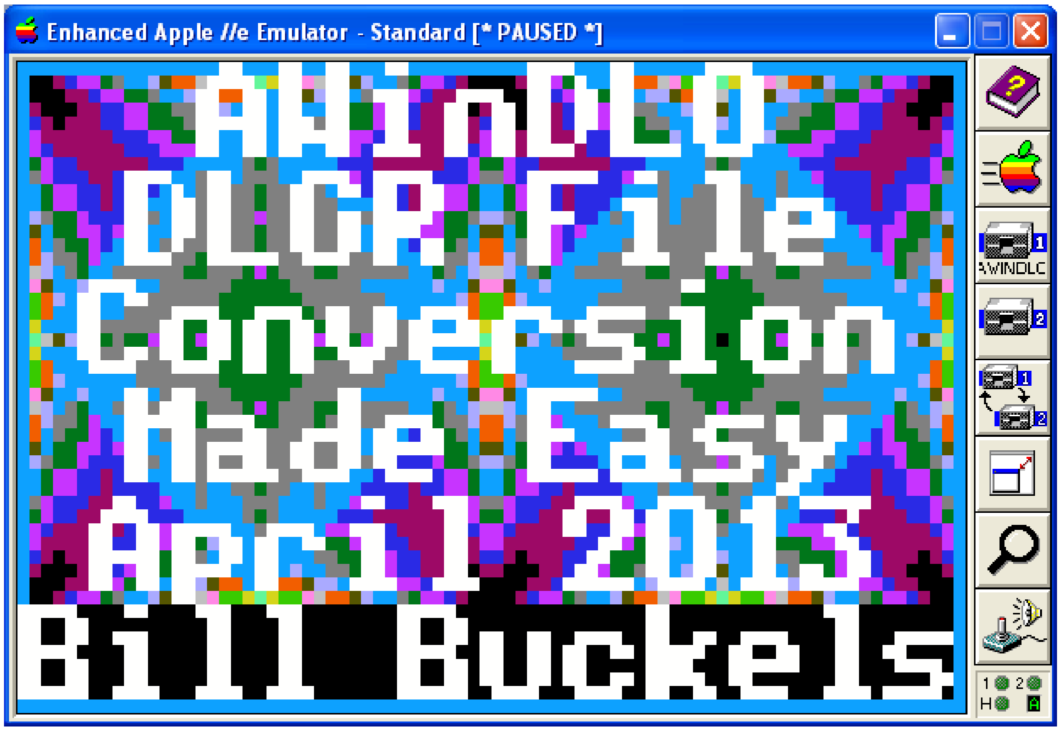
Bill Buckels
bbuckels@mts.net
April 7, 2013



Licence Agreement

You have a royalty-free right to use, modify, reproduce and distribute AWINDLO and BMP2LO, 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.

Introduction



The AWINDLO utility was conceived to process AppleWin Screen Captures of Double Lo-Res (DLGR) screens back into native mode Apple II DLGR files, and to produce a smaller BMP file better suited for editing in Windows Paint. AWINDLO now supports conversion to 4 types of output BMPs (both single and double scaled), and now outputs both Double Lo-Res (DLGR) and Standard Lo-Res (LGR) Apple II Files:

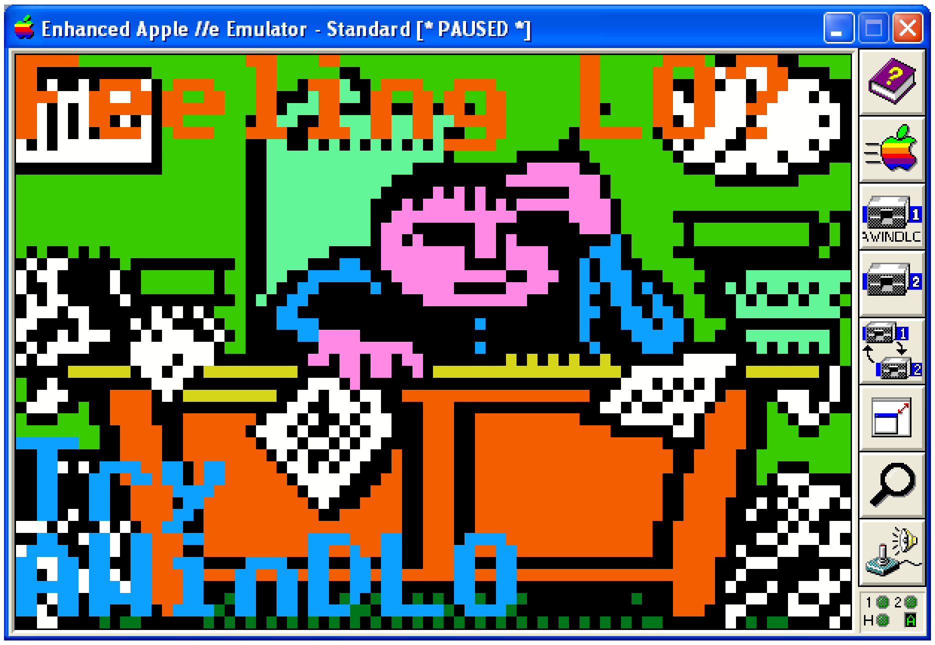
AWINDLO Output Files

Default BMP output	160 x 128 x 24 bit BMP with "color bar palette"
Optional BMP output	160 x 128 x 24 bit BMP without "palette"
	80 x 64 x 24 bit BMP with "color bar palette"
	80 x 48 x 24 bit without "palette"
Default Apple II DLGR	80 x 48 x 16 color double lo-res BSaved DL1 and DL2 and Raster DLO Images
Optional Apple II LGR	40 x 48 x 16 color standard lo-res BSaved SL2 and Raster SLO Images
Note: BaseName	AWINDLO Output Files have the same” basename” as the input file. AWINDLO “backs-up” input files as needed, by renaming them with the same basename with a different extension to avoid overwriting original files if a naming conflict occurs.

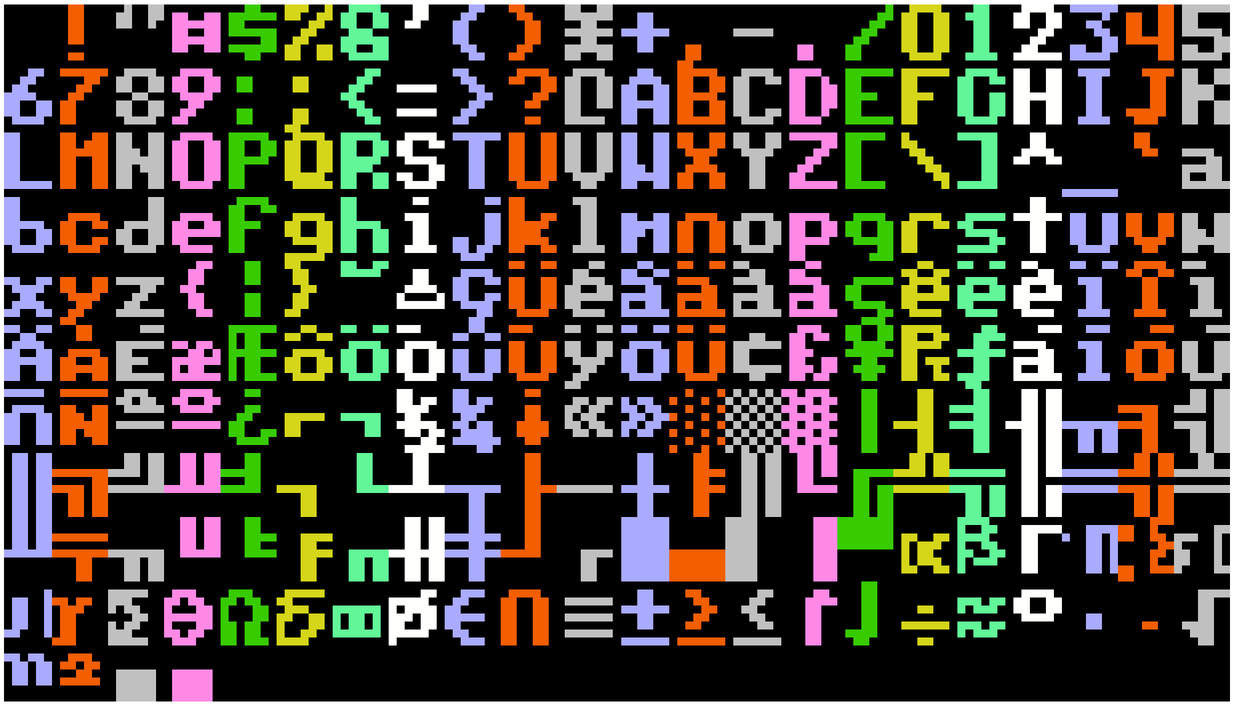
AWINDLO Input Files

AppleWin Single Lo-Res (LGR) screen captures are in the same colors and format as their DLGR counterparts (the only difference is that they are twice as “coarse” horizontally). A Single Lo-Res BSaved Image (SL2 file) is in exactly the same format as a double lo-res DL2 file, because they are both simply a screen memory dump of the same area in main memory (bank 0). An easy way to half-scale a double lo-res image for the Apple II single lo-res display is to load a DL2 into memory… no additional work is necessary unless you want to title it, simplify the image, or otherwise tune-it up so it looks better.

BMP	560 x 384 x 256 Color AppleWin Screen Captures with .BMP extension. LGR and DLGR screen captures from versions 1.16, 1.17, 1.20, and 1.22 Renamed to match MS-DOS 8.3 character naming convention.
256	560 x 384 x 256 Color AppleWin Screen Captures with .256 extension. AWINDLO renames the AppleWin .BMP to .256 during conversion. They are not renamed after that and can be repetitively processed or deleted.
Apple II BSaved	80 x 48 x 16 color double lo-res BSaved DL1/DL2 file pairs 2032 bytes 40 x 48 x 16 color single lo-res BSaved SL2 files 1016 bytes
Apple II Raster	80 x 48 x 16 color double lo-res DLO Raster Images 1922 bytes 40 x 48 x 16 color single lo-res SLO Raster Images 962 bytes 80 x 40 x 16 color double lo-res DTO Raster Images 1602 bytes 80 x 40 x 16 color single lo-res STO Raster Images 802 bytes
Text Files TXT ASC	1-6 line plain text files can be used alone to create a text only graphic In both single lo-res and double lo-res formats. When used with one of the other input files, they produce a titled copy of an existing single lo-res or double lo-res image.



<u>On a Clear Font You Can Text Forever</u>



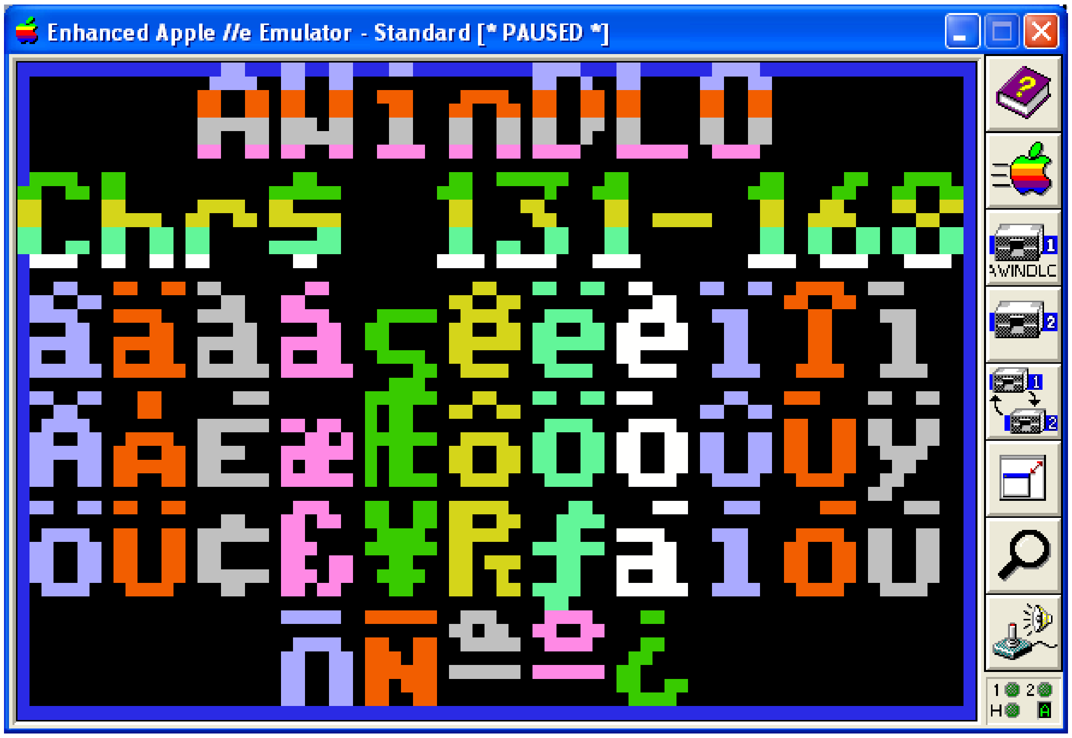
AWINDLO’s LGR/DLGR font is shown above in Ascii order. This is an extended version of the 7 x 8 bitmap font used throughout the AppleX Aztec C65 distribution. AWINDLO’s font starts at Ascii character 32 and ends at Ascii character 255. (The AppleX version ends at character 168… the “upside-down question mark”.)

Multi-Lingual Characters, Currency Characters, Extended Characters

AWINDLO’s font offers the ancient IBM-PC hi-Ascii “multi-lingual” characters in addition the usual printable low-Ascii characters. When I made the original version of this font (late 80's, early 90's), I captured the bitmaps from the IBM-PC's 8 X 8 x 40 character CGA display and narrowed each one to fit the Apple II's 7 x 8 x 40 character HGR display. I needed the special characters for a bi-lingual (English-French) series of Apple II programs I wrote that were used in Schools across Canada.

