

Black Belt Systems AVT Master  
Update Documentation for Beta Release Version v3r8

Posted April 20th, 1988 by Ben Williams, AC4BS

This software is Copyright 1990 Black Belt Systems and ADA incorporated.  
ALL RIGHTS RESERVED

Under no circumstances is it to be redistributed without express written permission from AEA or Black Belt Systems.

Note to the users of the AVT system from AA7AS:

\*\*\*\*\*

The AVT system was extensively tested, and for quite a long period. Unfortunately, due to the complexity of the system, there were a number of "bugs" or problems that have gotten into the release versions. Everyone, myself, the Beta Testers, and the crew at AEA worked very hard to avoid this, but... oh well. Now, here's what we are doing about it: First of all, I listen to the hams on 14.233 almost constantly. These people are the most "serious" users of the system, they use every feature they can. When I hear of a problem, I either fix it immediately (if the problem is clear to me) or I get on the air and work with these people until it is clear to me, and then I fix it.

Once we have a stable version that has the most recent bugs corrected, it is posted on CompuServe, in the AmigaVendor forum in Data Library 6. You can log onto CompuServe and download it in a very short time. If you do not have modem capability, you may obtain the upgraded software from another AVT owner who does. Less often, due to cost factors, the software is presented to AEA and they will mail out an upgrade to everyone, either free of charge or for a small fee. Once an upgrade has been obtained in these methods, you are authorized to pass it on to any other AVT user. Do not give the software to non-avt owners, please - it is not to your benefit to do so, and it may hurt AEA or BlackBelt systems in a very direct manner. This software is not public domain!

All of us at BlackBelt systems and AEA hope that this solid support will make up for the fact that the product is not perfect. Very few software products ever are, and most do not receive this level of support, either. You can help us by reporting bugs as soon as you have clearly defined the problem. Please do so in writing, to the following address:

Black Belt Systems  
RR1 Box 4272  
398 Johnson Road  
Glasgow, Montana  
59230

In addition, for those of you who are hams and have bought the AEA version of the system, we would like to see QSL cards or notes from you indicating your level of satisfaction, ideas you might have for the system's improvement, and so on. We maintain a large world map with little pins stuck in it wherever we have identified an AVT ham

operator. It's just one of those "things", and we'd like to have a pin just for you!

\*\*\*\*\*

These are the program and disk changes from the v3.01 release:

1 - We have found in an Amiga 2000 that when a Supra modem is connected to the serial port and has power applied to it, the AVT system will not function correctly. This is because the AVT uses the "printer select" line to communicate with the Amiga, and the printer select line is also (for some odd reason) connected to the serial port chip inside the Amiga 2000. So, if you find that the AVT suddenly won't change I/o ports, or it won't go into transmit, check to see if your modem is connected and powered on. If so, just turn it off - that may well correct the problem. We are in the process of checking with CBM to see why this strange connection between the modem and parallel ports exists.

2 - Two bugs in the "framed grab" option were corrected. The program would not grab using the selected offset, and the offset requester said "256" twice, where in one case 384 was the actual value. Both of these capabilities are fixed.

3 - A bug in the color bar generation was fixed. If the program was in AVT 90, it would not generate the color bars correctly. This has been fixed.

4 - The manual did not document the "INx" ICON and CLI options. These options allow you to set the name in the I/o box for each of the five user defined inputs on the AVT hardware. To use this capability in the ICON, use the ADD gadget in the tooltypes window of the WorkBench menu selection to provide a new, blank line. Once you have the new blank line, enter the name of the button like this:

```
IN1=TS-950
```

You may find it looks better if you include a leading space:

```
IN1= TS-950
```

If you use this option from the CLI, you use it like this:

```
AVT_Master in: TS-950 in2 FT-980
```

and so forth. For more inputs in the ICON, just add multiple (up to five) lines using the ADD gadget in the tooltypes editor.

5 - The ARexx script interface that the AVT system uses has been changed in a significant way. The AVT, and ARexx, will now expect all scripts to end in ".avt" instead of ".rexx".

Scripts that end in .rexx will still function, however you will find it MUCH more convenient to rename any scripts that work

with the AVT from "script.rexx" to "script.avt".

