64drive

DEVELOPMENT TOOL

USER'S MANUAL

RETRO
active

NOTICES

Information contained herein reflects the
current state of the product at the time
of writing.

For the most current document please
see the Support page on the product
website at:

http://64drive.retroactive.be

See the About tab in the product Menu
to see more specific attributions and
software revision information.

CONTENTS

64drive DEVELOPMENT TOOL

INTRODUCTION

The 64drive is a tool designed to be used with the Nintendo 64 video game console for the ends of game development and/or modification.

Conceived out of frustration at the lack of options available on the market, it incorporates a capable feature set, robust hardware design, and incredible attention to detail.

I hope that you as the user will appreciate the polish and enjoy the product.

Physical Considerations

Please observe these guidelines when using the product:

• The cart is designed for indoor use only. Keep it dry, avoid dropping it, and don't let foreign objects get into it.

• Special attention should be paid to the edge connector on the bottom. If it gets dirty there will not be a good connection and in rare cases you might lose some data.

• This product supports upgrades in the field. It is possible to render the cartridge otherwise unusable if care is not taken when applying upgrades.

• All common sense rules for the proper treatment of expensive electronic equipment should be observed.

In Case of a Problem

Refer to the Troubleshooting section. In the event of a problem not detailed in this document, see the Contact section to send an email to Support.

GETTING STARTED

Check Formatting

The 64drive only works with memory cards that are formatted with the *FAT32* filesystem.
Older cards may need to be re-formatted to work. Under Windows you can check by viewing the card's Properties.
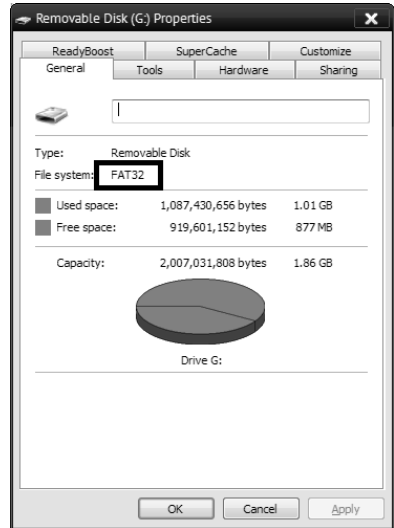
64GB and larger cards will default to exFAT, which is not supported. In that case, seek out a 3rd party program to manually make sure it is FAT32.

Install the Menu

Download the latest menu from the Support page on the website.
This is distributed as a compressed ZIP archive, containing the MENU.BIN (actual menu program) and also a text file changelog describing the new features or bugfixes relevant to that version.

Copy MENU.BIN to the root directory of your SD or CompactFlash memory card. This means that the file should not be within any other sub-folders on the card.

Copy Image Files

Next, copy over any images you want to run on the memory card.

Organization doesn't matter. However, there is a limit of about 1000 entries per folder. You may choose to categorize your images in folders, or you can dump them all in the root directory.

Both .Z64 (big-endian) and .V64/.N64 (byte-swapped) formats are supported. The latter is byteswapped by the hardware while loading so there is no speed penalty.

Turn It On

Put the memory card in your 64drive. MicroSD cards go in the right side with the gold contacts facing towards you.

Now power it on. The first thing you should see is the bootloader icon - a small

rotating symbol.
It should pause for just a second while it finds the menu program and loads it. Within just a moment you should see the menu for the first time (a skinned background is shown here -)

Getting Around the Menu

The buttons control the menu much the same way as they do in most games.

- 'A' button selects item or tab

- 'B' button cancels or returns to the previous screen
- 'C'-Up/Down buttons can be used to scroll quickly through long listings
- 'C'-Left/Right jump to the start and end of the current folder
- Both the analog stick and the d-pad may be used to move the cursor. The current highlighted item will have a pulsing border around it.

Loading an Image

Once a file has been picked, the loading options screen will be shown.
In most cases, the 64drive can pick the best settings and you can just press A to load the image straight away.

If the image to be loaded uses an internal cartridge save method, the 64drive will create a file to hold the save data on the memory card.

In some cases the software to be loaded may not be designed to run on your console's region. You can try forcing the region, but it is best to leave it on "Auto".

After You're Done

Press the RESET button on the N64 console to return back to the menu, or if the option was enabled, reset back to the same image.

(!) Save data is flushed automatically during play.

MENU - OPTIONS

In the "Options" tab you can tweak a few things. The changes are committed once you load an image, and are stored in the "*settings.cfg*" file on the memory card.

Use 'Saves' Folder

By default, when an image is loaded that supports internal savegames, the menu will create a new folder called "Saves", and also create a new save file within that folder.
Turn off this option if you want to keep the save files in the same place as each image file.