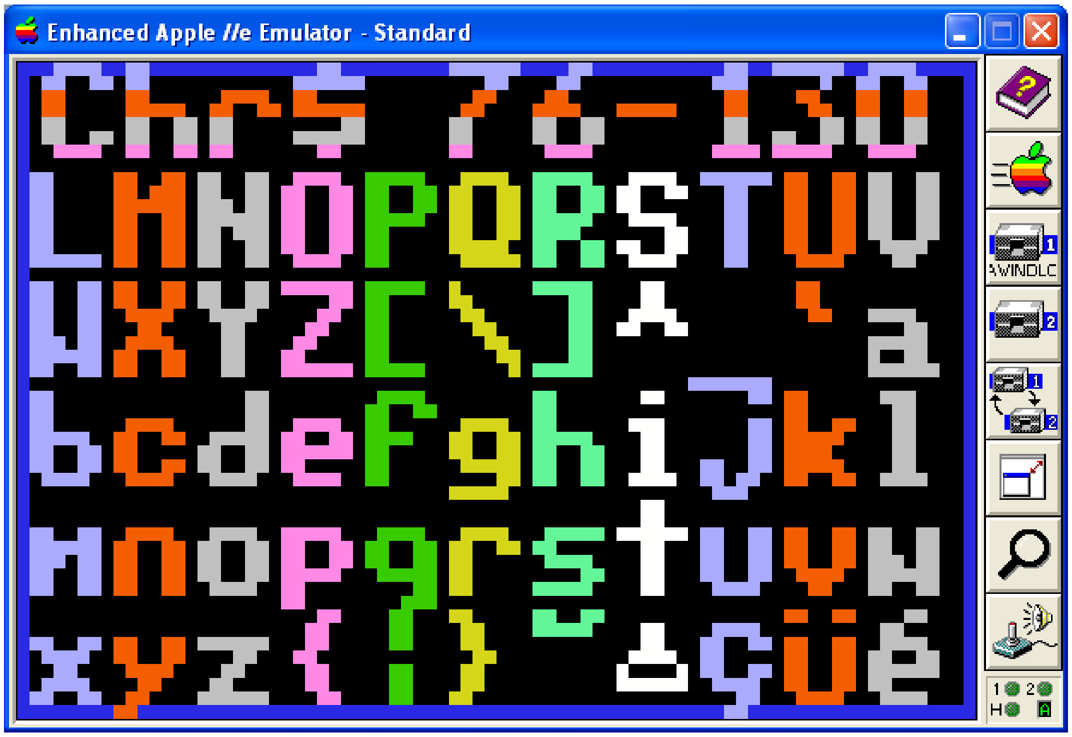
By contrast, when preparing the font routines in the Aztec64 AztecC65 Commodore 64 cross-compiler distribution, I used the bitmaps from the IBM-PC 8 x 8 CGA font without modification.

In order to use the Multi-Lingual characters, and other special characters like Currency Characters in AWINDLO’s extended character set, you must use an MS-DOS editor. Consider an MS-DOS “programmer’s” text editor, like Sammy Mitchell’s QEDIT (Q.exe) which offers full support for generating MS-DOS extended characters, (with NUM-lock on, hold-down the ALT key, enter the Ascii numeric value on the numeric pad on your keyboard, then release the ALT key). Q.exe also has a built-in MS-DOS Ascii look-up chart .

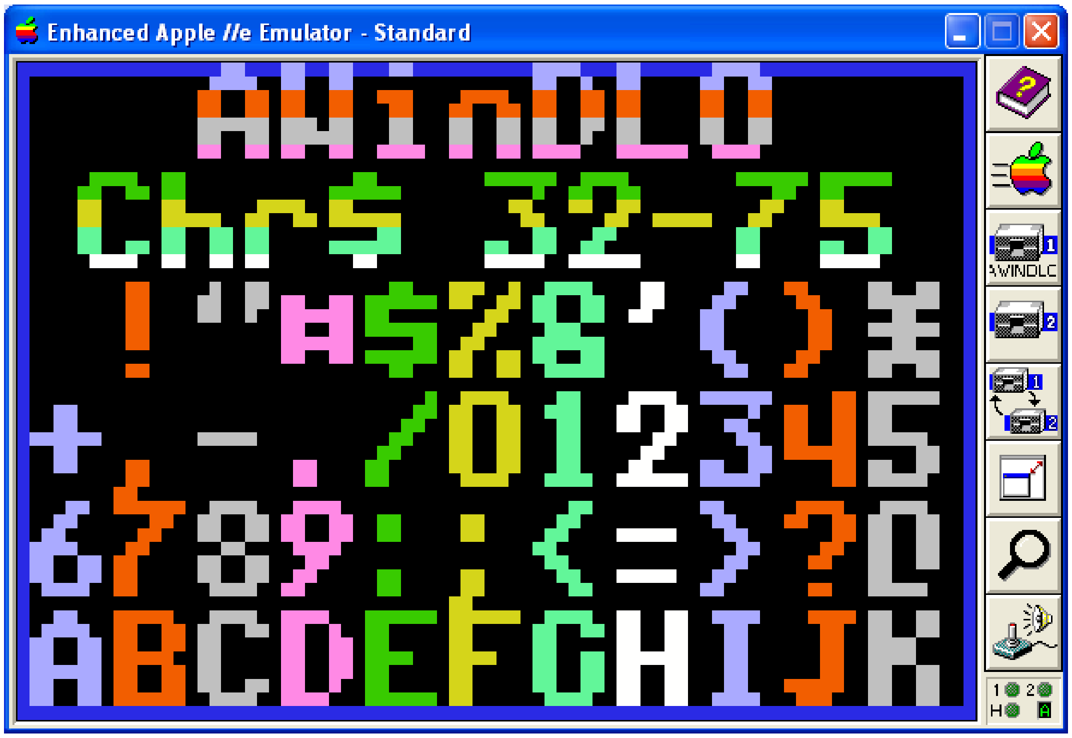


```
AWINDLO
Chrs 131-168
âäàåçêëèïîì
ÄÅÉæÆôöòûùÿ
ÖÜ¢£¥₧ƒáíóú
    ñÑªº¿
```

A Windows text editor might match the appearance of the characters you see in AWINDLO's character set when you use Alt-Numeric Pad entry, but those characters may not remap correctly to AWINDLO's character set.

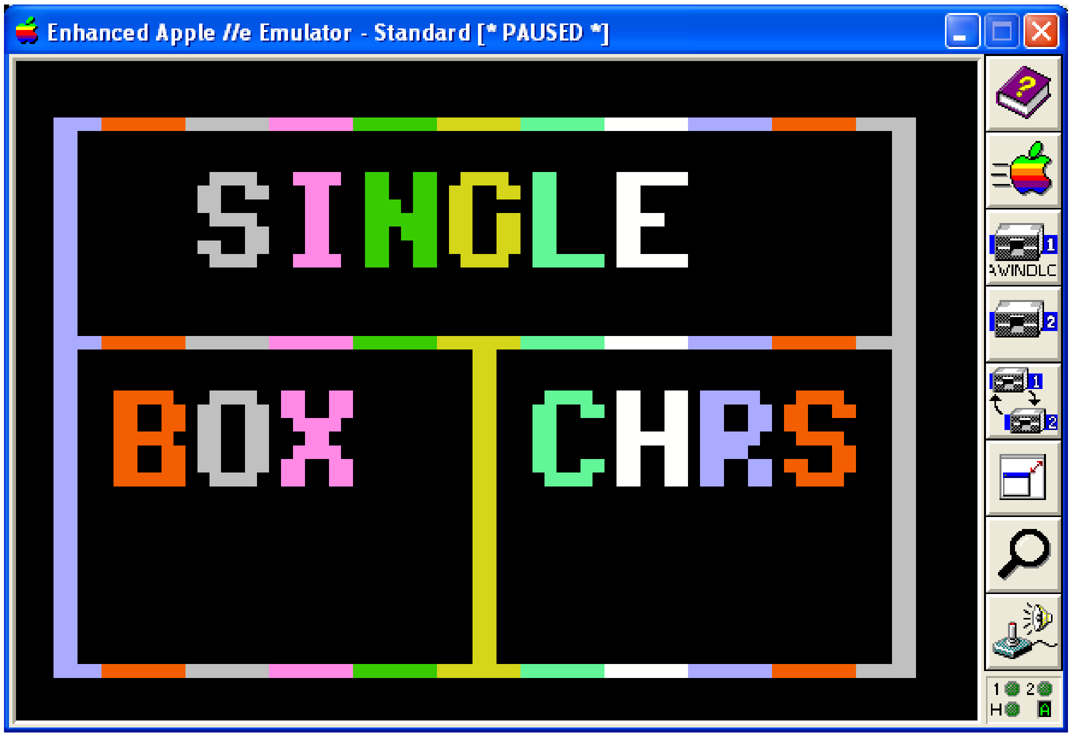


```
Chrs 76-130
LMNOPQRSTUV
WXYZ[\]^  `a
bcdefghijkl
mnopqrstuvw
xyz{|}~⌂Çüé
```



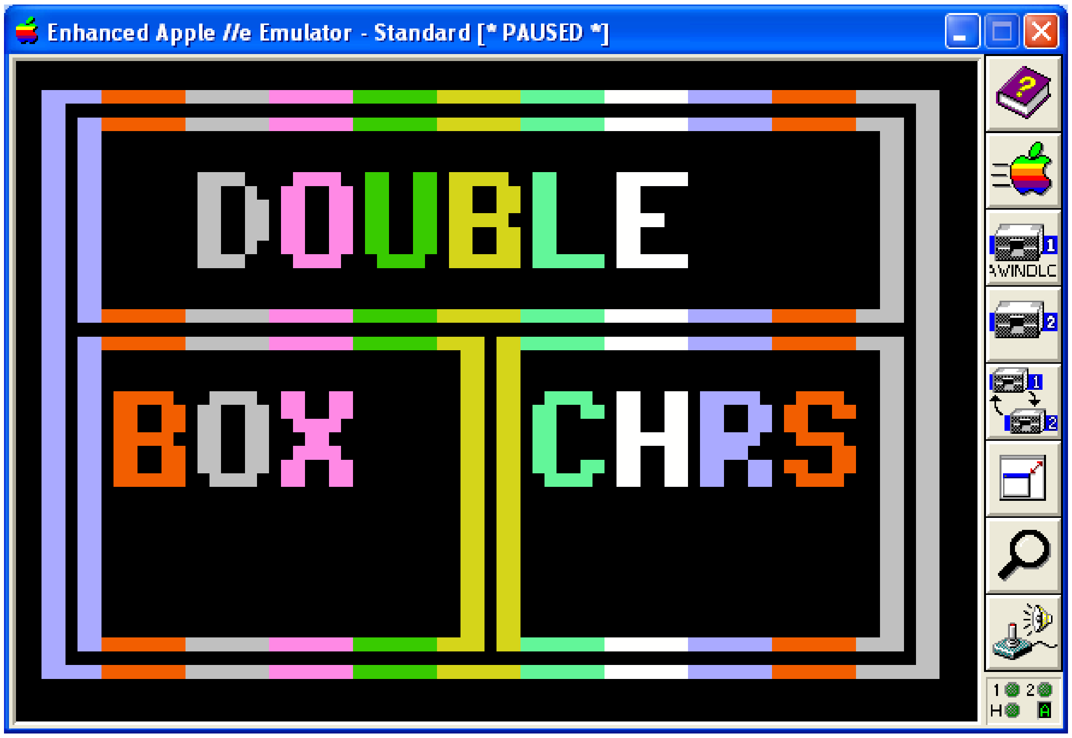
```
AWINDLO
Chr$ 32-75
!"#$%&'()*
+,-./012345
6789:;<=>?@
ABCDEFGHIJK
```

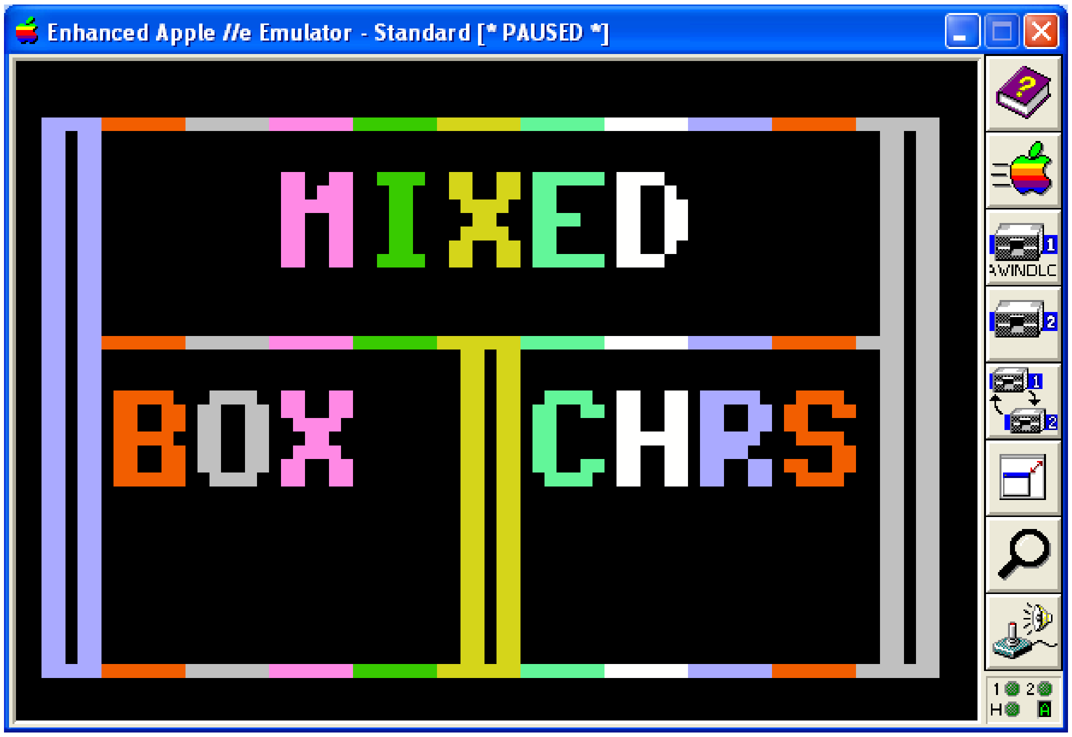
Drawing Graphics Characters in AWINDLO's Font