6 - The system archive manager has been completely rewritten. This has been done for two reasons, (1), the new one is more compact and uses less of both CHIP and FAST memory; (2), the user interface needed to be (and has been) improved. The new archive manager has the devices and directories columns in different, more logical locations, and you can now "click" directly upon the devices, directories, and paths you wish to use.

7 - The Pseudo-Visitel mode was broken in the v3.00 and v3.01 releases. It's fixed now.

8 - When you saved IFF from the FAX modes in v3.00 or v3.01, the system only saved the images as 16 grey level images, even if you had used the pseudo coloring capabilities from ARexx. This was unintentional, and has been fixed. Pseudo colored IFF images now save as normal IFF's which can be handled by paint and show programs, complete with bizarre colors of your choice.

9 - There is a new parameter in the ICON and the CLI invocation. This is the "TLENGTH=nn" parameter. It sets the length, in 1/50th of a second increments, of the touch-tone outputs. It is set to a good default for telephone use.

10 - System transmit amplitude is limited during telephone operation by the software. When operating on the phone line, the level adjustments have no effect. This has been done to comply with PCC regulations.

Here are the program and disk changes from the v3.00 release:

- 1 - The program BackShow in the For\_C and Tools directories was broken when used on normal Amiga systems, although it worked fine with 68020 and 68030 processor based Amigas. The program has been fixed for this release.
- 2 - The program NextScream, mentioned in the manual, was omitted from the v3.00 release diskette. It has been included on this release diskette.
- 3 - The values that set the Martin M1 and Martin M2 mode timings for transmit were omitted in the v3.00 release. They have been determined and verified, and are set for you when the program begins execution.
- 4 - The Scotty S2 transmit mode was broken in the v3.00 release. It is fixed in v3.01.
- 5 - When operating in the FAX modes, the autostart and auto-phase capabilities have been changed so that they translate the incoming FAX signal through the demodulation curve before they examine the signal for start and phase information. This means that if you set up the demodulation curve for

inverse, the system will phase and autostart correctly, assuming only that the fax signal was inverted.

6 - Included on the v3.01 release disc is a sample demodulation curve that implements an inverse function. We have found this particularly useful when operating FAX using a TVRO system, and operating the receiver in the FM mode. You seem to get a much cleaner signal than you do when operating sideband as recommended in the AVT manual. The catch here is that the signal is inverted (in this particular case) from it's normal positive/negative orientation, and as a result you can "flip" it by inverting the demodulation curve. If you're operating FAX with a TVRO and you have FM demodulation capability, we suggest you try the inverse curve.

7 - The original "readme" file on the v3.00 release diskette was the readme file for the Beta Testers, not the readme that should have been on the disk. No particular harm was done, as the "real" readme file would simply have indicated there were no substantial changes to the program as regards to what the manual said.

8- More information for users with One Megabyte Amigas:

A - Amigas with one megabyte of memory are unable to operate the AVT 188 mode if it is sent in the normal manner. However, if the transmitting station sends the 188 signal using the QRM mode, and you do not use QRM mode and enter 94 AVT receive mode manually (Don't use the AUTO button, just press the RX BUTTON), you will receive a usable, correctly formatted image. You cannot transmit in the 188 mode until you add more memory to your Amiga.

B - The 1.3 Workbench disk, as shipped by Commodore, is set up in such a way as to use quite a bit of your memory for completely non-essential functions. So much memory is wasted in this fashion that the AVT is unable to obtain the memory it requires to operate correctly. To set up the system such that this memory is available, you'll need to edit your startup-sequence and startupII files in the S directory on your WorkBench boot disk. In addition to the various "extra" items setup by the stock WorkBench disk, we will remove the workbench itself - this allows us to recover the memory that the workbench normally uses. Here's what to do in your startup-sequence and startupII files:

I - Remove any line that has the word addbuffers in it.

II - Remove any line that has the word resident in it.

III - Remove the line that has FF >NIL: in it.

