Beginners notes for NitrOS-9 "EASE OF USE" image
VERSION 1.0.0

THIS IS DOCUMENTATION FOR NITROS9 EASE OF USE (EOU) VERSION 1.0.0 FOR PEOPLE NEW TO THE NITROS9 OPERATING SYSTEM. PLEASE SEND FEEDBACK TO L. CURTIS BOYLE (curtisboyle@sasktel.net), BILL NOBEL (b_nobel@hotmail.com) AND DAVID LADD (davidwladd@gmail.com). We want to try and make this package as easy to use/understand as possible.
The NitrOS-9 "EASE OF USE" project's goal is to make available a "pre set-up" version of NitrOS-9, so that the beginner NitrOS-9 user does not have to worry about setting up a boot disk/file, setting up Multi-Vue (the GUI program), set up programs to have their needed file(s) set up in specific directories, etc. It is meant to be "point and click", and have (almost) all programs set up to run from a command prompt without having to change between various CMDS directories, etc. It is also meant to have more robust HELP than the stock OS-9 help has; by including various 3rd party apps/utilities, etc. This by no means indicates that a power user can't use this package - they can do everything from the command line like they are used to, if they wish. With a lot of really useful extra commands compared to stock OS-9 Level II.

It should be mentioned that, in order to make the experience as painless as possible, that their are system requirements for this package that are higher than the stock OS-9 Level II. We are also officially limiting the hardware supported to 4 basic configurations to make it easier for us with testing (and that we have the hardware needed to test), and are likely the most commonly used ones by most people.

NOTE: Other distributions are available that will be handled by other people with our blessing; but any issues/questions that you have with those configurations should be directed to the people making those builds. A list at the time of this writing are:
- Matchbox Coco version - Bill Nobel is in charge of this distribution, which includes some enhancements. Contact him at **b_nobel@hotmail.com.** This version we will put on our main download site, though it will be delayed from the normal releases.
- Coco3FPGA version - Michael Furman is in charge of this distribution. Contact him at **mikeyn6il@gmail.com.**
- XRoar version - Michael Furman is in charge of this distribution (which uses a virtual IDE Drive interface instead of EMUDSK). Contact him at **mikeyn6il@gmail.com.**

For the "official" versions:
System requirements for this version are:
- A Coco 3 with at least 512k, either 6809 or 6309, with a CocoSDC, **OR**
- A Coco 3 emulator (MAME or VCC or OVCC) set up as a Coco 3 with the specs above, but replacing the CocoSDC with a hard drive emulation (built into both MAME and (O)VCC).
Because the 6309 version of NitrOS-9 has extra features and is faster, if you have the hardware to support it, we would highly recommend installing that version. We should also mention that we have defaulted the initial boot screen to 32 columns, and the main window (called "/term") is defaulted to 40 columns. This is help users running on RF or composite screens, so they can read the text easily. The GUI (called "Multi-Vue", and the main program to start it called "GShell" (short for "Graphical Shell")), also defaults to 320x200 resolution for the same reason. Two other windows are currently fired up by default that are 80x25, so people with RGB monitors can choose those to start on if they wish. **We have added an easy way (requiring no changes to the OS9Boot file) to change the /term default screen type.** You can change the **GShell** default screen resolution with a text editor and editing the /dd/sys/env.file.

The hard drive images (6809 and 6309) are common between real hardware and the two emulators; the emulators also include a floppy image (containing the main boot, with the emulated hard drive drivers), that will then kick into the hard drive image. So, on an emulator, you must mount the file "63EMU.DSK" into floppy drive 0, and mount the file "63SDC.VHD" into hard drive 0. (or "68EMU.DSK" and "68SDC.VHD", respectively, for the 6809 version). Then, type DOS to boot.