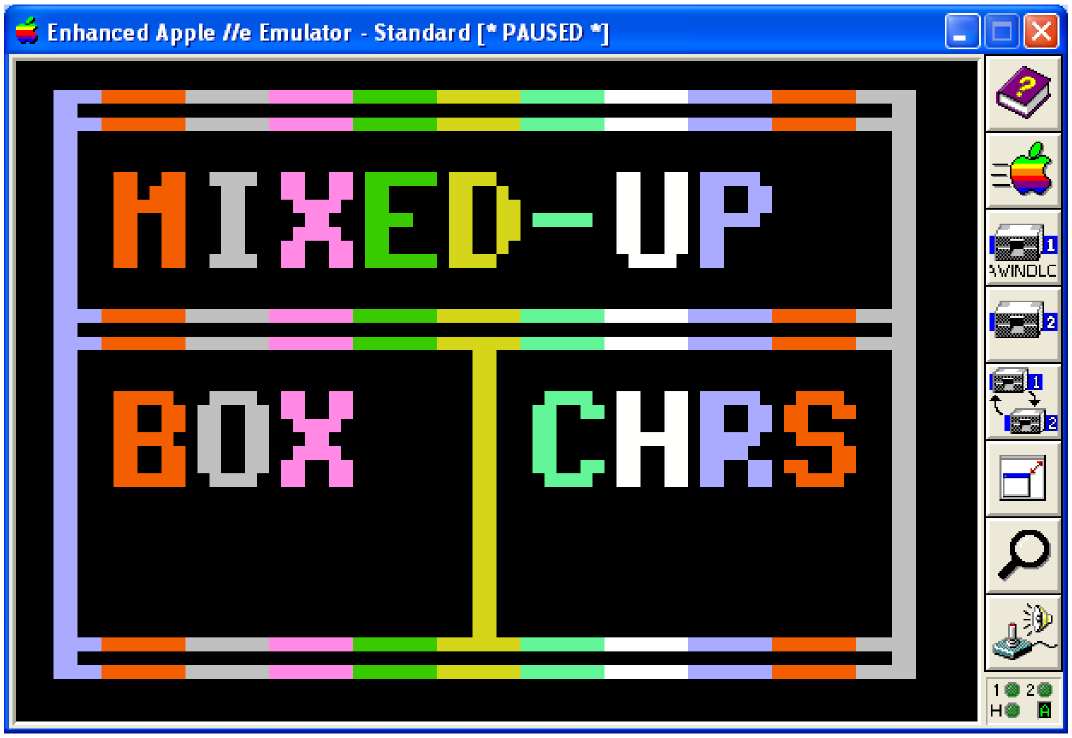
```
SINGLE
BOX   CHRS
```

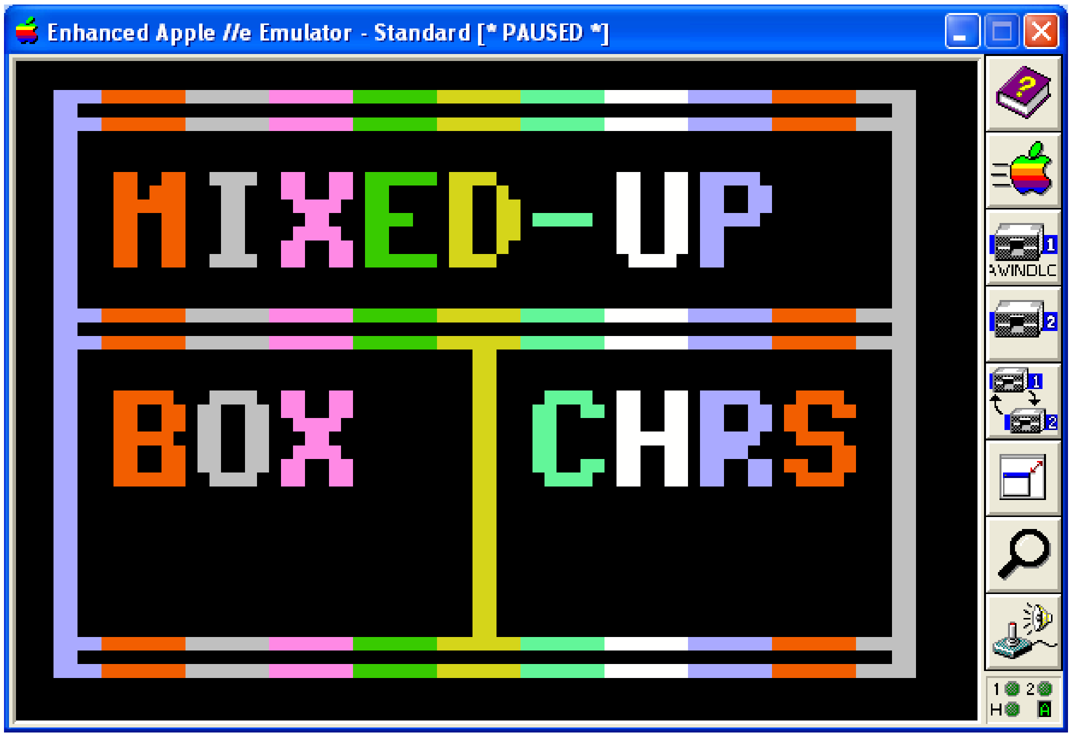
AWINDLO offers the complete set of the IBM-PC's MS-DOS "box drawing" characters.



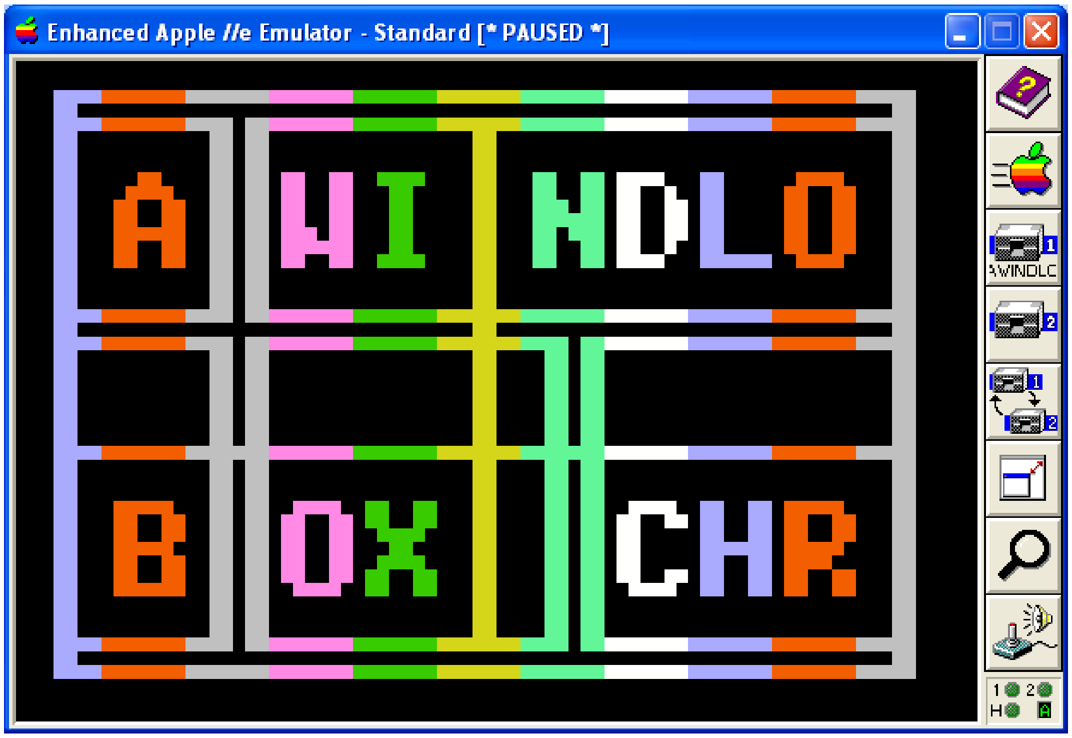


Some of you may remember that decades ago IBM-PC programmers regularly used these extensively to embellish colored text screens in most of the applications of the day.

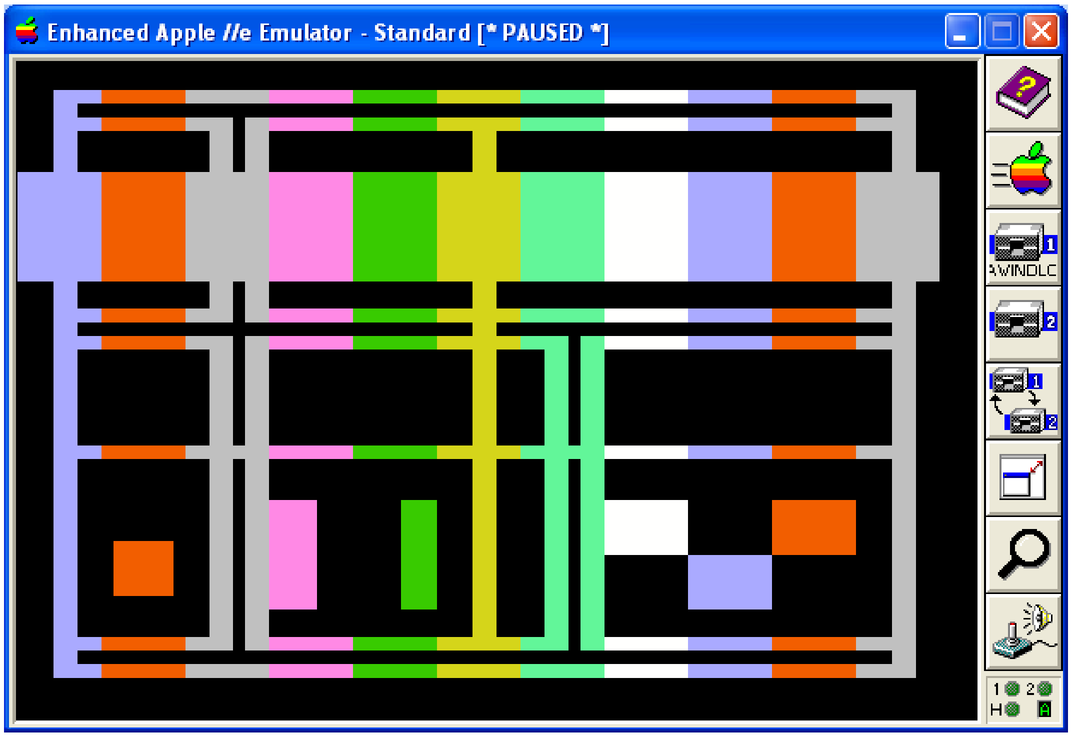




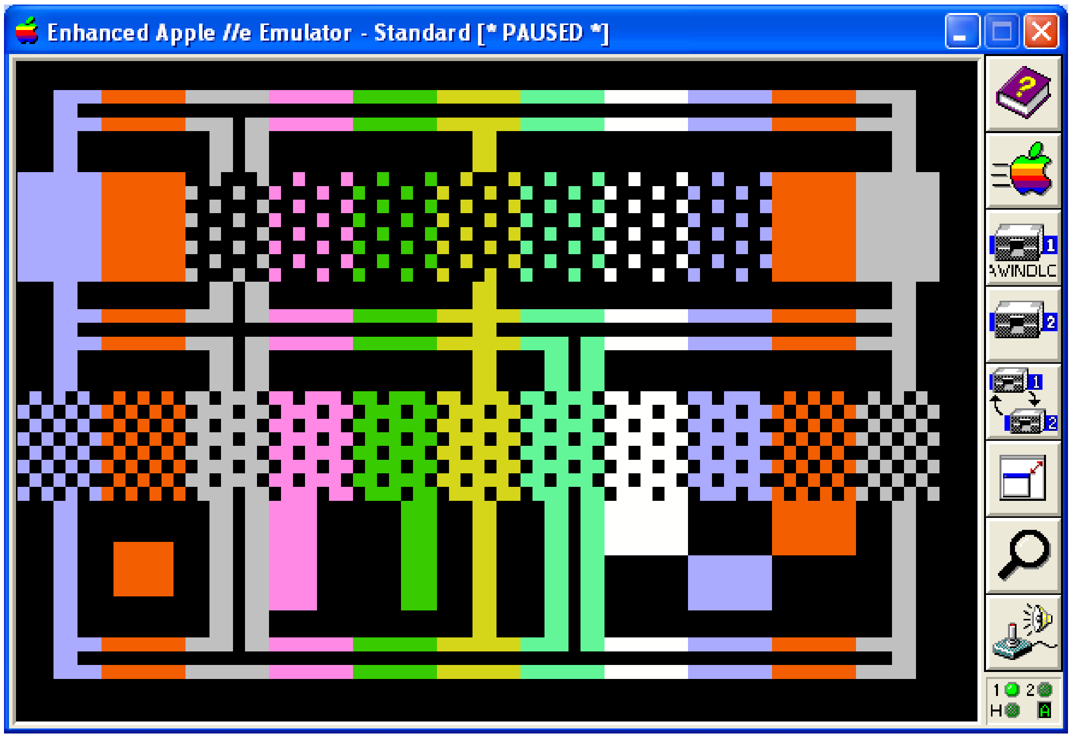
MS-DOS Screen designers like Ian Davis's "The Draw", which is still distributed with ClipShop were part of many PC programmers' toolboxes. In fact I used The Draw to quickly generate "ASC" files to produce these example "box" screens with AWINDLO.