Reset Button Action

When the RESET button on the console is pressed, this option determines whether the menu is loaded or it instantly reboots to the same image.
If you want, hold the 'R' trigger while holding RESET and 64drive will invert this setting temporarily. Useful if you don't normally go back to the menu.

Font Size

Changes the width of the menu listing font. The smaller font looks better when stretched on widescreen displays.

Controller Pak Saving

Opens a submenu for management of virtual pak files. More details follow in the next section.

MENU - ABOUT

This tab displays the list of tools and people that helped and shaped the development of the 64drive.

It also displays the current versions of the software on your cartridge. These are detailed below:

Menu

This is the version of the MENU.BIN file stored on your memory card, and is upgraded by copying a new file from the Support page onto the memory card.

Firmware

Inside the 64drive itself is some memory that controls the details of how the cartridge works, at a very low level.
It is upgraded with a special program over USB. Periodically there may be a new firmware available for download from the product website that may fix problems.

Bootloader

Also contained within the cartridge is a small program that loads the menu itself at each power-on. Generally this won't need to be updated, but it's possible.

OTHER FEATURES

Background Skinning

The menu supports background images. However, the console must have an Expansion Pak installed.

- Create a JPEG image that is *exactly* 640x480 pixels, and less than about 100KB in size.
- Rename the file to "background.jpg".
- Copy the picture to the root directory of the memory card, just like you did

for the MENU.BIN file.
- Turn on the console.

Every time the 64drive is booted, it will check for the presence of this file. If found along with an Expansion Pak, it will decode the JPEG, convert it to an intermediate format and render a transparent overlay using Floyd-Steinberg dithering.

The menu will create a system file called "background.bin" after the conversion is complete.

The menu will detect a change in the JPEG's filesize and apply the changes if it is found. To remove the background, delete both "background.jpg" and "background.bin".

Quick-Start

While in the menu, you can instantly load the previously started image by pressing the START button.

Additionally, you can hold down the START button when turning on the console, and bypass the menu completely.

Controller Pak Saving

While the 64drive can easily handle and isolate savegames that are supposed to be stored internally, it cannot access the Controller Pak while it is running user software.

A lot of software will use up the entire Controller Pak and because the 64drive can't easily get around that, this option exists.

With this option you can have 1 physical Pak plugged into your controller, and copy the entire contents into a virtual "container" file stored on the memory card.

The menu supports 6 of these virtual "containers", so that if you run out of

space on your real Controller Pak, it may be set aside. You may move around the pak contents as whole, but at this time removing individual entries is unsupported.

All software designed for the N64 console utilizing a Controller Pak should have a menu for editing Game Notes - this is accessed by holding START on the controller just as the software starts (just after the menu loading screen goes away).

The virtual containers are stored on the memory card as "*virtpak1.bin*", "*virtpak2.bin*", and so on. The format is a raw 32KB dump.
Astute readers may wonder if this is the same as a DexDrive save file - it is, but not quite. DexDrive saves have an additional 0x1040 byte-long header at the start of the file.

Force NTSC Menu Mode

For PAL users only, the menu will be slightly squashed vertically due to the higher line count of the PAL video standard.

If you have a TV that supports 60hz, you can press Z+DpadLeft, and then Z+DpadRight to switch the display mode to NTSC.

This option is persistent. If your TV does not support this mode, you may see "SYNC LOST" or something similar. Just enter the button combo again to toggle it back off.

Alternatively, you can delete the "*settings. cfg*" file from the memory card to revert to the defaults.

Using Saves From an Emulator

Internal game saves can be used in an emulator, and vice versa. However, file naming is very important.

PC emulators such as Project64, Mupen, 1964, etc., generally create their own save files when they run games that need them. EEPROM saves can be simply copied back and forth, if the file names are what the emulator or 64drive expects.

SRAM and FlashRAM saves tend to be stored by emulators as little-endian, due to the heritage of the Intel x86 platform the emulators were designed for. 64drive saves in the game's native big-endian format. So to exchange these you will have to perform a 32-bit endian swap on the data.

The UltraSave tool on the support website page has the option to do this conversion for you (The Ultrasave hardware is not necessary).

NES Emulation

While the menu does not directly emulate anything, it does support automatic loading via an external emulator.

Files with the .NES extension are loaded via the *Neon64* program,written by "hcs". Download his emulator and extract the "neon64bu.rom" file to the memory card's root directory.

The INES image is automatically placed at the 2 megabyte offset in SDRAM, and the emulator loaded at 0x0 before it.