For the CocoSDC, you need only copy the 63SDC.VHD (or 68SDC.VHD) file onto your SD card, and after you have fired up the Coco with that card in the CocoSDC, change to that drive image, and type DOS. If you have Coco Explorer, you can use it's menu system to select the VHD file as well. If you plan to be using NitrOS-9 a lot, you can also set up the CocoSDC to automatically mount that image when you turn the Coco on. (Another option is use a separate SD card for NitrOS-9, and have that auto-boot). For those of you lucky enough to have a **GIME-X**, there is also special versions (68GIMEX.VHD and 63GIMEX.VHD). These are pretty well identical to the regular versions, except they turn on the 2.86 MHz turbo mode as much as possible, and run much faster.

The VHD is a "skitzo" image that contains a very small partition that contains a BASIC program. If you rename it to AUTOEXEC.BAS, and you have your CocoSDC config file set to do so, you can have NitrOS-9 automatically boot as soon as you turn the machine on. Thanks to David Ladd for setting up this feature, for those that need/want it. You can also directly launch the .VHD file from SDC Explorer, as normal.
If you need more room then is free on the main VHD, you can also download a second blank hard drive image (with almost 128MB free):

http://www.lcurtisboyle.com/nitros9/EOU_USER_empty_harddrive.zip

(This should be mounted as the second hard drive in MAME, VCC/OVCC, or the CocoSDC).

This is mainly for the programmers and tinkerers; you can put all of your own code, copies of files on the main drive that you have modified, etc. This way, when we issue a new release of EOU, it won't overwrite any changes that you have backed up on this 2nd drive. We are not including this image in the distribution ZIP files, so that you don't accidentally extract and overwrite that hard drive image with a completely empty one. If you do not have it yet, please use the direct download link above.

Upon booting, you will first encounter the "NitrOS-9 BOOT" screen. All of stuff that flies by on the screen is a listing of all of the system modules, as they are loaded, and is mainly meant for debugging, and you shouldn't need to pay much attention to them (and it does give you something to look at while the system is booting). After the operating system is loaded, the screen will clear to black on white, 40 column text (unless you change this; see the **env_file_documentation.rtf** file for details on how), and it will load in large library of graphical fonts (49 as of this writing), paint patterns, mouse cursors, etc. that are used by the GUI and other programs. It will then prompt you for the date and time (unless you are using a **DriveWire** boot), and fire up two more (80 column) windows in the background, and then give you instructions on how to start up Multi-Vue, for the point and click mouse interface.

NOTE: If you have a Smartwatch installed (a real time clock chip), you can edit the /dd/startup file (using EDIT, MINTED, VED, etc. - whichever text editor you prefer) and replace the line that has SETIME (etc.) on it with 'SWREAD'. This will set the software clock time to your Smartwatch time, and you won't have to type it in any more.

SPECIAL NOTES FOR EOU:

1) If you are running on a real Coco 3 with a **MiniMPI** (from zippsterzone.com), there is an oddity (with workaround). If you have just the CocoSDC by itself, or with a 2nd cartridge that does not have an auto-boot ROM (so like the Sound/Speech pack, Glenside IDE controller, etc.) it works fine. If, however, you have either a game cartridge that normally auto-starts, or a hardware card that also has an auto-start ROM (like the Orch-90, for example), then you must put the CocoSDC in the slot closest to the Coco in order for the CocoSDC detect to not hang. This is at the hardware level, so there is no software workaround. 4 slot MPI's & the MegaMini MPI don't seem to have this issue, from out testing.

2) If you are using the **DriveWire** boots, it should be noted that NitrOS-9 tells the DriveWire 4 server to shut HDBDOS translation off (so that it can access the hard drive images properly). If you depend on this being on when you are running Disk BASIC (HDBDOS or YA-DOS, for example) then you will have to turn it back on on the **DriveWire 4** server after exiting NitrOS-9 (it's a simple checkbox). Just something to be aware of.

3) For the emulators, you have the option of using an emulated real time clocks. However, we have noticed some... quirks with various versions of the emulators, where the clock sometimes works, and sometimes doesn't. If the following doesn't get the emulated hardware clock to work, you can always switch back to the software clock, which works irregardless:

A) For **VCC** or **OVCC**, Set slot 3 in the multipak to "Hard Disk + Cloud9 RTC"
B) For **MAME**, select the Disto clock.

