Handling D64 disk images on IDE64

by Paweł Król a.k.a. DJ Gruby/Protovision/TRIAD

Welcome to another article in the series of **IDE64** tutorials presented continuously on the pages of **Attitude** magazine. In this chapter I will provide you with all the necessary details you need to know about handling **D64** disk images and help you understand the mechanics of working with **D64** files directly from your **IDE64** device, regardless whether it is a real piece of hardware or only a **VICE** emulated environment.

As usual, writing this article has been inspired by **Visac/Cult,** who is an inexhaustible source of **IDE64** tutorial ideas. I hope that this text will become your primary reference when looking up any kind of **D64** related information and that its content will satisfy even the most demanding reader.

Throughout the years the **D64** format has become increasingly popular way of distributing and sharing C64 releases on other platforms than the C64 itself.

So, is it possible for your old hardware equipped with an **IDE64** disk drive to directly access any content stored on **D64** disk images: create, read, and write to them? Yes, of course! And everything you need is just a click away. This article will guide you through all the necessary steps you need to undertake to make it work on your personal setup.

So let us begin with a quick overview, what utilities exactly will be covered in this tutorial:

- **D64it** - an old (developed in the year 1998) but still very powerful and reliable tool to convert **D64** files from/to a 1541 disk drive.

- **ID64** - another program to convert **D64** files from/to a 1541 disk drive, very fast, however not as reliable as a previous one.

- **D64 list** - **D64** content browser, extractor and instant loader.

- **VD64 plugin 0.03** - virtual file system for the **MAN** file manager that also allows you to load files straight from **D64** images.

As always the **IDE64 Warez Site** [1] should be your primary source of information when looking for any **IDE64** downloads. It makes sense that you download all previously mentioned utilities before proceeding with reading this chapter.

Now, if you had already headed over to the **Appz** section of **IDE64 Warez Site,** you would probably have noticed two different versions of a **VD64 plugin** available. Versions 0.03 and 0.11. The first one works with **IDEDOS 0.90+,** the latter one only with **IDEDOS 0.91.** So why are we using an **old** 0.03 version instead of bravely proceeding with an upgrade to the latest and shiniest 0.11? Why even sticking to an old **IDEDOS** version, when there is a newer release available? The answer is very simple: stability. Please note that **IDEDOS 0.91** is still in an experimental beta phase, and we get warned before downloading it: *"IDEDOS 0.91 beta release is not very well tested and could destroy your data!"* This is a kind of notice that should be taken seriously. No, we do not want to destroy our data, therefore we cautiously stick to the latest stable version 0.90.

Before we start copying disks and disk images back and forth, we need to undertake a few preliminary preparation steps. Let us extract the content of all downloaded tools and look into what we may find useful. As this tutorial focuses solely on dealing with **D64** images, we will skip any unrelated files, but I guess you can easily imagine how similar a procedure to deal with **D71** or **D81** images would be.

The **D64it** archive consists of three files, but we are only interested in the program named **d64it1b** (we skip **d128it,** which is a native C128 version of **d64it,** and the **d64it.txt** note). Let us copy it into a separate directory called **utils.** This directory will contain standalone executables that do not require to be executed via the **MAN** plugin system.

The **ID64** package provides a couple of different but useful tools: **ID64 Reader** , **ID64 Verify,** and **ID64 Writer** plugins. Let us dive into the zip archive and see what interesting things we can find in there. The **plugins** directory contains a file named **d64.** This is an **ID64 Writer** plugin that we later are going to use to, as the name suggest, write **D64** images directly onto floppy disks. Extract the file into the **plugins** directory. We can safely skip all the other writer plugins, docs, and a registry configuration template. In the **utils** directory we can find two programs worth extraction: **id64 reader v0.6** and **id64 verify v0.6.** Let us move them to the locally created **utils** directory that already contains **d64it1b.**

D64 list provides the **d64l** plugin that we should copy over to the **plugins** directory. We are not interested in extracting other files, but feel free to analyse them carefully, as they might provide extremely valuable details of inner-workings of both the plugin system of **IDE64** and low-level access to **D64** image data directly on your C64 (given the fact that a source code has been made available for everyone and kindly included in this archive by **Fenek/Arise**).

Last but not least we look at the **VD64 plugin.** It is a "modern" way of accessing **D64** files from within the **MAN** file manager, as it implements a complete virtual **D64** file system driver for **IDE64.** At the moment only listing and loading of files are supported, but hopefully **Soci/Singular** will invest some of his time to further develop this plugin, as it looks very promising. It basically allows you to browse **D64** image contents as if they were regular **IDE64** directories. This is extremely convenient and makes **D64** usage even more comfortable than on your PC, where you still need to start up a specialised tool if you want to extract individual files out of your **D64** images. From this archive we will only extract a file named **vd64** and ignore the rest, as it was already mentioned earlier in this article that we focus solely on dealing with **D64** disk images, and not **T64's, D71's,** nor **D81's.** Put the extracted file into the **plugins** directory.