- IV - Remove the line that has SetMap in it.
- V - Remove the lines that have Mount speak:, Mount aux:, and Mount Pipe: in them. (these are in the startupII file)
- VI - Remove the line that has LoadWB in it.
- VII - Remove the line that has EndCLI in it.
- VIII - Add a line at the end of the startup-sequence file that reads: stack 4000
- IX - Remove the line that has Mount NEWCON: in it. This is used only for the shell, and you are running a CLI, not a shell.
- X - Remove the lines that have assign env:, assign t:, and assign clips:
- XI - Remove the lines that have mkdir ram:t, mkdir ram:env, and mkdir ram:clips
- XII - Once the AVT\_Master program has been started, go back to the WorkBench (using left Amiga+M and N) and shrink the CLI window that you used to start the program. This helps when you open the file requester to load the SSTV files.

- C - When your Workbench boots, you will no longer have the "icon" interface available. To run the avt program, you must enter the command in the CLI. We suggest you boot your system using the modified WorkBench, then change to the AVT disk and enter something like this:

```
df0:AVT_Master chip compress=<enter>
```

- D - If you have two (or more) disk drives, you will find that removing them (unplugging them - and ONLY when the power is OFF!) returns approximately 25-30k per drive, which is very useful. We strongly suggest that you use only one drive until you obtain more memory.
- E - You will get better AVT performance if your machine has the "fatter agnus" chip. If you have recently purchased an Amiga 2000, it is likely that you're already have this chip installed. The currently shipping 500's are also supposed (we've not verified this yet) to have the fatter agnus installed. If you have an older 500, 2000, or 2500, you can get the fatter agnus installed for (usually) about \$130.00 at your dealer. We very strongly recommend that you do this, especially if you plan to stay at the one megabyte memory level!
- F - Running the NextScreen program, while it certainly makes

things much more convenient, also uses a small amount of memory. In a one megabyte machine, it's better to manipulate the screens manually so as to save memory used by the NextScreen program when it's installed. The solution, of course, is to get some memory.

- G - You should be aware that adding just another 512K to your machine will help the AVT program to operate in a much more satisfactory, manner. It will operate faster, and the AVT 188 and all FAX modes will become available.
- H - You will be unable to run the ARexx software until you obtain more memory.

9 - New functions:

A - The keyboard in1, in2 in3, in4 and in5 have been added to the CLI startup procedure. These keywords allow you to specify (from the CLI) the names for the five gadgets in the I/O window. In version 3.00, you could only do this if you started the program from the WorkBench ICON. That also meant that starting the program from a "boot" disk, using the startup-sequence precluded you from being able to

name

the I/O gadgets. This was not our intention, and so the new CLI options were added.

B - The keyboard ichan has also been added to enable you to select which of the six input channels will be used when the program starts. Use:

```
ichan <n>
```

Where (N) is a number from 1-6.

C - The Zoom function has been modified to work slightly faster, and it now will operate upon any resolution including the AVT 188 second images.

D - We have added a script in the For\_Rexx directory called "CoBang.rexx". This script generates a series of random color sets, each set having a specific ID number. It is to be used with the FAX modes to "PseudoColor" an image all in one shot. You must have ARexx installed in order to use this script. Copy the script to the rexx: assignment in your system, and then place this in a function key:

```
rexx:cobang.rexx
```

Then you can execute the macro anytime you are in FAX mode by simply pressing the appropriate function key.

E - One button inside the Synthesize window has had it's function completed and assigned to it. This is the inset operation. It is only effective if you have enough ram

to have more than one image memory. The operation will place buffer zero inside the currently selected buffer (other than zero) as a 1/4 frame image anywhere you like. To use it, load memory zero with the image you want to place as an inset 1/4 frame. Now change your selection to another memory, and load it with the image you want to place the inset image in. Open the synthesize window and press the inset icon (next to the N icon, just to it's right). You are presented with a rectangular 1/4 frame. Move this to the location you want to have the inset image placed at, and click the left mouse button. That's all there is to it. This button appears as a small square inside a larger square. This is intended to convey the idea of the frame inset.