For the best reliability, **VCC** should be version 2.1.0b or higher (not to be confused with 2.0.1b, which has problems on some systems). For **MAME**, make sure you set the clock from the built in MAME "GUI"; frontends like MESSGUI don't always pass that setting on correctly, and even lie about it. :-)

To switch between the various boot options, use the new **SwapBoot** utility. Depending on what system you are running (emulator, real hardware), your options (and number of options to choose from) will change.

4) For the **GIME-X** or the emulators, the system boots we have set up default to 80 columns and RGB. For regular hardware it is set up for composite and a 40 column boot and 40 column GShell. You can change these by editing settings in the environment file. There a lot of new settings in there that can be changed with a text editor, or the latest **CONTROL** program (from the Tandy menu in GShell). There is a separate documentation file, called **ENV_FILE_DOCUMENTATION.RTF**, which explains both the older settings and the new ones. If you wish to use a text editor, your current settings are in the /dd/sys/env.file. If you use **SwapBoot** fairly often, you will want to edit the env.file* files that you commonly use in /dd/ boots, so that they will carry over after rebooting.

Quick notes on special keys in NitrOS-9

I will group these into 3 sections: **Universal** (keys that do special things no matter what program you are in), **SHELL** (the main SHELL command prompt), and **MULTI-VUE** (the GUI). This is ***not*** a complete list (see the manuals, available online at the Coco Archive at **http://www.colorcomputerarchive.com**), but the most common ones that you will find yourself using a lot.
NOTE: I will be listing the keys from an actual Coco 3; consult your emulator documentation to see what equivalent key(s) will do the same thing (whether it be Windows, Mac or Linux). Also, emulators like MAME allow you to change these, anyways.

UNIVERSAL

Key repeat is on by default, turned on (hold down a key, and after a short delay, it will start repeating). You can change this in the Control Panel in GSHELL, both to change the delay length and the repeat speed, and even shut if completely off. Programs can also do this themselves, if they wish.
<CLEAR> = Go forward to the next window. These can be completely separate full screen windows, or multiple windows on the same screen.
<SHIFT-CLEAR> = Go backwards to the previous window.
<CTRL-CLEAR> - Toggles the "keyboard mouse" on or off (the system boots with it off). This will change the normal function of the 4 arrow keys, and the F1/F2 keys, to simulate a mouse. The four arrow keys move the mouse in the four directions (you can hold 2 down at once to move diagonally), and F1 and F2 are the left (red) mouse/joystick button and right (black) joystick/button respectively. Most programs only need F1, as a lot of people back in the 1980's when OS-9 Level II was released, still had the older 1 button joysticks or mice.
While the keyboard mouse is on, the arrows jump 8 pixels each time you hit one. If you hold <CTRL> and an arrow at the same time, the mouse cursor will jump to the far edge of the screen corresponding to which arrow you hit. If you hold <SHIFT> and an arrow at the same time, you will move 1 pixel at a time. Please note - if you are running on a graphic window, but just printing text, the keyboard mouse overrides the normal functions of the arrow keys (like backspacing, etc.). If you find that you can't backspace, and you see the mouse cursor on the screen, hitting <CTRL>-<CLEAR> will switch back to normal keyboard mode.
<CTRL-0> (that is a zero, not the letter 'O'). This toggles the CAPSLOCK on and off (the system boots with it off). This forces all text you type to uppercase.
<CTRL-W> - when listing text files to the screen, this will Pause the display (like <SHIFT>-<@> in Disk Basic). Any other key un-pauses.

There are some other **<CTRL>** keys used to generate special symbols that the Coco keyboard doesn't directly support (an example is **<CTRL-minus>** to do an underscore); you can find those in the OS-9 Commands section of the OS-9 Level II manual, in Appendix D "OS-9 Keyboard Control Functions", so I won't repeat all of them here.

SHELL

Shellplus 2.2A is the default shell we have installed. It has some special features, like command history, that are incredibly useful as a time saver. It also allows users to change the default prompts, and we have picked (as a default) one that give you a lot of info about the window you are looking at, without taking up too much room. Each window with a shell on it will show you 1) the window name (like "/term"), the process # (like "02"), and the data directory you are currently in (like "/dd").