MP3 Playback

While not a supported feature, the menu has very basic music playback support. MPEG Audio layer I, II, and III can be played.
When an image is loaded, playback is stopped.

USB MODE

The 64drive has a dedicated USB 2.0 High Speed FIFO that can be used to:

- Load images into SDRAM
- Select save emulation type
- Upgrade firmware
- Set emulated CIC
- Do custom communications between code running on the console and the PC

How it Works

When the 64drive detects that a USB cable has been plugged in, it immediately switches modes.

In this mode,
- Save writebacks are disabled
- Bootloader is disabled
- 64drive appears as a plain cartridge
- USB registers are enabled

When the USB cable is NOT connected,
- Save writebacks are re-enabled
- Bootloader is enabled

This functionality is ideal for development. Since the bootloader is disabled, you can reset the console immediately without corrupting the RDRAM state.

Considerations

64drive has a brand-new switchable lockout chip (CIC) that can be programmed on the fly to match any previous lockout chip existing during the N64's lifespan. However, in USB mode, you must explicitly tell it which lockout chip to emulate.

Because the bootloader, menu, and by extension, bootemu are bypassed in this mode, the header type of the image you upload *must* match the CIC parameter you pass to the USB upload tool.

For example, when instructed to emulate a 6102, you could boot *Super Mario 64*, and when emulating a 6103, *1080 Snowboarding*.

As for what type of lockout chip each image expects -- that depends on on the bootcode in the first 4 kilobytes of the image - (0x40 - 0xFFF).

Homebrew images can easily be patched with the proper header type using a hex editor or the "RN64CRC" tool.

The power-on default is 6105/7105.

Tools

The "64drive_usb" tool is available from the support website. With it you can transfer your homebrew images to the cartridge for testing and debugging.

You can also read and write directly to the USB FIFO using a set of memory mapped registers.

The specifics are detailed in the separate Hardware Specification document available from the support website.

Sample code and examples are also on the support page.

TECH DETAILS

This section pertains to information that some may find useful as background knowledge, or to further their understanding with troubleshooting an odd issue.

Region Information

- NTSC - Used in the United States, Japan
- PAL - Used in most other countries such as the UK, most of the EU, Australia
- MPAL - Brazil

For more detailed country listings consult your favorite search engine.

To see what region console you have, flip over your N64 and look at the printed label on the bottom. The region is located after the model number (NUS-001).

Cartridges original intended for sale in the USA have notches on the rear lower corners of the cart case. Conversely, all other parts of the world received cartridges whose notches are inset by about 1cm.

It is possible for software running on the N64 to detect what region of console it is using. Some titles were able to adapt to these changes and apply video and sound changes to work in both regions. Others were hard-coded to only accept one region.

It is possible to have the 64drive to "fake" the proper region, but since the actual N64 hardware has different clocking schemes, the video encoding (PAL vs NTSC color encoding) and audio timing will not be updated. The result is some minor audio pops, and not all TVs will be able to sync to the unconventional video signal.

More information on this subject can be found on the Internet.

Saving Methods

Games originally could contain custom chips that held game progress inside the physical cartridge. Appearing in decreasing frequency in production carts:

- EEPROM (4Kbit)
- EEPROM (16Kbit)
- SRAM (256Kbit)
- FlashRAM (1Mbit)
- SRAM 3x (768Kbit)

Generally, first-party titles utilized these kinds of built-in save mechanisms. As they added cost to each unit built, cheaper games relied on the user already owning a Controller Pak (which is also confusingly a battery-backed 256Kbit SRAM).

The last save type was used in only one game, *Dezaemon 3D*. This Japan-only title had three physical SRAM chips with some glue logic to do rudimentary address mapping.

The 64drive can emulate all of these methods. A look-up table is used by the menu to determine what kind of save type should be assumed for each image loaded. While this has been thoroughly tested, it is still possible to override.

There are prototypes in the wild that assume a certain save method and will crash if it can't be accessed. An example is *Mini Racers*. In the worst case you can try all 4 save types if a title hangs upon boot.

As detailed before, save files can be exchanged between emulators and the 64drive. All types except EEPROM files must be 32bit endian swapped when changing platforms.