F - The program has added a basic "clip" cut/paste system. This implemented with three new buttons inside the synthesize window. One has the letter "C" on it, the other two both have the letter "P" on them. There is a difference between these, one has the "P" on a blue background and the other has the "P" drawn on a black background. When you operate this function, you first need to "CUT" a clip from an image. This is done by opening up the synthesize window, pressing the "C" button (cut), and then dragging a rectangle around the area in the current image you want to use as a clip. Once this has been done, the clip is defined. To place it in an image, open the synthesize window, then press either of the "P" buttons. The difference between the two is the one with the blue background pastes the clip exactly as you cut it; the one with no colored background will not rewrite image pixels wherever there was pure black in the source clip. This means that you can draw your call letters in a paint program against a black background, grab the image into the AVT, then cut them out. If you use the no-background "P", only the letters themselves will be pasted into the target image - but not any of the black background. You will find (in the SSTV menu) new

functions

to free the memory taken by the clip (may be up to 200k!),  
to load clips from disk, and to save them too.

#### 10- Text generation fixed/updated:

In the v3.00 release, the text generator would lock up the machine if you attempted to use it in the AVT 188 mode. This has been fixed, and in addition the registration of the pre-placement "box" has been tweaked to be much closer to the actual target location where the text will appear. It is much easier to use as a result.

#### 11- Stack Requirements lessened:

Important!!!

Previous versions of the AVT software required 15000 bytes of stack. This was setup for you in the ICON, and was also recommended for the setting in the CLI (using the stack

command). The program should not now require more than 4000 bytes total stack. If the program does crash on your system, you may try increasing the stack, in increments of 1000 bytes. It is expected that 4000 bytes is now sufficient. This will have the greatest impact on users with small amounts of memory, but is welcome in all cases.

#### Mouse Pointer bug fixed:

In version previous to 3.06, after the AVT software had run and been terminated, the mouse pointer would still be able to get "below" the bottom of the Workbench screen when it was pulled down, and programs that started up after the AVT was run would tend to think that the screen they were running in was larger than it really was. The problem has been fixed by resetting all the appropriate values that are modified in the system when the AVT software runs.

Note: You may still experience problems with some software if you try and run it while the AVT\_Master program is executing... Programs that try to see what the system screen/window/view sizes are when they start will likely have problems.

#### File Requester changed:

In version 3.05 the old file requester (archive manager) was replaced with our new one, which is more compact, more efficient, and take less memory to run. In 3.06 it was enhanced by adding two new "toggle" switch buttons. The first one, which is labelled "2 Clk" will, when selected, allow you to select files by clicking twice on them. If it is not selected, then you will have to hit the button at the upper right of the requester. The second button is marked "Confirm". When this is selected and you attempt a "save" operation and the file already exists, a requester will appear asking you to confirm the overwrite operation. Other changes include placement of the list of files, dirs, and roots in a different order, and the moving of the other buttons around to new locations, removal of the 'delete' button, and removal of the status line.

#### Bug when looking at files in RAM: fixed:

When looking at files in RAM, the requester would usually only show them to you the first time you looked at them. This was due to a bug in the CBM-supplied ramdisk software. We have found a workaround for the bug however, and you should no longer have this problem. The new software will continue to work should CBM actually fix the bug - we no longer make use of the broken procedure call at all.

#### Possible fix for Narrow mode:

In v3.06, using AVT in the Narrow mode would GURU. We believe that this was due to exceeding the stack size; we have accordingly reduced the stack usage in those routines.

If you still experience a GURU when attempting to use narrow mode, increase the size of the STACK in the icon (assuming you start the program from the ICON) or use the stack command in CLI previous to running the AVT program. We suggest increasing the stack in 1k increments until the problem disappears. The default stack is 4k, so try 5k, 6k and so on.

#### Size reduction:

v3.07 contains more rewrites, and is slightly smaller than v3.06 and previous.