By default, we have command line history enabled in the shell with full editing keys. This means that you can bring up previous commands you have used on this particular window/shell, and re-use them (or edit them), without having to retype the whole thing. You can see these by tapping the **UP ARROW** (to go backwards through your command history), or **DOWN ARROW** (to go forwards through your command history). (this is similar to as you see on Linux, Windows and OSX command lines). You can use the line verbatim again by hitting **<SHIFT-RIGHT ARROW>** to go the end of the line, and then hit **<ENTER>.** You can also edit the line, using the following keys:
<LEFT ARROW> - move left one character
<RIGHT ARROW> - move right one character
<SHIFT-LEFT ARROW> - move to beginning of line
<SHIFT-RIGHT ARROW> - move to end of line
<CTRL-LEFT ARROW> - delete character at current character position; shift rest of line left.
<CTRL-RIGHT ARROW> - insert character at current character position; shift rest of line right.
<ENTER> - accept current line up to current position as is, and execute it.
<UP ARROW> - abort current line edit, go back up one line in command history
<DOWN ARROW> - abort current line edit, go forward one line in command history

Note: One difference compared with Windows command prompts - hitting **<ENTER>** in the middle of line will cut the command off at that position and execute it; Windows will take the whole line. If have made edits but want to use the remaining part of the line verbatim, simply hit **<SHIFT-RIGHT ARROW>** first and then **<ENTER>.**

A further note: The editing keys listed above (but not the history keys) are actually universal to all programs that do the equivalent of an INPUT or LINE INPUT (ie asking for a line of input at a time, not just a single character). So other programs, whether they are written in assembly, C, Pascal, BASIC09, etc. will all allow those editing keys - but they will be based on the last previous input ONLY. Some programs can also "pre-load" the text buffer with default text, and display it, with you able to edit that using the keys above, as well.

SHELL also has a special "enable wildcards" character (:) that, if you put it as the beginning of the line, will enable SHELL wildcarding. The types of wildcarding supported are:
'**?**' for single character wildcard match.
'*****' for multi-character wildcard match.
'**[a-b]**' for a single character wildcard match within a certain ascii range (**[a-b]** would mean that the character has to be an 'a' or 'b'; **[0-9]** means it would have to be a numeric digit, etc.).
There are some limitations to SHELL wildcards that you should be aware of, though:
1) They only work for commands that accept multiple filenames as parameters on the command line.
2) The internal buffer that SHELLPLUS uses to generate a wildcard file list is limited in size (2K in expanded form).
3) If you send them to a program/utility that does it's own internal wildcarding, you may confuse that program.
Some commands that do use it well are **DEL**, **CP**, and **TOUCH**.

MULTI-VUE (GShell)

Most of the time, you will simply be using your mouse/joystick (or even keyboard mouse) to select menus, programs, folders, drives, etc. But, there are some special keys that you can use as well in GShell:
<Q> - Quit GShell. This will ask you if you are sure; you can use the mouse to answer, or hit **<Y>** or **<N>.** This is the same as File->Quit.
<S> - makes an overlay window with a shell in it, so that you can run a command line. Type EX to quit the shell once you are done.
<$> - makes a resizable window shell on a different screen (same as File->Shell menu)
<?> - Calls HELP command for the currently selected file and displays it in an overlay window (same as selecting a file with a single click, and then clicking the question mark in the upper right part of GShell). If there is no help for the file, it will say "No help available"
<=> - Refreshes currently selected directory. Under NitrOS-9, vs. the original OS-9 Level II, it should do this automatically, if a file has been added/deleted/changed. Sometimes it will miss such a change if you have an overlay SHELL running on top of GShell at the exact moment a file has changed, so it can still be useful.
<SHIFT-UP ARROW> - will go up a page of icons (if there is a page up to go to). Same as clicking the up arrow on the right.
<SHIFT-DOWN ARROW> - will go down a page of icons (if there is a page down to go to). Same as clicking the down arrow on the right.