There are 11 characters per line shown above. 10 characters would have centered the box.



But AWINDLO is not a paint program. You already have Windows Paint which is a good Paint Program. AWINDLO is a programmer's utility, and you will need to generate the graphics characters in an MS-DOS editor and *NOT* in a Windows Editor.



AWINDLO was written as part of the tool-chain for the AppleX distribution of the Aztec C65 MS-DOS cross-compiler for Apple II ProDOS 8. Apple II Programmers can write their own vector drawing programs in Aztec C65, which already has many good “real” graphics programs complete with source code, for the 4 primary color graphics modes (and 2 primary “monochrome modes”) of the Apple II.

Scalability

Because of the small relative size of the LGR and DLGR screens compared to today’s monstrous size screens, using Windows Paint to put text on a BMP destined for the Apple II LGR and DLGR modes is lots of “busy” work, often with less than stellar results, because the fonts in Windows Paint are not designed to scale evenly, and in modes with higher color depth, they produce artifacts which do not map to the fixed Lo-Res palette.

It might seem strange to hear me recommending the 7 x 8 Apple II font used by AWINDLO for a display that can only support a maximum of 6 lines of 13 or so characters per line. However something like a 5 x 6 font would be less readable and only offer 8 lines of 16 characters. Consider that DLGR's mixed mode offers 4 text lines of 80 characters per line at the bottom of the screen. Mixed-mode is a far better alternative for text of any consequence, like instructions in a computer program, so any graphics font in the Lo-Res graphics modes will only be used for titling and stuff anyway.

AWINDLO’s Apple II font converts properly, it can be painted easily, and it is simple to do layouts in a text file (even in Notepad if that’s all you have) without mucking about.

AWINDLO Titling Functionality

After making myself weary with all the steps I needed to go through to put readable text on a Lo-Res screen using Windows Paint, I added support to AWINDLO for creating LGR and DLGR "title" screens from plain text files and my own Apple II font.

AWINDLO "titling" features are all done from the command prompt in Windows or MS-DOS (or in an MS-DOS emulator like DOSBox.

AWINDLO offers 2 modes of operation for producing "titled" LGR and DLGR output.

AWINDLO Text-Only LGR, DLGR and BMP Output

When a text file name with the extension TXT or ASC is passed to AWINDLO as an input file, AWINDLO creates a BMP and native Apple II files in either DLGR or LGR format using the first 6 lines of the text file.

AWINDLO Titled "Overlay" Command – Command Line Option T

When an AppleWin Screen Capture file name, or a Native Apple II DLGR or LGR file name, is passed to AWINDLO as an input file, if the command option T follows the input filename on the command line, AWINDLO tries to open a text file of the same "base name". If such a file exists, AWINDLO uses the first 6 lines to overlay the input file with text during "conversion".

AWINDLO Text Row "ROWver-ride" – Commands that begin with R

The "BIG PICTURE" - Think Globally, Act "RowBowLy"

Any of AWINDLO's Font Control Commands (commands that begin with F) are in effect for all 6 rows of text. The F (font) commands are listed further in this tutorial. The R command over-rides the defaults or the F command if any and for one row of text.

<u>Row by Row ("RowBowCop") Control – Commands R0 to R5</u>

Any Font Command (commands that begin with F) is also a Row Command with the prefix F removed and the prefix R followed by the row number added; i.e. R0 to R5.

Rows are numbered 0-5 (top to bottom) which also correlates to the lines in your AWINDLO 6-line text file.

If you want to control your font on a row by row basis, instead of using an F command use an R command followed by the row number. For example, the following row commands are in effect for row 0 only… if any F commands are set for the other rows, they will be over-ridden for row 0 only, and any defaults will be over-ridden for row 0 only, if F commands are not set:

Row 0 Command Example

R0FC0=FC0=Font Color Black (default is white)
R0BC15=FB15= Font Background White (default is black)
R06=F6=6 Pitch Font Spacing
R0F=FF=Fixed Pitch
R0L=FL=Left Justified Row

Please see the F commands further in this tutorial for the details of the R commands. The F command details should tell you everything you need for individual text row control.

But if you do not need row-level control and if an F command will do the job, do not multiply your commands by 6 by using R commands instead of F commands.

<u>Turning FF (Font-Fix) off using R0U to R6U (Row-Unfix)</u>

There was bound to be a snag in AWINDLO's logic. When fixed-font spacing is in effect at the "screen level" and the row wants to save some space by plotting thinner letters and spaces 1 pixel smaller, the row must turn-off fixed-pitch at the "row level" or run the risk of running out of space and truncating the right edge..

<u>Row Background Color – Always in Effect</u>

The behaviour of the R command version row-level font background color differs from the F command version. When over-laying an input file with text (using the T option) the font background is ignored. But when setting a row-level background color the area behind the text (and only the area behind the text) will be set to the color selected.

If you want to blank more of the row, or the entire row, use leading and trailing spaces to "pad out" in your text file. If you want to blank the previous and next row pad out an entire line (based on your font pitch) but 16 blanks will always do.

Default Font Colors – White Letters on a Black Background

My decision to use default White Letters on Black is a consideration for readability on the Apple II display which, after all, is the only thing that is important to AWINDLO.

Somewhere in my distant past the glare of a light back-ground with black text on a monitor screen left my eyes so traumatized that I wore dark glasses day and night for about a decade, and I never forgot stumbling around and seeing only red and black spots and even green and amber spots, and grabbing a coffee and doing a compile while waiting for the attack of blindness to pass. It was for very good reason that we called black on white "reverse-video", and a desktop was where you put your ashtray and hoped you didn't start a tractor-feed paper fire or get too many coffee stains or too much ribbon carbon on the stacks of dead trees and confetti that seemed to spring-up everywhere.

If you want a default white background with black text, write your own program!

Colored Border Option – Commands B0 to B15

This command draws a colored border around the screen.

In order to use border color commands, you need to know the LGR/DLGR "color numbers". Colored Borders can be applied to any input file, whether you are overlaying with text or not.

If you are overlaying with text (using the T option) the border will be applied to the input file first, and the text will be overlaid last of all.

If your text runs right out to the edges of the screen and you wish to put a border on, consider using a different color or the text will not be readable in the portion that the text shares with the border. All LGR and DLGR needs this type of special attention because color is used to replace detail somewhat, and poor color composition between too many objects will cause the screen to blotch with varying degrees of colored chunks that are difficult to decipher.

BMP Output File Disable Palette Option – Command P

By default, AWINDLO provides a "color bar" palette. The top of the AWINDLO BMP is the LGR or DLGR image and the bottom contains a "color bar palette" of the 16 available colors for editing. The "eyedropper" tool in Windows Paint can then be used to set the "draw color", "fill color", "eraser color", etc, without "mucking about" entering RGB values. You cannot just use any color you please in a BMP image and expect it to map properly to the Apple II. You must do any editing with the 16 colors supported by AWINDLO and BMP2LO if you want to properly convert to the Apple II after editing.

But the color bar palette uses disk space, and you don't need it to convert AWINDLO's BMPs to Apple II files using BMP2LO.exe.

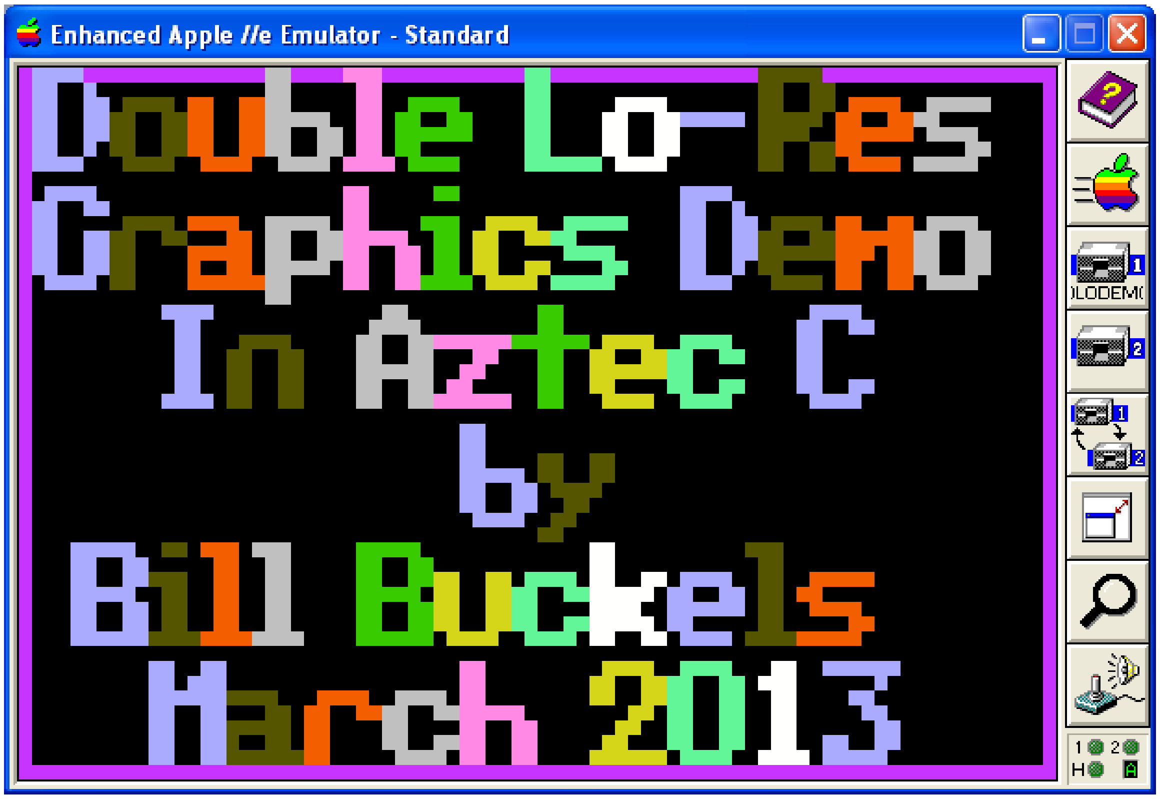
To turn off the color bar palette enter P on the command line following your input file name.

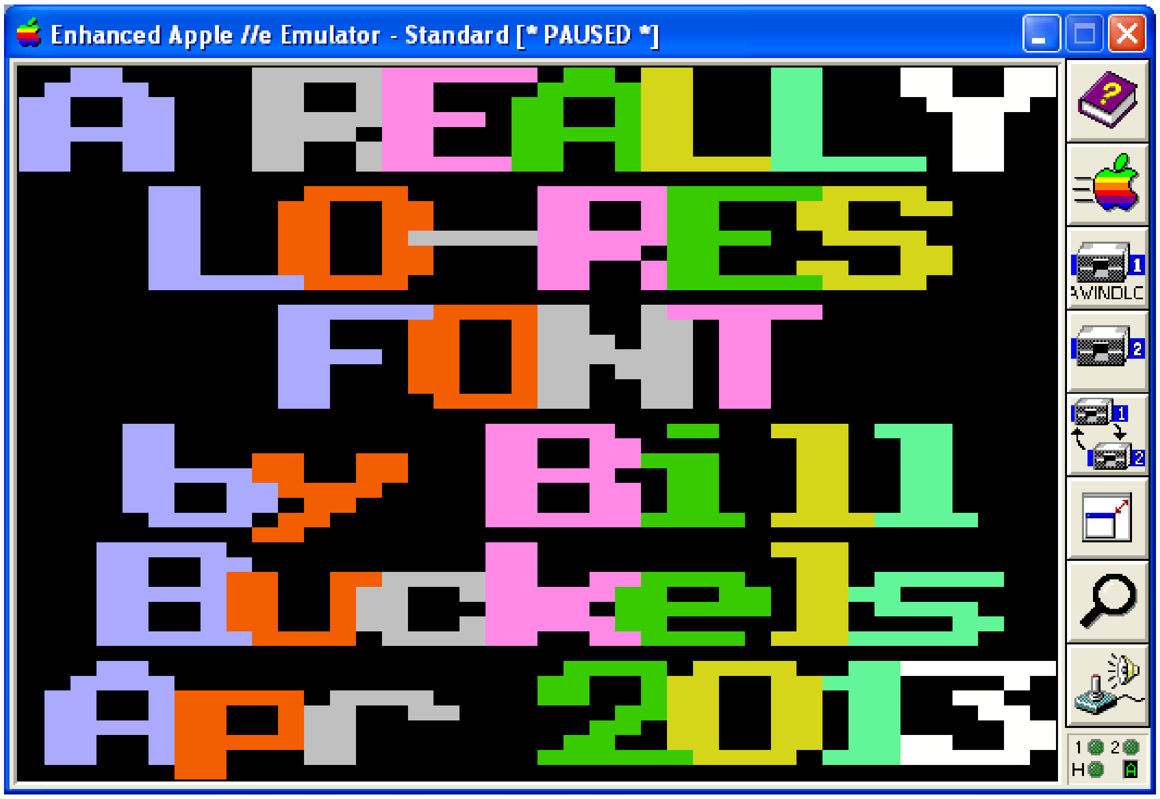
BMP Output File Size Option – Command S

By default, AWINDLO creates a double-scaled BMP for DLGR. When you are editing a double-scaled BMP, produced by AWINDLO, you need to consider the size of your Apple II pixels relative to your BMP pixels. For DLGR they are 4 BMP pixels (two high and two wide). For LGR Apple II pixels are two BMP pixels high and a whopping 4 wide.

