"APA Partition Logic"

With the launch of the PS2, Sony was the first to introduce the hard drive to the console market. Due to the already absurdly high price, it was then decided to use HDD as an optional accessory, attached to an optional network card called Network Adapter. The disk served as a den for operating systems, firmware updates and, in a few exceptions, also for saves, games and their add-ons. Of course, this is not the same level of integration as on the Xbox, as it was neither a mandatory part of the console (except for DESR xxxx models), nor even popular with gamers, and therefore also game developers. Us, i.e. the enthusiasts of the so-called scenes, of course, of interest from a different angle, i.e. how it can be used in a way unforeseen by the creators of the console.

The following wall of the text deals with several different issues, but they overlap so much that I decided that it makes no sense to separate each of them into a separate guide. Therefore, everything together forms a coherent whole, but you do not have to install everything, or even do the same as me, because there are many HDD images with already uploaded and configured wonders by. the whim of the authors. The information I have included here is for people who value order - and as always - enthusiasts of the so-called scenes with too much free time. ;)

Ways of connecting HDD

Officially, the hard drive can only be connected to "fat" PlayStation 2 models, only with a device called **Network Adapter,** which, depending on the PS2 model, takes the form **of a network card** (with or without built-in modem).

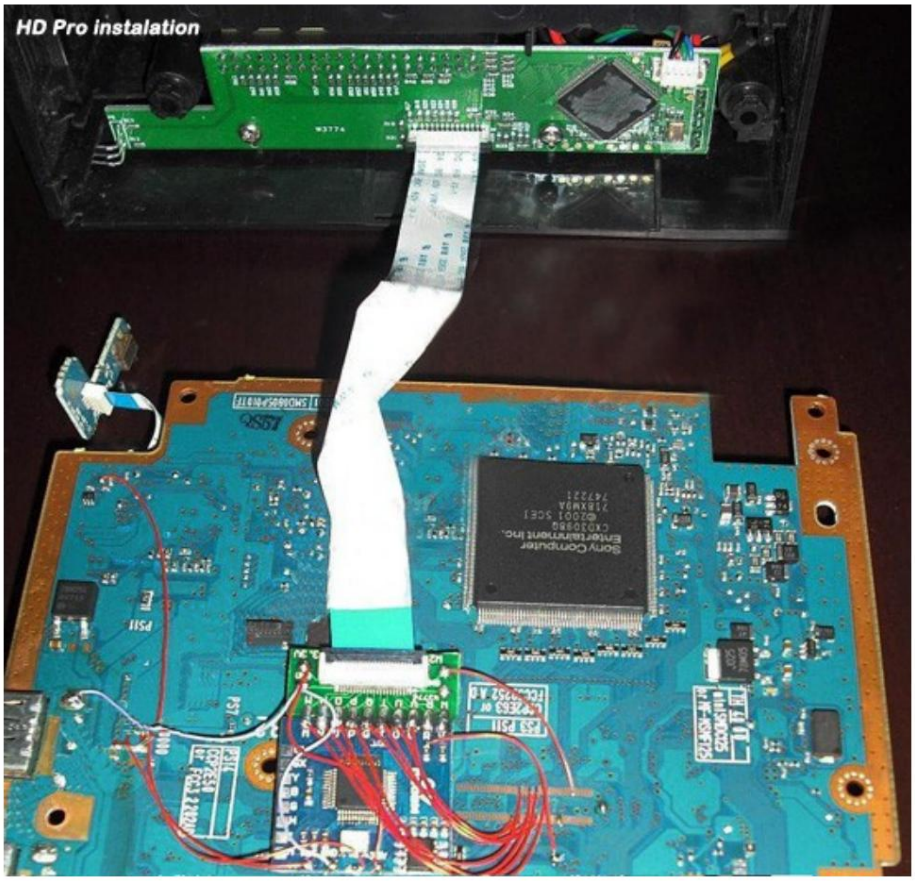
- In the case of **SCPH-1xxxx** series consoles (e.g. SCPH-18000), a card inserted into the console, connected via **PCMCIA** - and in this case the hard drive is external, enclosed in a special casing with an attached cable to the NA (obviously not to be confused with the external to USB).

- In the case of consoles from the series from **SCPH-3xxxx** to **5xxxx** inclusive (e.g. SCPH-30004), a slightly crude card attached to the console through a special port, and the hard disk is connected to the NA through a standard pair of **PATA** and **Molex** (unfortunately not every HDD has the same port spacing). The whole thing slides into the depths of **Expansion Bay,** where the disc is hidden in the console, and the NA sticks out like a hump on the back.

Unofficially, the hard drive can still be connected to the **SCPH-70xxx by soldering a stern** cobweb to the motherboard, because in these models the IDE controller is still present on the MOBO and still supported in the PS2 firmware (only the port has been removed).