General notes on MultiVue (Gshell)

When you first fire GShell up, it reads it's environment file (which is a list of default settings, etc.), and then displays the floppy and/or hard drives it knows about on the left side. (There is also a printer and a trash can). You select a drive (make sure that you have a disk in it, if it is a floppy!) by pointing the mouse cursor to it, and left clicking once. You will then see a display of the files/folders/applications on it, which you can scroll through using the up and down arrows on the far right side. To select a file/folder/application, left click it once (it will highlight it), and then you can click on the Files menu and do various things to it (depending on the type, some selections will be non-bold and not accessible to you), such as list the file, copy it, get statistics on it, rename it, delete it, etc. You can also click the Question mark in the upper right corner to get help for that application, if any is available (any application with it's own icon should have help available). There is also a **Sort** option, to permanently sort it alphabetically (it actually remakes the directory in alphabetical order). If you follow the OS-9 "Standard" (not enforced), then directories will be all uppercase and will show up first, as well as AIF (Application Interface Files - these are used for apps, like the old PIF files from back in Windows 3.xx days) files, and then all other files. There will be folders (also known as sub-directories) that contain more files, data files (which look like little pieces of paper with their corner folded), regular program files (which look like mini-GShell windows), and AIF program files (which will actually have unique icons).

If you see a program icon (generic or AIF), you can double-Left click it to launch the program. Some programs will then immediately run (like most of the game icons in the Games folder), others will prompt you for parameters to pass to the program first, in an overlay window.

A lot of programs will create their own full sized screen all to themselves. A few, and most of the 'Tandy' Menu mini-applications (the funny looking 'X' type thingie left of the 'Files' menu - it is meant to look like a miniature version of the TC (Tandy Corporation) logo) will allow you to resize and/or put more than one program on the same screen. These will show a box that you move around with the mouse (to position where you want the upper left corner to be), and at the same time shows you the **minimum** size the window has to be for that program. Once you have the upper left corner where you want it, left click once to lock it in place, and then drag the mouse to resize it to the size you want. Once you have the size you want, left click again to run the program in your new window.

Note: A few games support running in resizable windows; two I can mention are Rogue and Zone Runner. If you get a circle with a slash through it as your mouse cursor, that means that you can't place the window at it's current location (NitrOS-9 does not currently allow overlapping windows), so you will have to pick a different spot/size.

It should be noted that, even with a 512K Coco 3, some games take a large amount of RAM, and you may not be able to run too many of those at the same time. It should also be noted that some programs/ games are smart enough to know that they are not the currently active window, and will either pause or sleep until you make them the active window again... but others will run full speed in the background while you are doing something else (for some games that may take awhile before requiring user input again - Flight Simulator II & Zone Runner come to mind... this actually works out in your favour).

We are thinking of having this package automatically start up GShell on boot - if we do do that, would it be best to fire it up on the starting window, or one of the secondary windows that get fired up as it boots?

Tips and tricks

- When printing text on a graphics window - 8x8 fonts display a **LOT** faster than 6x8 fonts do - although they fit less text on each line. A full screen window using an 8x8 font will show 40 columns for a 320 pixel wide screen, and 80 columns for a 640 pixel wide screen; you will get 53 and 106 columns respectively with 6x8 fonts. Please be aware that not all programs check this properly, and those that don't will assume 8x8 fonts at all times.

- Hardware text screens are the fastest of all; especially on a 6309 based system. Utilities like **LIST** (which lists text files) are **much** faster in these modes under NitrOS-9 than under Disk BASIC.

- For **GSHELL** - the default icons are all 4 color, so the two 4 color view modes (320x200 and 640x200) display and update much faster than the 16 color mode (which dynamically converts 4 color icons to 16 color each time they are drawn on the screen).