AppleWin BMP pixels are enormous when compared to AWINDLO BMP pixels, but you may wish to scale-down your output BMP even further, especially for LGR. A scaled-down AWINDLO LGR BMP is still 2 BMP pixels wide and but a scaled-down AWINDLO DLGR BMP has the same number of pixels as its Apple II counterpart.

If you are happy working with a very tiny BMP or if you wish to save some disk space (scaled-down AWINDLO BMPs take only 25% of the disk space compared to the default double-scaled size), enter S on the command line following your input file name.





Converting to a Single Lo-Res File – Command 1 (number one)

Yes that is right! To directly produce Apple II Files for Single Lo-Res (LGR) mode just enter the SINGLE number 1 as a command line option. 1 is the loneliest number because instead of producing a DLO, and a DL1 and a DL2 file pair, you get an SLO, which is half the size and an SL2 (which corresponds to the DL2 file). In fact when converting from an AppleWin Screen Capture of a Lo-Res screen a DL2 is exactly the same as an SL2 because it is a snapshot of Lo-Res Main Memory Bank 0 in both cases.

AWINDLO Font Control Command Summary – Commands that begin with F

AWINDLO.exe is run from the command prompt. Its command options are entered after the name of the file you wish to process; “AWINDLO my.bmp option option option … “.

Font Control Commands begin with the letter F (for “font”)

Font Color Control Commands

In order to use font color commands, you need to know the LGR/DLGR “color numbers”.

Font Color – Commands FC0 to FC16 (default)

The LGR/DLGR color numbers (color order) are the first 16 font colors, FC0 to FC15.

<u>Font Color Effects Switches - FC16 to FC18</u>

FC16 – "Rainbow" Colored Letters (one color per letter)

A repeating series of 8 brighter colors from left to right. Each letters gets its own color. The first letter is always the same color, the second letter is always the same color, etc. The color assignment is based on letter position in the string and not on the screen. In a "left justified" display with a fixed font pitch with the same starting screen position, the colors give the effect of a vertical stripe.

FC17 – "Rainbow" Horizontal Striped Letters

Rotating horizontal colored stripes; a repeating series of the same 8 brighter colors as FC17. Unlike FC17, "rainbowing" occurs "mid-letter" and is based on vertical position in the screen, not on letter position in the text string.

FC18 - "Candy Cane" Horizontal Striped Letters

FC18 is an alternating "Candy Cane" of 2 horizontal colors that behave like FC17 . Mid-letter "Barber-Poling" is based on vertical position in the screen and not on letter position in the text string (unlike FC16).

<u>Font Color Effects Command Modifiers - FCE0 to FCE15 and FBE0 to FBE15</u>

FC18 ("Candy Cane") defaults to orange and white stripes. FCE0 to FCE15 set the orange to any color (including orange) in the 0-15 color range. FBE0 to FBE15 set the white to any color (including white) in the 0-15 color range. You could also set both colors to the same color and AWINDLO wouldn't care because it knows that you know best, like any obedient programmer's utility. AWINDLO would never behave like a Windows program and stop a user from doing something that doesn't make any sense at all. If a user wanted to blow his foot off on the command line, AWINDLO would load the gun if I included a foot removal option. A toe modifier might be useful too.

I currently have no plans for extending AWINDLO's color effects, but it makes sense to have a couple of spare colors kicking around (and maybe a foot and a toe or two), in case I do an outline or a shadow font, or an underline font someday, and I need some color modifiers for something more than a letter that looks like a barber-pole. A guy can never have too much syntax in a command line utility, or too many utilities either. Just ask any "unix" programmer… the guy with the foot-gun that shoots the silver bullets.

<u>Font Background (Color) – Commands FB0 (default) to FB15</u>

The LGR/DLGR color numbers (color order) set the font background colors FB0 to FB15 when not overlaying an image with titling. When overlaying an image with titling, background color is ignored.

Font Pitch (Character Spacing) Control Commands

Fixed-Pitch Font Over-Ride – Command FF

The default is a variable pitched font, which takes-up less space for thinner characters and spaces. Sometimes you may need to line-up text in columns (like when creating a color chart or something similar).

Font Pitch – Commands F5 to F7 (default)

This is the nominal maximum width in pixels (5, 6, or 7) that each character in a font will use. The default is 7. If the font is plotted using variable pitch, spaces and thinner characters will use one pixel less than the nominal pitch. The font is not truncated by this command but if a small pitch like 5 is used, wider characters will slightly overlap the previous character.

Font Pitch Rationale

AWINDLO's default horizontal pitch is 7 pixels per character cell for wide characters and 6 pixels for narrow characters and spaces:

- 12 wide characters = 84 pixels so the right side will clip (lose) 4 pixels.
- 13 narrow characters = 78 pixels so the right and left size will be offset into the screen by 1 pixel

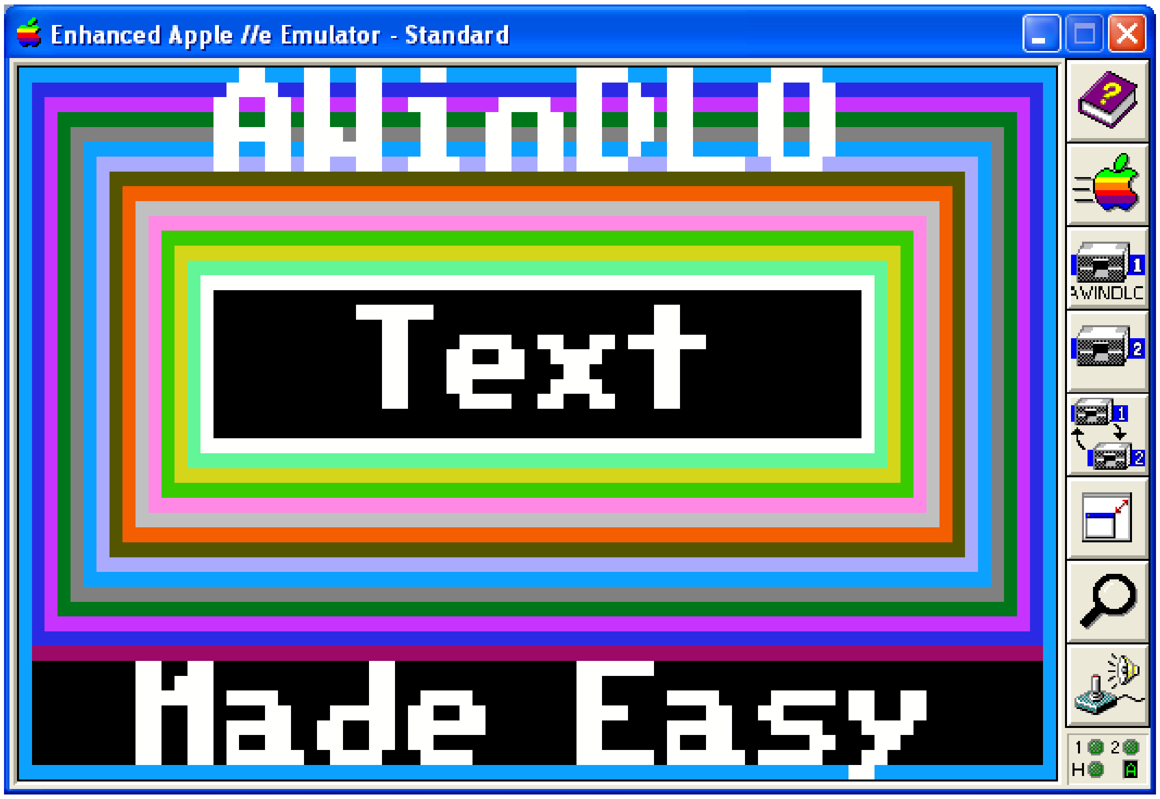
AWINLO's optional horizontal pitch, of 6 pixels per character cell for wide characters, and 5 pixels for narrow characters and spaces:

- 13 wide characters = 78 pixels so the right and left side will be offset into the screen by 1 pixel
- 16 narrow characters = 80 pixels – so no clipping would occur if all characters are thin or spaces.

Command Examples for AWINDLO - Creating Output from Text Files

The AWINDLO and BMP2LO distribution is not trivial. It comes with disk images and working Apple II programs of everything you see here and much more. What you see here is but the tip of a very large continent called the Apple X Aztec C65 distribution.

Now that we have returned from the commercial, before I forget, the text file scripts that were used to create the texted images you see here all contain the AWINDLO commands that I used to create them in the "comment area" below the 6 lines of text, and come with all this continental Aztec C65 stuff included with AWINDLO.exe.



Font Position Control Commands

When we speak of justification we are generally referring to how an individual text line is positioned around a 2 axis datum point which is described as x,y. To be accurate, AWINDLO's "justification" only describes x. Y is fixed at mid-screen, and doesn't care about x. The aggregate of 1-6 text lines expands vertically in both directions from midscreen, in increments of 4 screen lines up and down in either direction for each line in the aggregate "letter cluster", but only if your text file is limited to "active lines" and empty lines do not follow... in other words, less than 6 lines.

X is also somewhat unprincipled. By default x is fixed at mid-screen; "centre justified". Each text line expands horizontally in both directions. When the text hits the left side of the screen it stops and becomes "left justified".

In other words, AWINDLO uses the term loosely. AWINDLO stops shifting the text to the left when it reaches the edge of the screen. It stops shifting upwards when the screen is full of text. Text lines that are too long clip to the right, simply because it is better to see the start of a text line than to be clipped from both ends. AWINDLO does not believe in word-wrap either, unlike Windows Paint. And unlike Windows paint AWINDLO does not stop texting until it runs out of screen, and does not make its screen larger to fit something or other. AWINDLO knows its limits… WOZ defined them long ago.

<u>Horizontal Justification – Commands FL and FM (default)</u>

Left Justified – Command FL

The letters will start on the left of the screen.

Mid-Screen Justified - Command FM

The letters will expand outwards horizontally from the center of the screen equally in both directions. If there are too many letters for the screen, the letters will print from the left of the screen until there is no more room.

<u>Vertical and Automatic Justification – Based on Text File</u>

The text file layout is also used to position the text (both horizontally and vertically).

Vertical Spacing of Text Lines

AWINDLO counts the lines in the text file to a maximum of 6. The vertical spacing is based on 48 logical screenlines . The text file can hold up to 6 lines. Each text line represents 8 logical screenlines.

Empty lines (in the text file) move the logical “cursor” down, by 8 logical lines. If you want your text to start at the top you must create a text file with 6 lines, even if some are empty. AWINDLO doesn’t care how many comment lines you put into a text file below the top 6 lines. You can also put your comments outside the text file “view port” at the end of any text line you please. Just don’t make your lines too long. AWINDLO’s maximum line length is 128 characters in the top 6 lines, and results will be unpredictable if you go over that.

Putting-in less than 6 lines in will cause the "letter cluster" display to be centered evenly in the vertical axis, centred as well as by default in the horizontal axis, provided some of your text lines aren’t too long.

Horizontal Character Spacing in Text Lines

Leaving blanks (padding) to the left will move the first non-space character to the right by 6 or 5 or 4 pixels per blank (depending on 7 or 6 or 5 pitch). Leaving trailing blanks on the lines of the text file will have the opposite effect and push the text to the left.

Combining this method with the FL command option (left justify) allows you to manually center some lines and to left justify others (the R0L to R5L commands will do this without mucking with the text file) . So does combining this method with the FM default and requires less typing at the command line. Yes, I did say that!

Tab-Stops – Fixed at 8

No option at this time. Make your text file in a clean editor or set your tabs to 8. Tabs are expanded with spaces internally by AWINDLO prior to plotting text. They are not lazily collapsed so you should get the text you want if you are using a standard tab-size of 8 and a standard Ascii character 9 as a tab control character.

Text File Line-Feeds – MS-DOS or "Unix" Only

AWINDLO accepts standard text files from MS-DOS/Windows. It also accepts text files from Unix-like systems which use only an Ascii 10 line feed instead of the MS-DOS Ascii 13, Ascii 10 carriage return-line feed pairs at the end of each text file line.

Text files from Apple II-like systems are not accepted. AWINDLO will not likely blow-up if you feed it an Apple II text file with carriage returns only as an end-of-line (EOL) character but the results may be pretty ugly. I have left some of these things for the user to try for himself being the generous guy that I am. Hard disks are cheap these days.

Text File Characters Below Ascii 32 –"Blanks"

AWINDLO does not support fancy stuff like "soft returns" or "page feeds" in a text file that use Ascii device (terminal, printer) control values like Ascii 14 or Ascii 12. These will be printed as Blanks (with the exceptions of carriage returns Ascii 13, line-feeds Ascii 14, and tabs Ascii 9) which are discussed above.