The next step requires us to set up a **man,usr** configuration file that enhances functionality offered by the **MAN** file manger to instantly recognise **D64** format upon pressing any one of the manager "function" keys (**2, 3, 4,** and **RETURN**) in the context of an indicated **D64** image file.

We do not need **MAN** to create **D64** images from our floppy disks, however we will need it to facilitate browsing of the contents and copying of our **D64** images downloaded from the Internet straight onto floppy disks. Let us have our configuration prepared already now at the beginning of the setup procedure, as you will most likely rush to inspect copied disk contents shortly after creating them.

This tutorial does not cover the details of **man,usr** configuration file setup. You can refer to the **IDE64 Plugins Setup** chapter published in **Attitude #14** to remind yourself all the details.

Here is a proposed setup you can use as an input for your configuration file generator:

```
1//plugins/:
sid *,sid
txt *,txt
vd64 *,d64
=
d64 *,d64
=
vd64 *,d64
=
d64l *,d64
```

Once executed using the familiar command **perl create_man_usr_file.pl config.txt,** you will get a new **man,usr** file generated in the current working directory. This file used together with **MAN** file manager on your **IDE64** device provides the following key/file associations:

- Pressing **RETURN** on any selected file with **sid**, **txt**, **d64** extensions will respectively trigger execution of the **sid**, **txt**, **vd64** plugins.

- Similarly, pressing keys **2**, **3**, **4** on any files of a **d64** type will respectively run the **d64**, **vd64**, **d64l** plugins using selected file name as their sole argument.

Now transfer all the files straight to your **IDE64** drive using either the CFS mount utility (see **How To Setup FUSECFS Driver?** chapter published in **Attitude #10** for more details on that) or simply by starting up an **ideservd** deamon and connecting a PC directly to your C64 via (USB) PCLink.

Make sure that you copy **man,usr** to the top directory of your primary partition. The same rule applies to your **plugins** directory. You can transfer the **utils** directory anywhere you want, as its location is not in any way bound to the plugin configuration of the **MAN** file manager, so you can simply execute those programs from any subdirectory. Also make sure that all transferred files have the "executable" flag set after copying, otherwise you will not be able to load them from your **IDE64** drive.

With all this initial effort being already put in place, it is time to act. I will show you how to accomplish the following operations directly from your **IDE64** device:

- Creating a **D64** image file directly from the content of a 5,25" floppy disk in your 1541 drive.
- Verifying that a created **D64** image has been correctly transferred from a source floppy disk to your HDD (and vice versa).
- Writing back any **D64** image (for example a trackmo downloaded from the Internet) onto a 5,25" floppy disk.

From a disk drive to a D64 image

Creating **D64** image files is pretty straight-forward. There are two ways to achieve it. Head to the **utils** directory you have copied over to your HDD during the preparation phase. You should see i.a. these two programs there: **d64it1b** and **id64 reader v0.6.** Let us begin with the first one. Select **d64it1b** from the directory listing and press **RETURN.** All of the available options should be self-explanatory. Most of the settings will require some modifications though, however at least source and target devices should be configured automatically from the start. You might want to change the target directory by pressing **-** (minus) and entering something like **0://stuff/import/** if the **utils** directory does not seem like the right location to create new **D64** files in (keep in mind that you can also use the **MAN** file manager later on to conveniently move **D64** files around). Any time you can press **F3** to verify that you are really writing into the desired directory by listing its contents. You will definitely want to change target file name by pressing **F5** and entering a custom name for your **D64** image. Now comes the most crucial part. You have to pay attention to the transfer **Mode.** Pressing **F6** flips between **1541 to d64 file** and **d64 to 1541 disk.** You want to always make sure that it says **1541 to d64 file,** otherwise the content of your floppy disk will be irreversibly destroyed. After checking that, there is nothing left to do but pressing **F8** to begin the transfer process. Since this program allows you to perform the reverse process of copying disk images from **D64** files to 5,25" floppy disks by simply flipping the direction with **F6,** you could be tempted to try it out immediately. Hold your horses for now, however. You'll soon learn that there are better ways to achieve the same result!