#### 3.08 Beta updates:

- Touch Tone Pad output level has been modified to conform exactly to FCC requirements.
- Added "Custom Colors" menu item. This selection offers you the added functionality of "fixed-up" HAM mode. Fixed up ham mode provides a much more error free display, but takes several seconds more to calculate. Select under the SSTV menu to activate. Fifteen color registers are used to provide sharp edge fixups after the entire image is analyzed for pixel errors.
- Sharpen Area: This is an advanced operation that works in a manner similar to the Custom Colors option. However, rather than analyze the entire image for errors, the SSTV screen is brought to the front, and you use the mouse to select a rectangular area. This area is analyzed for errors, and the 15 color registers are assigned accordingly, then the entire image is redrawn. You will often see a great deal of increases in detail for the area you select. The smaller the area, the better it works.
- You can press the "2" key (at the upper left of the keyboard) at any time when you are in SSTV mode to see how the color registers are currently assigned. This lets you know what color registers you'll get if you select save IFF.
- Fixed bug that caused displays after Rx of Scotty/Martin modes to show in the wrong mode.
- Complete color bar is now generated for avt 94 and 188.
- Added direct drive of the Black Belt Systems "SoftPanel", shows LED bar graph for tuning even while receiving. Contact Black Belt Systems for details on the SoftPanel.
- Special PAL version is now available - in testing at present.
- Added three new ARexx commands for use with Amiga frame buffer products. These provide external software with important information about the internal image the AVT is

currently loaded with. The commands are:

- 1 - XDIM; this returns the x width (128, 256, 320) of the current image.
- 2 - YDIM; this returns the y height (128,256,240,400) of the current image.
- 3 - BUFFAD R; this returns the hexadecimal address of the RED data buffer.
- 4 - BUFFAD G; this returns the hexadecimal address of the GREEN data buffer.
- 5 - BUFFAD B; this returns the hexadecimal address of the BLUE data buffer.

These commands are used to provide everything needed for an external display device driver to show the picture in the AVT's buffers. You get the addresses of the three buffers, the x and y dimensions, and from there the external buffer driver software can draw the image. Our new HAM-E product driver uses this strategy to display a 262,000 color image right on your Amiga's screen. Contact Black Belt Systems for details on pricing and availability. Currently, June is the release date and retail price is expected to be in the \$300.00 range.

- many other bugs have been fixed in this beta version.

#### Black Belt Systems file reader - Read

This program will read any text file either from the CLI (Command line Interface) or from the WorkBench environment.

To use from the CLI, type: read filename

To use from the workbench, the document file must have an icon, which must be a "project" type icon. Using the WorkBenches "info" menu selection, you may need to change the icon's "default tool" to point to the Read tool. Examine the icon for this document for an example on how to do this.

Using the reader, once it has opened a document file, is easy. Click on the gadgets provided at the right edge of the display to move thru the text, or use the menus, or use the "Amiga Keys" indicated in the menus. Point and click file handling!

You can also use the cursor keys to move thru a file line by line, and press the ESC key to close the reader window. Using ALT with a

cursor key moves by pages, and using SHIFT with a cursor key moves you to the top or bottom of the file.

For those of you who wish to use this reader to prepare/handle text files of your own, be aware that it supports the full console device - this means that colored text of various types is supported, as is underlining, italics, BOLDFACE and so on. Keep the text lines in your document to 70 characters/line or shorter;

This way, the lines will not "wrap", causing the person reading the document to become disoriented, page-wise. Refer to the Amiga documentation for specific control codes for the console device if you want to produce effects like colors and underlining and so on. These codes are generally of the forms:

ESC[xxxm

The ESC stands for the ASCII character "Escape", which is hex 1B, or decimal 27. The numbers depicted below are also ASCII characters; So to set underline mode, you send four full characters, of which only the escape character is a "non-printable" type:

ESC [ 4 m (sets underline)

For instance, to make orange text, you use: ESC{33m  
and...

white: ESC[31m

black: ESC[32m

blue: ESC[30m

Normal: ESC[0m

BoldFace: ESC[1m

Underline: ESC[4m

Italics: ESC[3m

Inverse: ESC[7m

Due to some subtle bugs in the Amiga Console.device, you should take care that when you turn on a feature you do it individually in each line. By this we mean that you should not turn on underline text in one line and turn it off in a later line. Turn it on at the beginning of each area within a line that needs it, then turn it off at the end of the line. Otherwise, when scrolling data backwards, the console device may apply that effect (underline, in this case) to every line on the screen. It's a pain in the ribs, but there's nothing to be done about it until Commodore releases a new version of the operating system - if they fix it then!

Enjoy! AA7AS, Ben