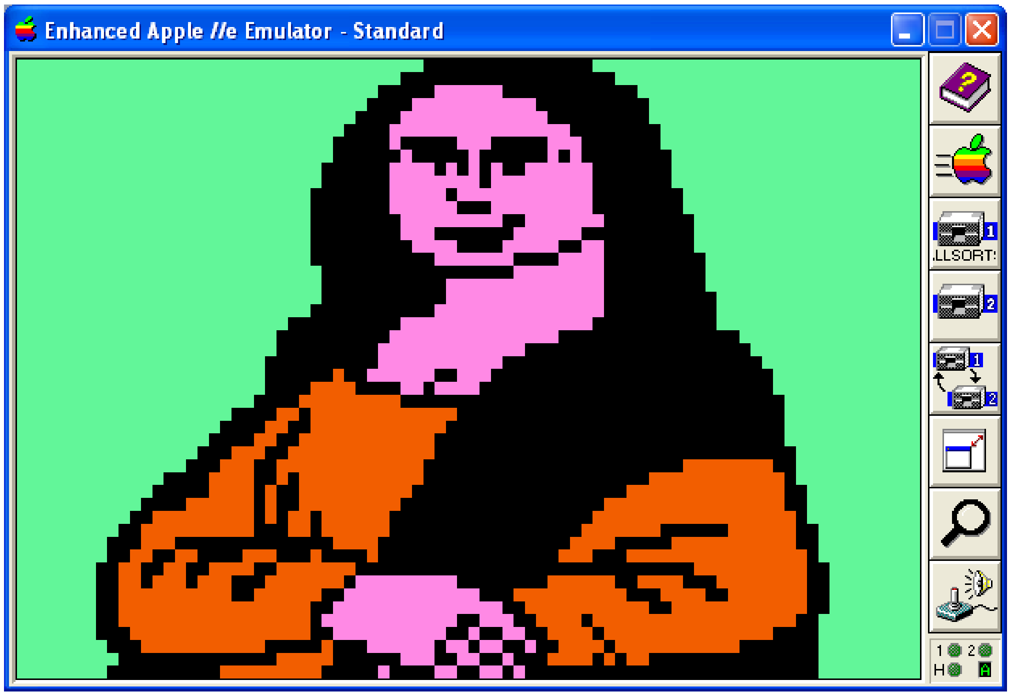
"Hi Ascii"Text File Characters – Characters above 127

As previously discussed, AWINDLO's Font goes-up to Ascii 255.

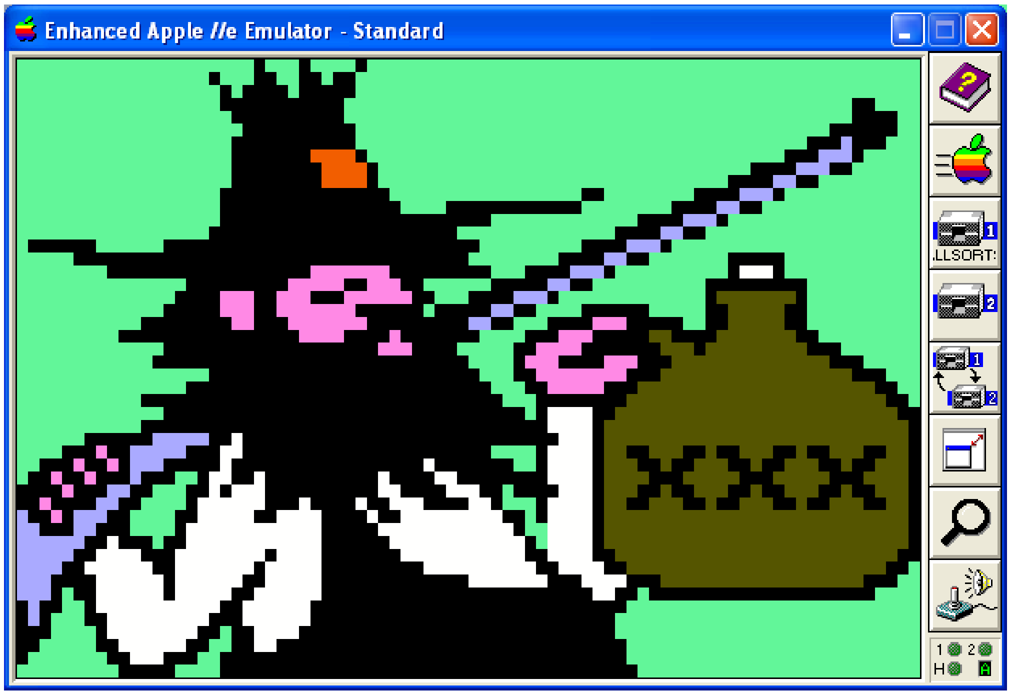
It is unlikely that your Windows Editor supports correct conversion to MS-DOS text. Even Windows editors like TextPad, with a Save As DOS text feature, have problems. When you start editing a line of MS-DOS high Ascii "graphics" characters in TextPad, the "hi-bits" become displayed as their 7 bit equivalents and the original values become unrecoverable.

Use an MS-DOS editor like QEdit that saves MS-DOS ASCII text in the format that AWINDLO uses. The appearance of a Windows multi-lingual character in an ANSI editor does not mean it will map correctly in AWINDLO. If you already have a wonderful editor, then ignore the preceding recommendation and have a wonderful life. If not get a wonderful editor (optional), but have a wonderful life regardless.

Data Sources for DLGR and LGR Pictures

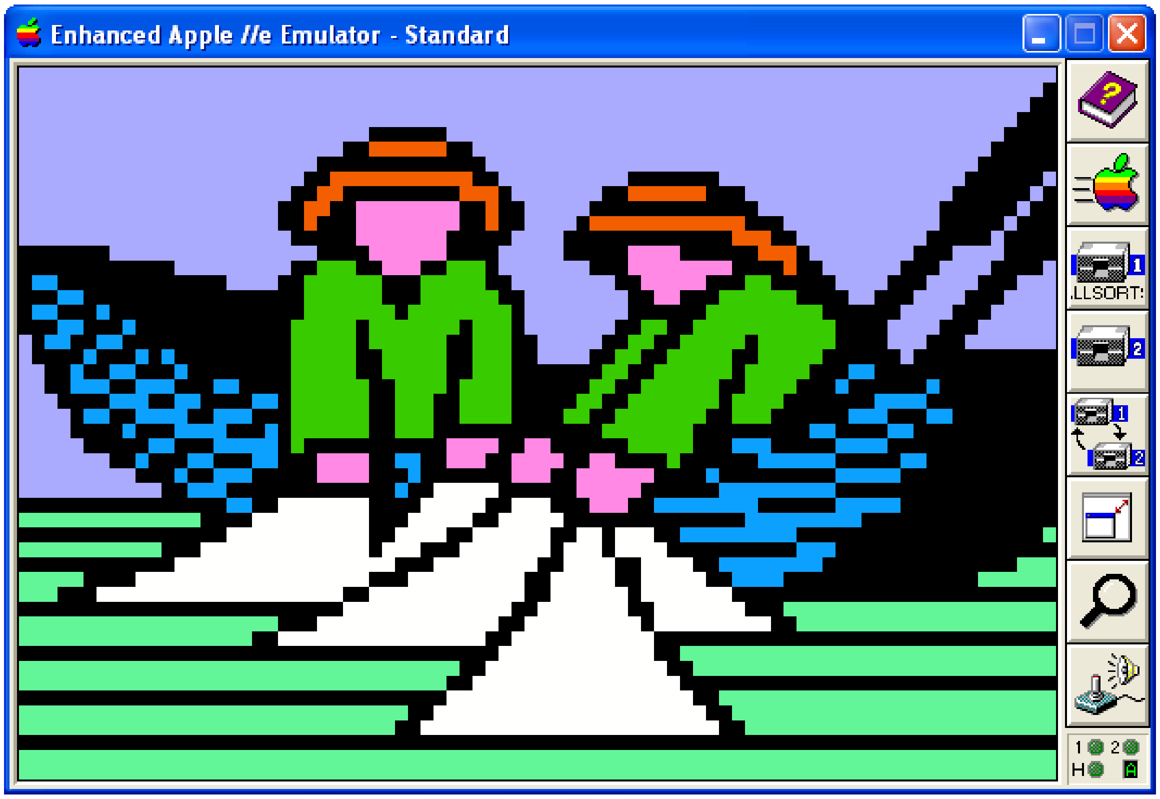


You can use Windows Paint to Create BMP Pictures for conversion to Apple II DLGR and LGR Images and then use BMP2LO.exe to convert your pictures to native Apple II DLGR and LGR files then pass them back to AWINDLO for titling if needed.



After conversion these files can be titled in AWINDLO.exe using the methods described in this tutorial, or used as-is if you don't need to embellish them further. You are now an expert in using AWINDLO.exe to create titled DLGR and LGR output files, so I won't be saying much more about its use in this section.

<u>Using BMP2LO.exe to convert Windows BMP's to LGR and DLGR Files</u>



BMP2LO.exe Version 3.0 has been much improved and expanded-on.

BMP2LO.exe has two "modes" of conversion to LGR and DLGR files:

- Clipping Editor – Interactive – Single and Double Lo-Res Output. MS-DOS 4-color BAS and PCX files and 16 – Color BMPs only. Note: the Clipping Editor can be (or could be) called from the command line and uses the same command line options as the command line only mode (see below), but launches into the clipping editor's MCGA graphics mode where you can clip from an input file of up to 9 individual graphics per input file and remap colors.

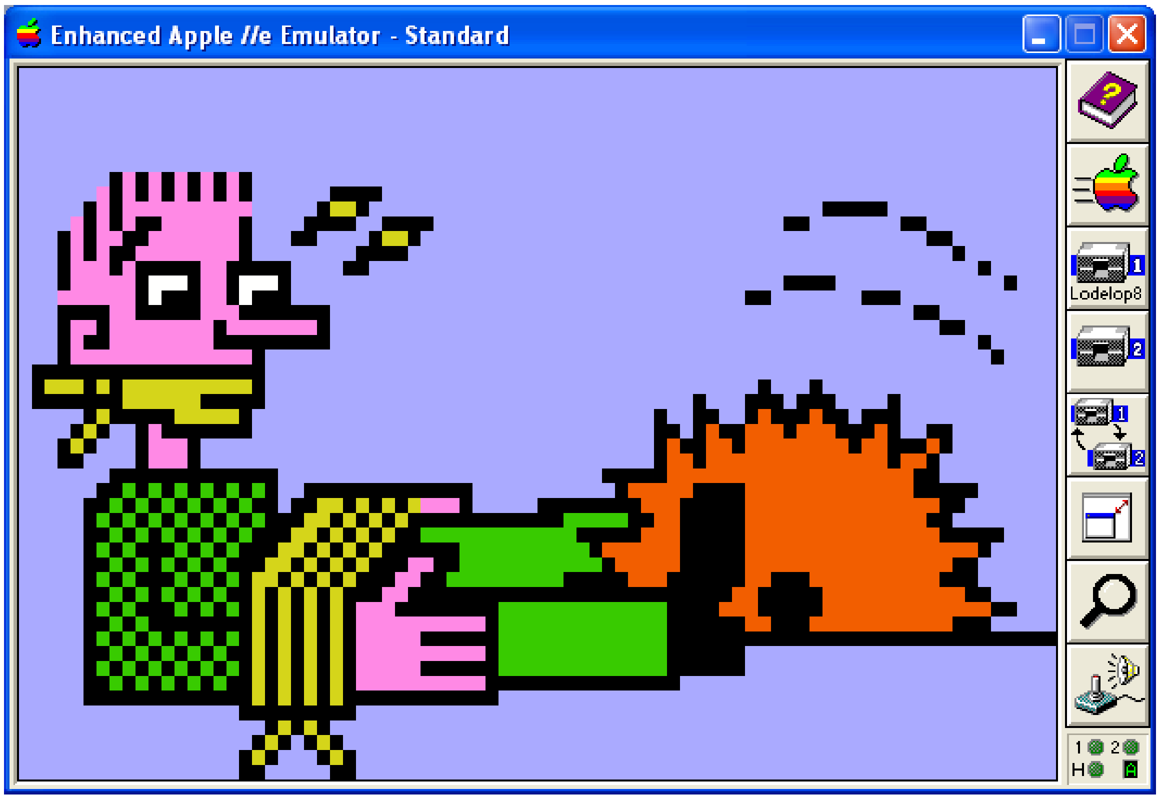
- Command Line Only - Non-Interactive – Single and Double Lo-Res Output. 24 – bit BMPs only. This mode is also "text mode" only. On systems that do not support MS-DOS graphics applications, this is your only option, besides running within an MS-DOS emulator like DOSBox that supports MCGA video mode.

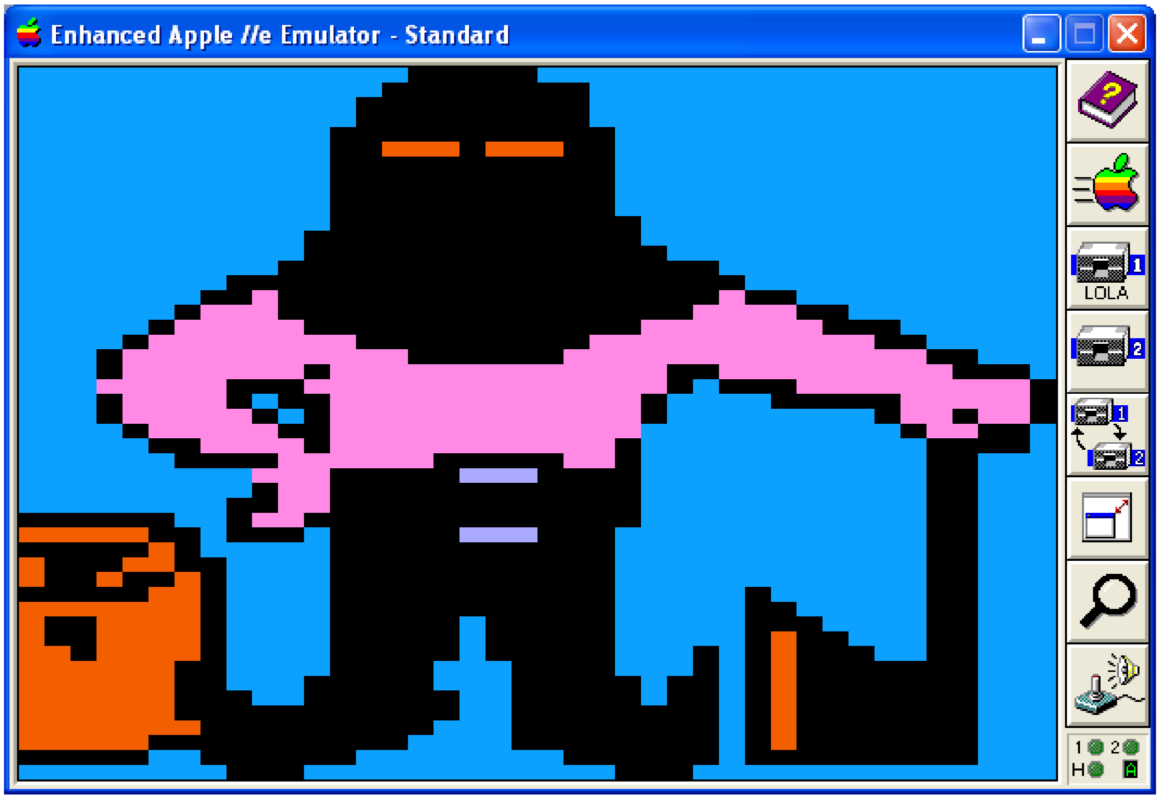
Interactive conversion is designed for "size-on" conversion of sheets of up to 9 "MiniPix" (88 x 52 Old Printshop Graphics) created by clipboard pastes from my ClipShop utility into a Windows Paint 16 Color BMP file. Another utility called Mini2BMP.exe is also distributed with BMP2LO for converting native Apple II Minipix to 16 color bmp files suitable for use with BMP2LO's interactive conversion mode.