Now, you have surely noticed how slow **d64it1b** is. It is true that this program is not what we are used to in terms of disk copy speed. This is where **id64 reader v0.6** enters the stage. Load it from within **MAN** and try it once, I am sure it will become your program of choice when it comes to transferring your old 5,25" floppy disks. After starting a program you are prompted to interactively specify a couple of options. Target device is obviously drive number of your HDD (defaults to **12**, when in doubt consult the device numbers section of your **CMOS setup utility**). Then you enter a name for the target **D64,** answer yes or no to a question about copying 40 tracks (in most cases you will just press **N**), and finally specify device number of your disk drive (unless you modified your hardware configuration or connected more than one disk drive to your C64, this is always going to be **8**). As soon as you pressed **8**, the transfer process begins. Observe how fast it is. You will soon enjoy a backup copy of your disk, preserving it for future generations and maybe uploading its content to the CSDb.

Verification of a D64 image

It is time to verify that the transfer process did not end up in creating a damaged **D64** image file. Go to the **utils** directory in the **MAN** file manager and execute the **id64 verify v0.6** program. The procedure is similar as in the case of the reader utility. You are interactively prompted to specify a couple of options before the actual verification process begins. As the source device you specify drive number of your HDD (**12**), source **D64**'s name is naturally the image file you created in the previous step using the reader program, then again you answer yes or no to the question whether you want to verify 40 tracks, and finally specify the device number of your disk drive (**8**). If everything goes fine, in about 20-30 seconds you will see a confirmation message: *"Verify passed ok... bye."* Now we are ready to get back to the **MAN** file manager and look at the content of the **D64** image files as well as transfer some of them to a 1541 disk drive.

From a D64 image to a disk drive

Browse into a directory (no, not **plugins** directory!) where an interesting **D64** image file is located. We will inspect it and write it back to a 5,25" floppy disk. After selecting a file using the cursor keys and pressing one of the **MAN** manager shortcut keys, you can check what your **IDE64** device has got to offer for the **D64** file type.

Pressing **2** loads **id64 writer v0.6** utility from the **plugins** directory and sends the selected **D64** file name as its sole argument. **ID64 Writer**, as the name suggests, is used to copy **D64** image file directly to a 5,25" floppy disk of a disk drive. It works in a similar fashion to the already discussed family of **ID64** tools (**reader** and **verify**). After starting the program you are prompted to interactively specify a couple of options. First comes the target disk drive number (**8**), followed by whether you want to copy 40 tracks, and another question about formatting target floppy disk first before transferring the data (and when answered positively, another prompt appears, asking for a source of a disk ID). Please note that disk formatting takes a little while, and is slower even than the copying process itself. When done, you can use the verification procedure described in a previous section in order to validate the copy.

What is in the D64 file?

Pressing **3** or **RETURN** opens the content of a **D64** image directory directly in a manager window. This is possible since the **MAN** file manager supports implementing virtual file systems. You can browse the content of any **D64** disk image this way. Unfortunately, the possibilities of **VD64 plugins** are still very limited and do not allow you to, for example, copy files from/to a **D64** image. Additional support for copying individual **PRG** files from/to **D64** images would be great. If you think we are asking too much, consider the fact than an experimental **VD64 plugin** version for the unstable **IDEDOS 0.91** supports this and even more features already. I keep my fingers crossed for **Soci/Singular** to eventually backport copying functionality into a stable version of a **VD64 plugin**, or even better to fix all the outstanding issues with the new **IDEDOS** system and to eventually declare it stable after many years of development. It will surely be a big milestone in the **IDE64** development efforts, and an important day for the entire community of **IDE64** users.

Pressing **4** loads the **D64 lister** plugin, developed in 2006 by **Fenek/Arise**. It is a fairly old tool, but it still gives you an excellent deep view into the heart of your **D64** images. It is also very powerful, as it implements a feature that is still missing in the **VD64 plugin**, namely extracting individual files from disk images. Press **INS/DEL** to select/deselect individual disk files for extraction, **+** (plus) to select all files at once, **-** (minus) to deselect them, ***** (asterisk) to invert current selection, and finally **E** to extract all selected files to the current directory (the directory where the current **D64** disk image is located).

You can of course load and run the selected file while still browsing an image directory simply by pressing **RETURN**. Another excellent feature of **D64 lister** lets you fetch and display load addresses of all files on disk simply by pressing **A**. Pressing **RUN/STOP** lets you return to the **MAN** file manager.

Summary

I have not covered every single existing way to access **D64** image files directly from your **IDE64** device in this article. You can browse the **IDE64 Warez Site** at any time if you crave to discover more options. If the ones described here do not appeal to you, you might be lucky to find more suitable tools. The utilities I covered in this chapter are my personal and arbitrary choices, which I am strongly inclined to recommend you. They serve me well, and I am sure they will do the same for you. With direct **D64** read/write access on your **IDE64** device, you can introduce an even better directory structure on your HDD and experience more pleasure from satisfactory use of your hardware.

Good luck!

Paweł Król a.k.a. **DJ Gruby/Protovision/TRIAD**

References

[1] IDE64 Warez Site http://singularcrew.hu/ide64warez/index2.php