- <**CTRL-E**> is usually used to abort a program (unless it has over-rode that - kind of like ON BRK in Coco 3 BASIC), but if the program has done funky things like turn the cursor off, or shut off screen echo, or turned page pause on, etc., those weird settings will likely stay that way on the window you are on. Try to quit a program through it's own quitting mechanism (Quit menu, CTRL-Q. etc.) to avoid this. If a program gets "stuck" and you can't get it stop no matter what, you can <CLEAR> to another shell window, then type PROC to get a list of processes. Find the program name on the window that is running amok, and then type KILL <process #> to forcefully abort it. The same problems with hitting <CTRL-E> listed above will apply, though.

- On a real Coco, hitting the RESET button once will immediately reload the NitrOS9 boot file and relaunch NitrOS9. Hitting it twice will boot back to BASIC. You can also type REBOOT at any Shell prompt to reboot to BASIC.

- If you want to change the defaults for Multi-Vue, you can use one of the text editors included (not all documentation is on for all of these yet), or the latest **CONTROL** program. Probably the easiest it use is called ED, and it uses a mouse. To edit the currently active Multi-Vue/GShell configuration file, you will need to go to a SHELL window, and type the following:

CD /dd/sys

ed env.file (or whichever editor you prefer; my personal favorite is VEd).

Most of the settings are documented in the Multi-Vue manual on the Coco archive, and you should set them to how your system is set up (like whether your mouse/joystick is in the left or right port, whether you have a high-res mouse interface, your keyboard repeat delay/speed, and monitor type), but there are also some new ones that you may find useful:

GSHPAL0=r,g,b

GSHPAL1=r,g,b

GSHPAL2=r,g,b

GSHPAL3=r,g,b

The above 4 settings set the first four palettes that GShell itself uses, without changing the system wide default palettes. The RGB settings are exactly the same as used by PALET0= from the manual, and elsewhere in the env.file. NOTE: These should be set with color 0 as the darkest (usually black), and brightening as you go until color 3 is the brightest (usually white). This is preserve the 3D look in the menu bars, and also for some graphics viewing programs coming soon that you can incorporate into your own programs. As long as you go from dark to light for GSHPAL colors 0 to 3, you should be fine.

DEFTYPE=<window type>

Window type can be: 6 (320x200, 4 color), 7 (640x200, 4 color) or 8 (320x200, 16 color). This tells GShell what mode to start in when you first fire it up. Set this to your liking; we currently have it as type 6.

Many more new settings are now in the env.file to let you have more control on how the system boots. Please see the included **env_file_documentation.rtf** file for details on the new settings. The env.file now also includes comments, so that shouldn't have to look up the documentation to make further changes in the future. Also please look at the new **SWAPBOOT** command, to allow switching between settings (such as SDC only and SDC with Drivewire support). This command is explained both in the Beta 6 updates documentation, and the Driverwire documentation.

- While a lot of people just hit <ENTER> for the date and time, it is sometimes useful to enter them in, so any files you edit (even if you are unaware of it - like high score save files, etc.) are more easily findable by recent written to date/times. Some software development environments (like the DCC C compiler, and RMA assembler) run much more efficiently with real date/times.

- For viewing manuals in the /dd/docs directory, we highly recommend using the **VU** utility.

NOTE: Two copies of this documentation are in /dd/docs as well, although they are older versions.
NitrOS9-Beginner_docs.txt is the plain text version that you can load into a text editor, LIST, run MORE or VU on, etc.
NitrOS9-Beginner_docs.os9 is an experiment - native OS9 formatted documentation. Use LIST on this one; it should automatically switch to an 80x25 graphics window, switch to a slightly shorter font (so the letters don't run together vertically) and then have bolding, underlining, and even an italic or two. You can pause the listing either with **<CTRL-W>**, or by typing TMODE PAUSE before you LIST it. Do NOT list it from GShell, though - it redefines the window and kills the menu bar, which *really* confuses GShell).

Well, that's it for quick beginners notes for now. Please let me know if you have any questions, need any clarification, any suggestions for me to make this easier to understand for beginners, etc.

L. Curtis Boyle
curtisboyle@sasktel.net
NitrOS9 EOU Version 1.0.0 Beginners Documentation
12/02/2022