Command Line only conversion is designed for conversion of the 24 – bit BMP's produced by AWINDLO and other 24-bit BMP's in a fixed-size range. See below:

BMP Image Formats accepted by BMP2LO version 3.0

Non-Interactive Conversion	
24Bit.BMP	Sizes (resolution) below
AWINDLO or similar	160 x 96 - 160 x 128 - 80 x 48 -80 x 64
MINIPIX single file	88 x 52 or 176 x 104
Interactive Conversion	320 x 200 resolution
MINIPIX 9 clip maximum	CGA 4- Color GWBASIC BSaved Images
MINIPIX 9 clip maximum	CGA 4- Color PCX Images
Interactive Conversion	maximum 320 x 200 resolution
MINIPIX 9 Clip maximum MINIPIX single file	16 Color Windows Paint BMP Files of any size up to 320 x 200





BMP2LO LGR "Lo-Res" Output – Command Line Option L

BMP2LO produces LGR output as well as DLGR output. When converting a BMP to Lo-Res Apple II files you must enter L (for "lo-res") on the command line somewhere following the BMP input file name.

BMP2LO "Mixed Text and Graphics" - Command Line Option T

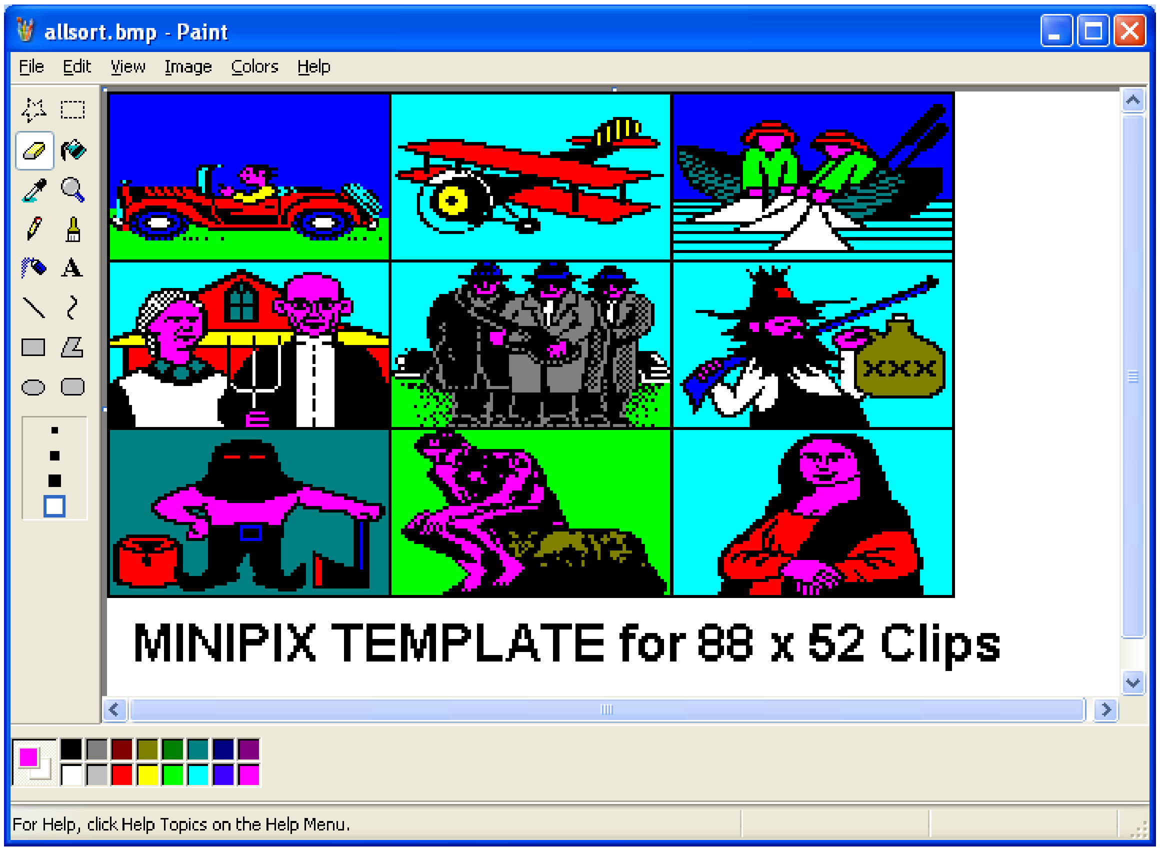
BMP2LO produces "Mixed Text and Graphics" LGR or DLGR Apple II raster files of 40 scan-lines in height rather than the default "Full Graphics" files of 48 scan-lines in height when the T (Top) option is used. The "lo-res" file has the extension STO ("single top") and the DLGR file has the extension DTO ("double top"). They use less disk space.

BSAVED Apple II output files are not supported for this feature because it would not make sense… the Apple II LGR/DLGR and text screen memory is a shared area, and is is interleaved and not contiguous, and the memory for the bottom 4 text lines is mixed between chunks of DLGR graphics, so you get a mess when you BLOAD. Raster oriented files are selective about where they are loaded and the mess does not occur.

BMP2LO Output File BaseName Option

Most of my utilities use the basename of the input file for the output file. This saves on typing and isn't generally a problem. But BMP2LO's interactive mode can have up to 9 – 80 x 48 graphics in a 320 x 200 input file… so BMP2LO supports using a different basename to accommodate files with more than one graphic. By default, if no output file basename follows the input file name on the command line, BMP2LO will just use the basename of the input file for the output file names. Remember MS-DOS 8 character basenames only!

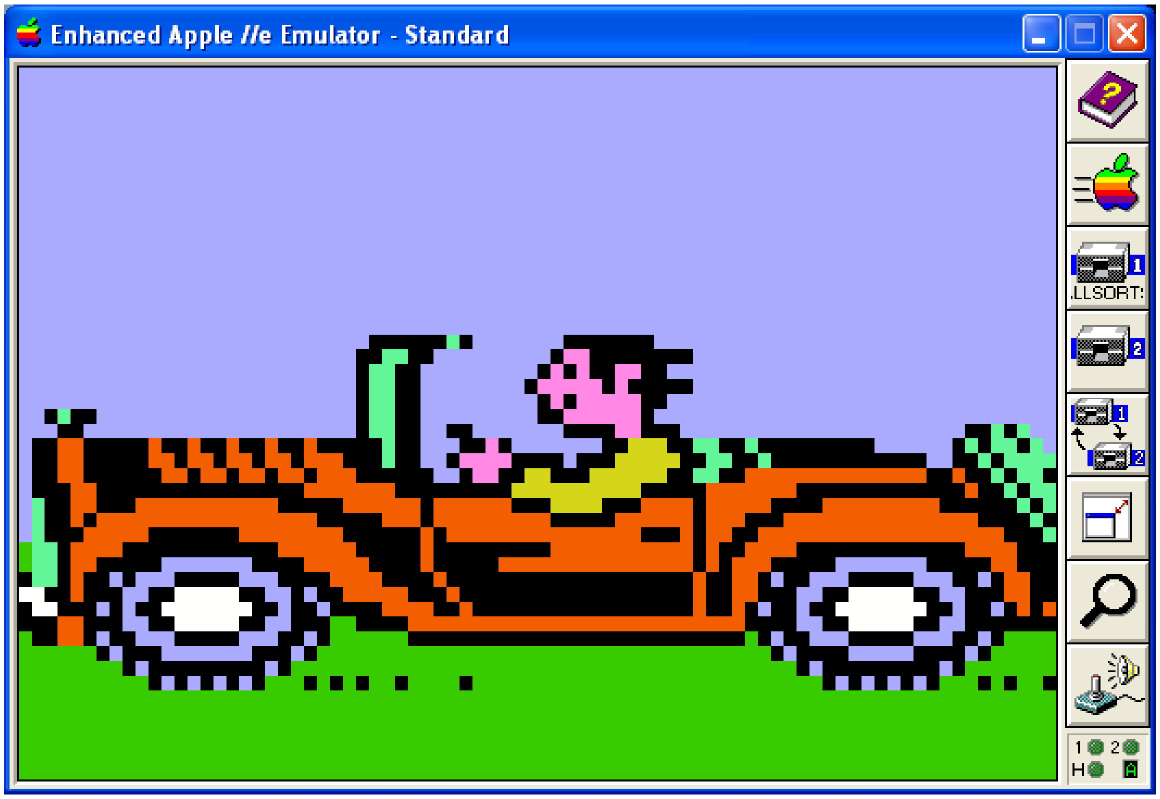
Windows Paint "Roll Your Own" - Creating and Aquiring PC Graphics


allsort.bmp - Paint
File Edit View Image Colors Help
XXX
MINIPIX TEMPLATE for 88 x 52 Clips
For Help, click Help Topics on the Help Menu.

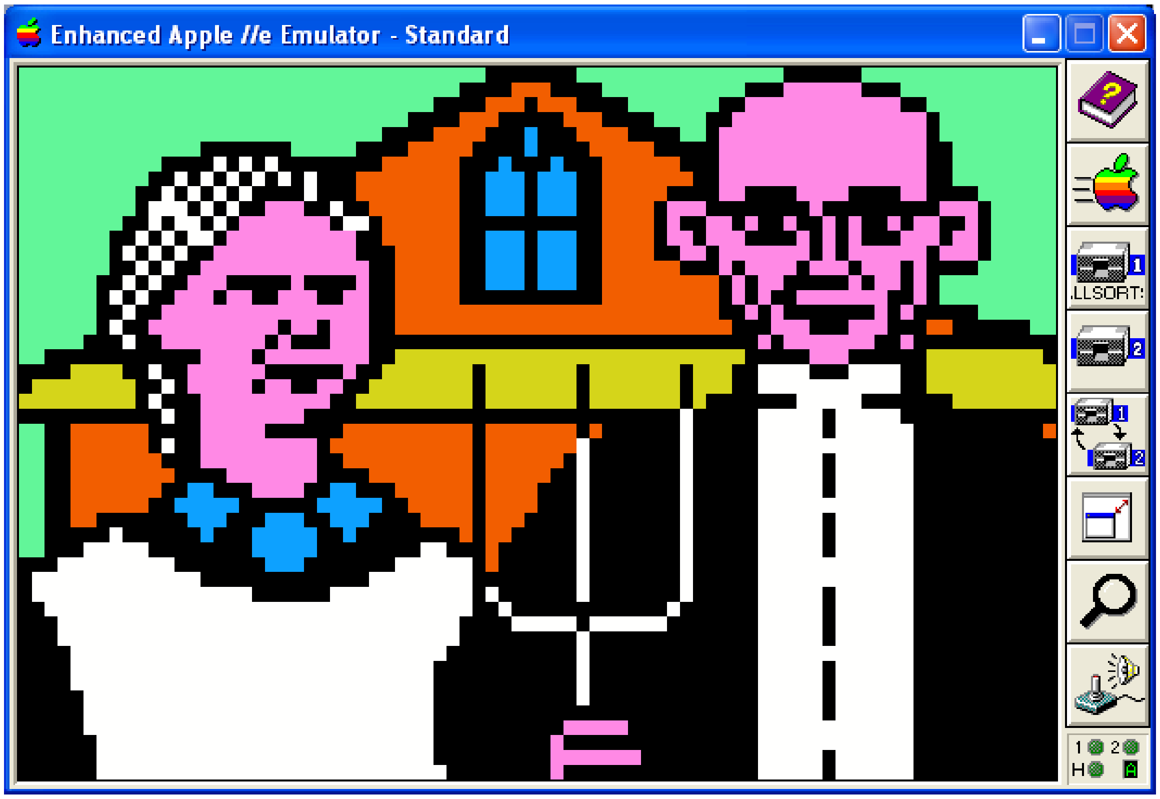
The image above was "hand built" by pasting-in IBM-PC Minipix from my [Clipshop](#) utility into the 16 color BMP template provided for you , and coloring them. I then converted these images to Apple II DLGR files using BMP2LO and built the DLGR slideshow which is on one of the Apple II disks that comes with AWINDLO and BMP2LO.