Lockout Chips

In addition to the physical notches in each cartridge case, there was additional region enforcement in original games.
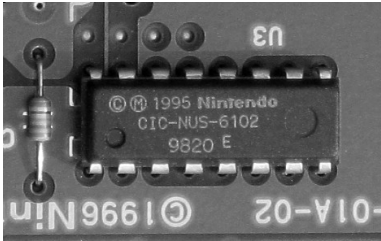
Each cartridge contains a small micro-controller (CIC) that maintains constant communication with a similar chip (PIF) inside the N64 console. If the chip is absent, the wrong type, or does not exactly match the region, the system will not boot.
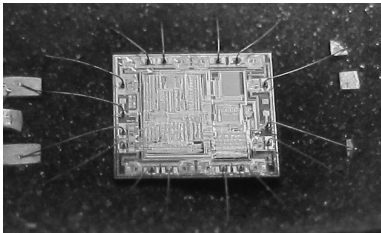
The known CIC types for NTSC are:
- 6102
- 6101
- 6103
- 6105
- 6106
- 8303 (64DD Japan)
- 5101 (Aleck64 arcade platform)

The known types for PAL are:
- 7101
- 7102
- 7103
- 7105
- 7106

Generally, the 6102/7101 is by far the most common. Star Fox/Lylat Wars was the first game to receive a new CIC, and the trend continued with each major first-party title.

Of special note is the x105 CIC, which contains additional functionality. It can accept a "challenge" request and provide a magic "response" value to the software running on the N64, if so required.

Jet Force Gemini was the first title to utilize this. In true Rareware style, the game would still run on non-x105 CICs, but it would crash randomly (on purpose) and disallow the player from jumping or shooting.

Banjo Tooie, another Rare title, took the idea further. All the game assets were encrypted, and during runtime the responses from the CIC challenges were used to decrypt them. This was not defeated until a few years ago.

Thanks to advancements made by the author and Mike Ryan since 2015-2016, 64drive HW2 now has a configurable lockout chip with cycle accuracy running on a PIC microcontroller.

FAQ

- Is the hardware region specific?
No, but the original gray donor cases will only fit their own region. US cases will only fit in US consoles, but European/Australian cases will fit into Japanese consoles.

- What SD cards are supported?
SD, SDHC, and SDXC cards have been tested up to 128GB. At least a Class 4 card is recommended. Class 10 is even better, and a quality Sandisk with UHS-I best.

- Can I use this with a Gameshark?
It is indeed actually possible. By carefully looking up the game's expected CIC and save type, you can set up the 64drive in USB mode to act just like the original cartridge. It's not recommended though, since Gamesharks have terrible connectors that tend to foul up every thing they touch.
Built-in Gameshark functionality is planned for future updates.

- What image formats are supported?
.Z64 and .V64 formats. Most .rom and .bin files fall under this category as well.

- Are there any compatibility issues?
All known N64 images will run, except if they required hardware not present on a retail N64 (such as clean Aleck64 dumps).

- Will hacked games run?
Depends. Many of these hacks, especially SM64 hacks, are poorly coded and only work because emulators are actually very inaccurate and let things slide.
A counter-example is Goldeneye X which since it is very well designed and executed runs on the 64drive with no problems.

- What about "high-res" texture packs?
Those are physically impossible to use on the N64, regardless of the hardware. The console's texture memory is limited to 4KB in silicon.

SPECIFICATIONS

Hardware (HW2)

- 256 Megabytes DDR2 SDRAM
- USB 2.0 High-Speed FIFO
- Secure Digital
- Custom FPGA logic
- 1Mbyte configuration flash and boot-loader
- ESP8266-based Wireless module
- Real-time cl

Hardware (HW1)

- 64 Megabytes SDRAM
- USB 2.0 High-Speed FIFO
- Secure Digital
- CompactFlash
- Custom FPGA logic
- 512KB shared bootloader
- 512KB configuration ROM

Transfer Rates (HW2)

- Memory card: 22Mbyte/sec
- USB: 20Mbytes/sec

Transfer Rates (HW1)

- CompactFlash: 9Mbyte/sec
- Secure Digital: 21Mbyte/sec
- USB: 8Mbytes/sec

Supported Save Types

- EEPROM (4Kbit)
- EEPROM (16Kbit)
- SRAM (256Kbit)
- FlashRAM (1Mbit)
- SRAM 3x (768Kbit)

TROUBLESHOOTING

Startup Problems

- I get the error "FAT32 partition not found".
  - → The card may be corrupted, or for-matted as FAT16/NTFS. Reformat.

- I pressed Reset after removing the card and it hangs.
  - → The cards are not hot swappable, power off and on.

- The spinny icon (bootloader) shows for a short time, then disappears to a black screen, and no menu.
  - → This may indicate a physical con-nection issue with the SDRAM. You can try loading an image over USB. Read the USB section for considerations.
  - → Possible incompatiblity with the memory card, causing corrupted data transfer. Email support

- "Undefined error" shows.
  - → This is intentional, and triggered by holding the L button. Try a dif-ferent controller. This may happen with cheap 3rd party controllers.