

HANDY RELEASE 0.7

1 February 1989

CONFIDENTIAL and PROPRIETARY

They're back ...

This document describes the V 0.7 release of Handy development environment. It includes these sections:

- Handycraft V1.46
- Handebug V1.46
- HANDY: Notes
- 6502: Notes
- What You Must Do To Start Using This Release

Handycraft V1.46

Lots of things have changed with Handycraft. This is a bug-fearful release. It all seems to work alright with my humble test programs, but I need you guys to try it out and give me feedback about anything that's wrong.

There's a small chance that Handycraft now copies all settings of the selected sprite when pasting a new sprite.

When you paste a sprite, the sprite name and data name are initialized to the name of the file you've pasted. I debated about the suffix to add to the data name, then decided to add none for now because this will be something that you can add to the Preferences when we get them implemented.

Handy sprite imagery color palette redirection is now supported by the emulator, maybe. It works with my simple test cases, but you must bang on this for me. The rule is a slightly complicated one, and can be found on the separate document **HANDYCRAFT EMULATION DATA DEFINITION**.

When Handycraft outputs SCB and DATA files, a .END is written at the end of each file.

It may be true that in emulation mode all of the SCB reload/reuse bits are handled correctly. This refers to both reloading the H/VSize, the Stretch and Tilt values, and reusing the previously-uploaded palette.

On the EDIT PARTICULARS display, the "Output Pixel Depth" gadget should now always reflect the minimum required number of pixels.

By popular demand, when a sprite is pasted its default position is in the middle of the display.

Handycraft recognizes the sprite types BACK_SPRITE and BACKNONCOLL_SPRITE. All other sprite types are currently treated as NORMAL_SPRITE. The next release will add the other sprite types except for XOR_SPRITE, which may not be supported.

If you ask for a huge sprite, larger than Handycraft can handle, in days of olde Handycraft would crash but now Handycraft laughs at you and doesn't render your sprite. Unless of course this isn't true.

There's a chance that you can paste 5-plane images into Handycraft now without causing it to explode. The extra plane is currently ignored (but not for long).

Other bug fixes:

- Gone, perhaps, is the mysterious scribbling into memory that many people saw as garbage in their imagery. Those who saw garbage in their imagery were the lucky ones, as Handycraft was always writing 16 bits of garbage somewhere, and if not in your imagery then guess where! This was probably the cause of mystery crashes that now might be exorcised.
- Used to be that if you asked to paste a non-existent file the system would go boom. Not no more, maybe.

Next release:

- Support for other sprite types
- Stretch and maybe Tilt
- New way to handle CHIP RAM to help avoid out-of-memory problems
- IFF output
- User-definable TRIM EDGES value

Handebug V1.46

You asked for it, you got it! Place the mouse inside the Display area, press and hold the left mouse button. Now move the mouse around *outside* the Display area. Wheee! OK, OK, that's enough fun and frivolity. How does it work? I'm glad you asked.

Placing the pointer inside the Display area and holding the left mouse button down activates scrolling. Scrolling will then occur whenever the pointer is outside the Display area. When the pointer is above or below the Display area the Display will scroll in that direction. Moving the pointer to the left or the right of the Display area increments or decrements the display address by one, causing the display to roll in the appropriate direction. The rate of scrolling is relative to the pointer's distance from the Display area. Releasing the left mouse button deactivates scrolling.

Occasionally scrolling may pause for a fraction of a second when a page is being uploaded from the 6502 machine.

Symbol loading is so fast it's gonna blind you. Loading a 14.5K symbol file with 900 symbols takes about 5 seconds. Whoosh!

In advance of booting off of real (sort of) Handy ROM, if either field of the Bootstrap requester is cleared then the bootstrap downloader will skip that portion of the procedure. This will be used to support getting the bootload.bin code from a pseudo cartridge rather than from Handebug.

Now you can select (double-click) the operand field of a disassembled instruction and the select value will be the result of evaluating the operand expression. If the operand is a simple memory reference, no problem. If the operand is something complex (e.g. (table+2),Y) the expression is evaluated before the value is stored. Note that if a register is part of the expression the contents of the register at the time of evaluation will be used in the calculation.

Bug fixes:

- MemWatch fields now don't get scrambled
- Upload Binary now works with lengths greater than 255 bytes
- Structure editor display is now updated after Single Stepping
- Structure editor handles single-byte HEX fields correctly, especially when using +/- keys

- Caught a GURU! Thanks to Steve L. another insipid GURU has been stamped out. Previously, if you BootStrapped Handy and then immediately reloaded symbols ... poof!

HANDY: Notes

You can now debug using normal interrupts. The CONTINUE, STEP and STEP FLAT functions don't reset the interrupt table anymore.

When you enter the debugger, the audio is now turned off. However, this may need further refinement, as the sound and music drivers have minds of their own.

6502: Notes

There's a new SETDISP macro in display.mac, which macro doesn't start EOL/EOF interrupts unless you have specified EOL_USER or EOF_USER. Formerly, the SETDISP macro always turned on interrupts for EOL/EOF.

The system variable Interruptus is now no longer defined.

The WAITSUZY macro no longer waits for the Interruptus field to go negative, and instead watches the SPR_SYS bit SPRITWORKING, which you weren't supposed to do before but now Needle-San sez it's OK.

By popular demand, the INITSUZY macro initializes SPRSYS and SPRSYS_RAM to zero.

Miscellaneous

Do you know that the hardware now provides a bit named NO_COLLIDE in SPR_SYS which you can set to turn off *all* collision handling systemwide? Well, it's true. When you set this bit, you can ignore hardware collision completely. If you don't set this bit, then if any one of your sprites isn't explicitly non-colliding you must provide a collision buffer using the SETCOLL macro.

There's a new S:Handebug.defs file, which has the new and improved SCB structure definition. Your old one is saved to S:Handebug.defs.old.

What You Must Do To Start Using This Release

If you use any of this release, you must use all of this release. The new monitor.bin and Handebug need one another to work correctly, and Handycraft needs the new monitor.bin and sprite.src.

Everything is saved to .old. You will find the old stuff at these locations:

- HANDY: can be found in SYS:Handy.old
- 6502: can be found in SYS:6502.old
- S:Handebug.defs can be found at S:Handebug.defs.old