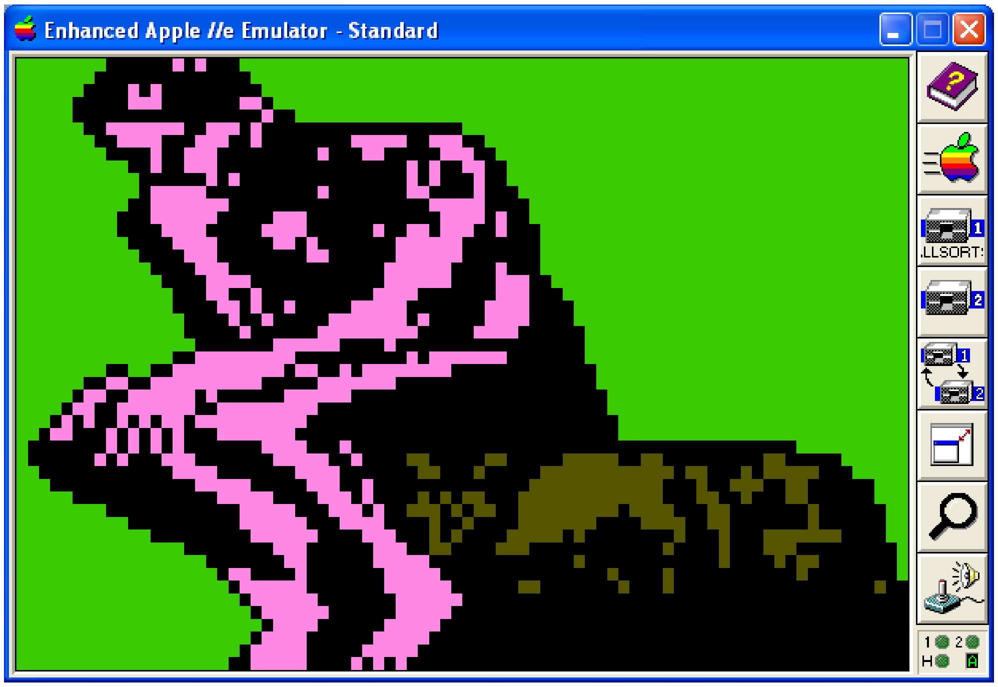
I could also have saved these as individual 16 color BMPs or as individual 24-bit BMPs and BMP2LO would have merrily converted them. If I saved them as 24 bit BMPs I would have needed to fuss with setting exact colors from the supported palette. BMP2LO is set-up to map "similar" standard Windows colors to their LGR or DLGR equivalent as well as supporting AppleWin colors.

But ClipShop doesn't know about AppleWin colors and I bet neither do a lot of others, so it is quickest and easiest to just paste these clips into a 16 color BMP in Windows Paint, and use BMP2LO's 16 color mapping from standard Windows colors.



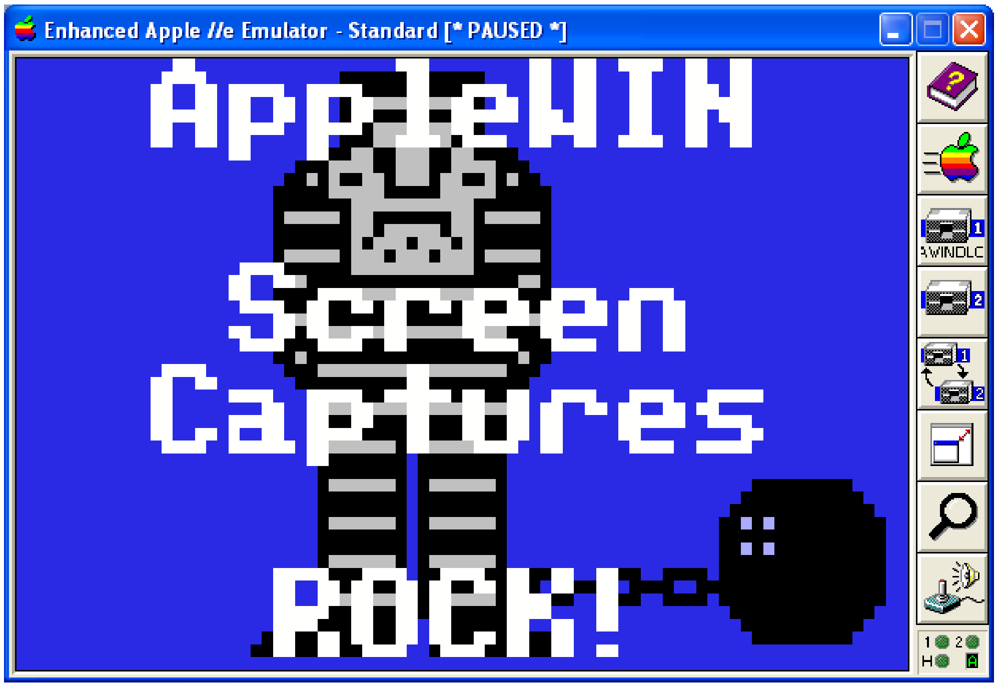
A Double Lo-Res Rider driving home while Ma and Paw Apple (below) wait for Sonny.





Whether you work with MiniPix or create your graphics from scratch, if you are comfortable using 16 color BMP's and work with an unscaled image, you should always get good results from AWINDLO for LGR or DLGR output. For LGR output you must always edit in horizontal pixel-pairs on the smaller size input file and in 4 x 2 "pixel blocks" on the larger size input file.

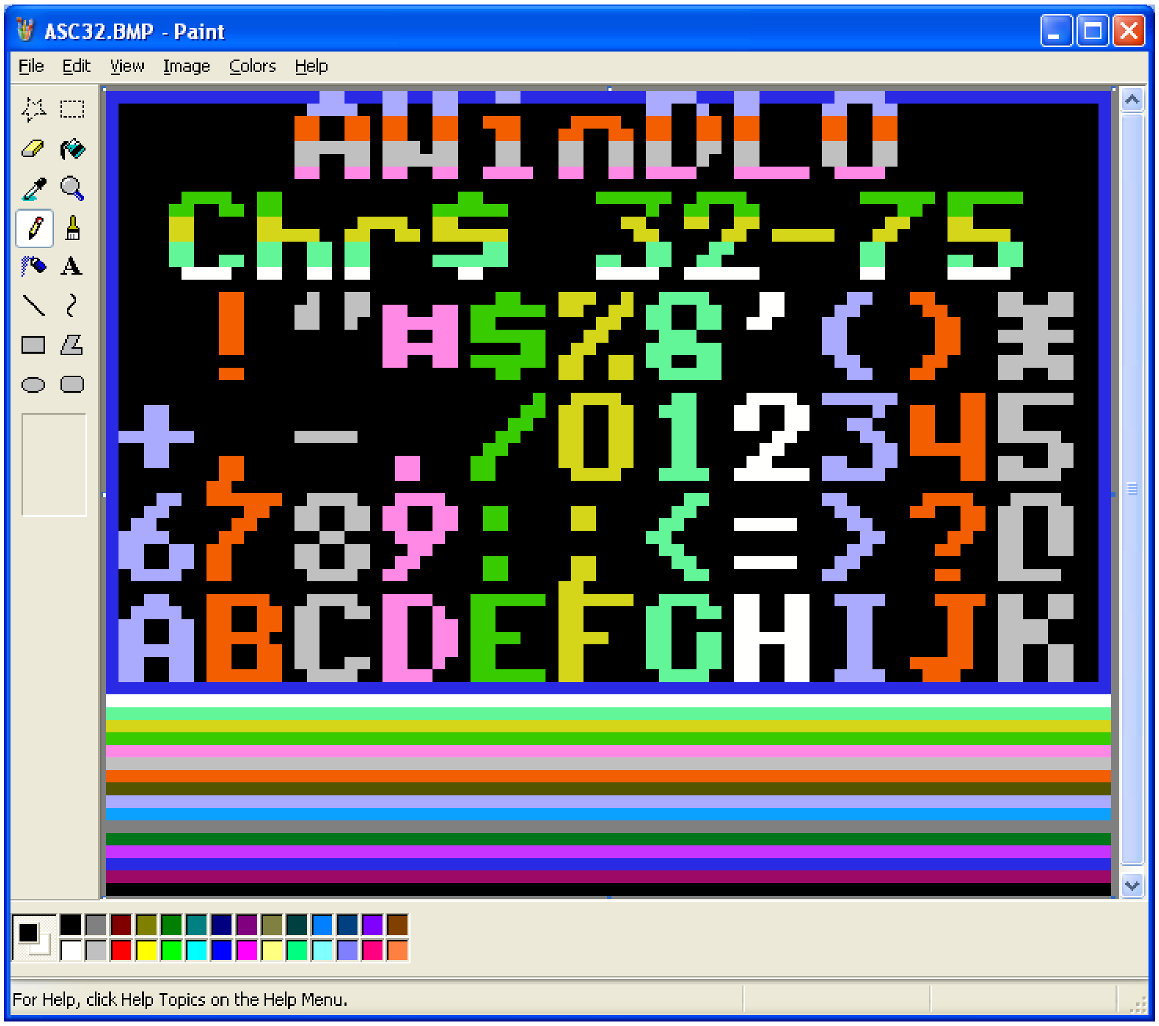
DLGR and LGR Screen Captures In AppleWin



1. The **AppleWin Emulator** can be used to capture Color (standard) mode DLGR and LGR screens from a running Apple II program and save to a Windows BMP file. AppleWin has "hot keys" built-in to do this. The [Print Screen] "hot key" captures to a 560 x 384 x 256 Color BMP. This particular size is symmetrically "laid out" in 7 x 8 pixel (color) cells for DLGR, and 14 x 8 for LGR, which is perfect for conversion back to native Apple II DLGR Screen Files.

2. These color screen captures are enormous compared to the "coarse" resolution of the 80 x 48 x 16 Color DLGR screen or the 40 x 48 LGR screen. When AWINDLO converts an AppleWin screen capture to native Apple II DLGR or LGR files, it replaces the AppleWin screen capture file with a smaller 24 bit BMP suitable for editing in the same colors as AppleWin Version 1.22.

The top of the AWINDLO BMP is the 160 x 96 DHGR image and the bottom contains a "color bar palette" of the 16 available colors for editing. The "eyedropper" tool in Windows Paint can then be used to set the "draw color", without "mucking about" entering RGB values.


ASC32.BMP - Paint
File Edit View Image Colors Help
AWINDLO
Chrs 32-75
!"#$%&'()*
+,-./012345
6789:;<=>?@
ABCDEFGHIJK
For Help, click Help Topics on the Help Menu.

3. After editing, BMP2LO converts your finished work directly back to native Apple II DLGR or LGR screen files, which can either be titled in AWINDLO and then re-converted ad-infinitum, or used as-is.

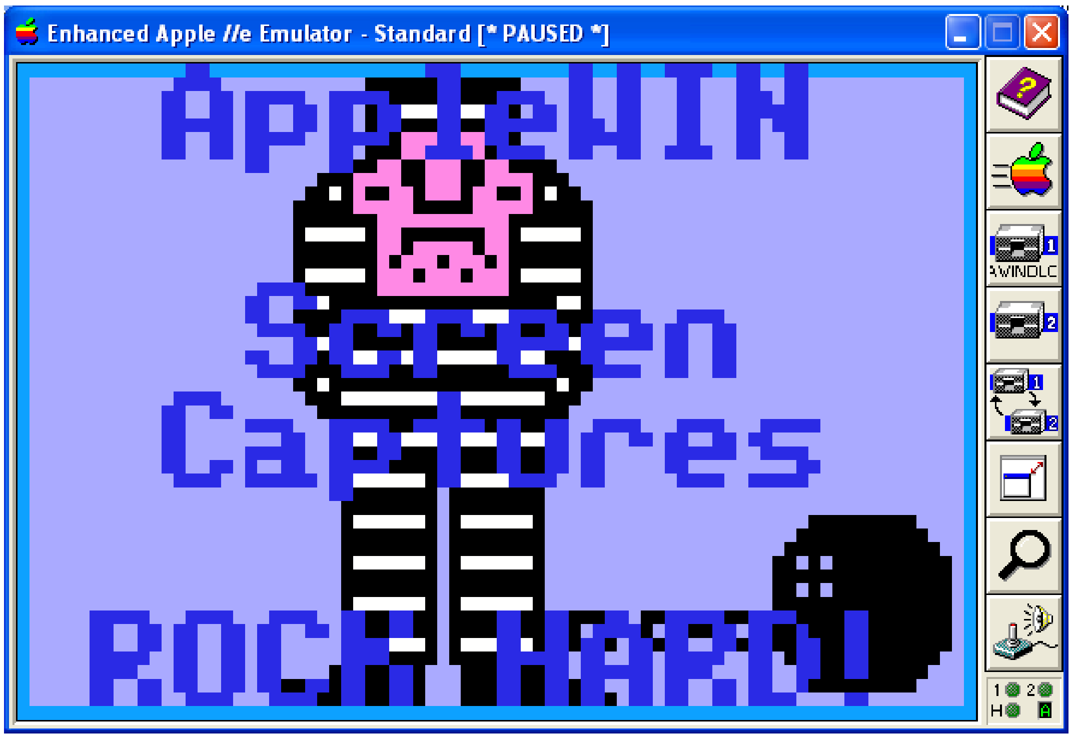
   When converting back to Apple II DLGR or LGR files from the BMP with "color bar palette" using BMP2LO, the color bar palette area (at the bottom) is ignored and left alone as a color reference for future editing.

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).

Please consult the documentation for AppleWin, Windows Paint, CiderPress, and the documentation and source code for my own utilities for additional information.

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.


Enhanced Apple //e Emulator - Standard [* PAUSED *]
AppleWIN
Screen
Captures
ROCK HARD!
AWINDLO