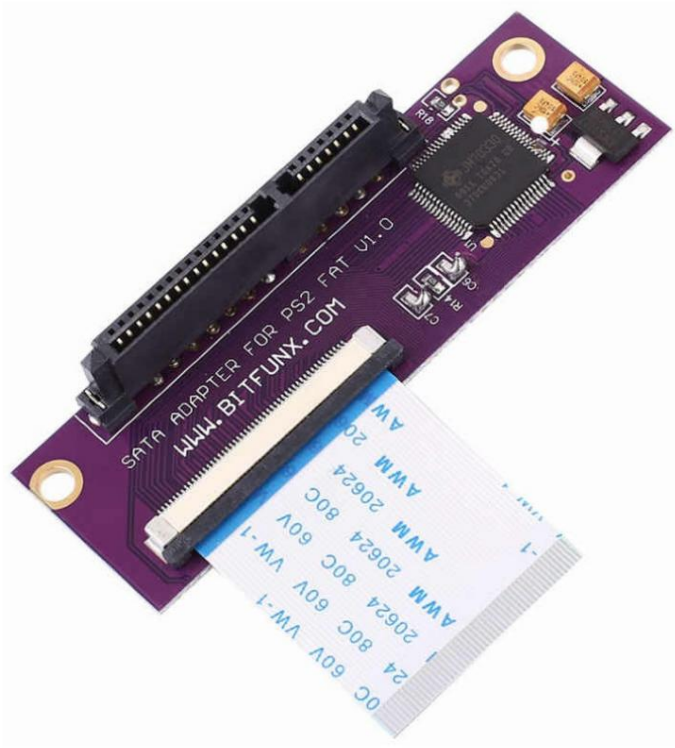
In newer versions (i.e. from **SCPH-75xxx)** there are already solutions via modchip, which deals with communication with an additional IDE controller and fw patching.

Alternatively, the Hard Disk can be connected via USB or i.Link, but this is a completely different tutorial, because only the one connected to the Network Adapter or MOBO can be treated as internal. For the external one, the storage logic is based on FAT32 or EXT2 on the MBR, while the internal one is mainly based on PFS on the APA (you can read about it on the following pages).



HD Pro for SCPH-75xxx models.

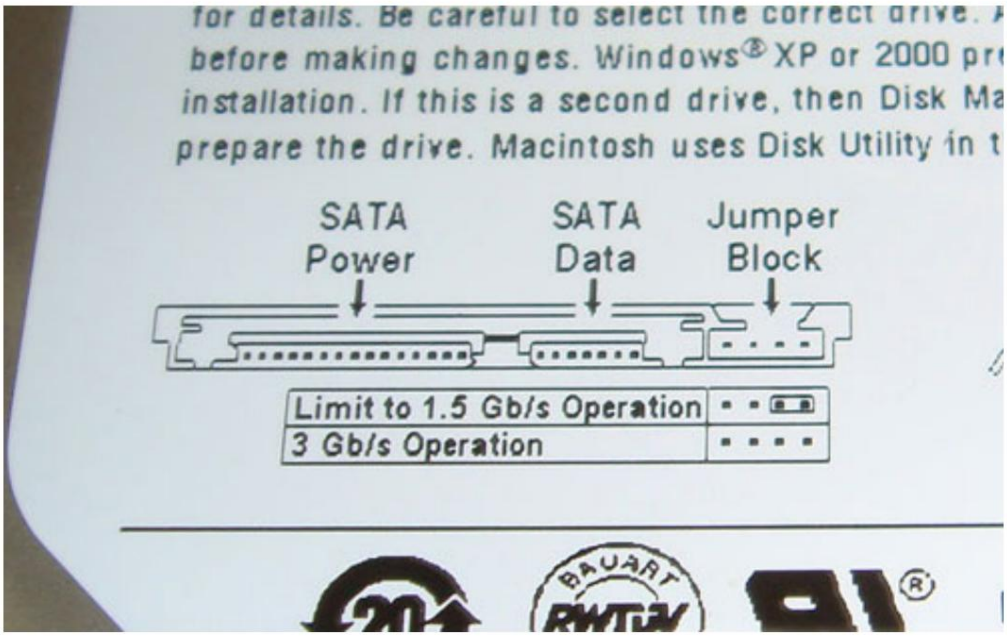
The hard drive must be on **PATA,** with the jumper set to **MA** (Master) plus possibly additional limiting capacity. Although there are different controllers that allow you to connect SATA drives (such as, for example, unofficial Network Adapters, which have SATA connectors instead of IDE and Molex), it is different with them, because even with tape drives, compatibility is not 100% ... At one time, various lists of HDD models were posted on various websites and message boards, with or without problems, but neither all reports were reliable, nor, for obvious reasons, complete lists. The hard drive I am playing with is ST3402111A, connected via the original NA, on the PS2 SCPH-30004R model.



Chinese Network Adapter guts replacement, replacing PATA with newer SATA.

If you decide to go with a SATA hard drive, unfortunately, there are problems related to the speed of data transfer. In the case of ATA-300 (and probably also ATA-600), games loaded as partitions (i.e. launched using Open PS2 Loader or HDLoader) may hang without setting the MDMA2 mode, while games from PSX launched via the POPS emulator, fail at all won't run.

If your SATA-II or III HDD has the ability to release it to ATA-150 (this can also be called 1.5Gb / s mode) using a slide-in jumper (similar to PATA drives for Master, Slave, Cable Select and optional LBA limiting), it is worth using and then POPS will not be fussy, and where it can, UDMA mode should be used instead of MDMA.



Not all HDDs have a description of how to set jumpers. Information on this subject can be found on the disc manufacturer's website, of course in the instruction manual for a specific model.

No HDD

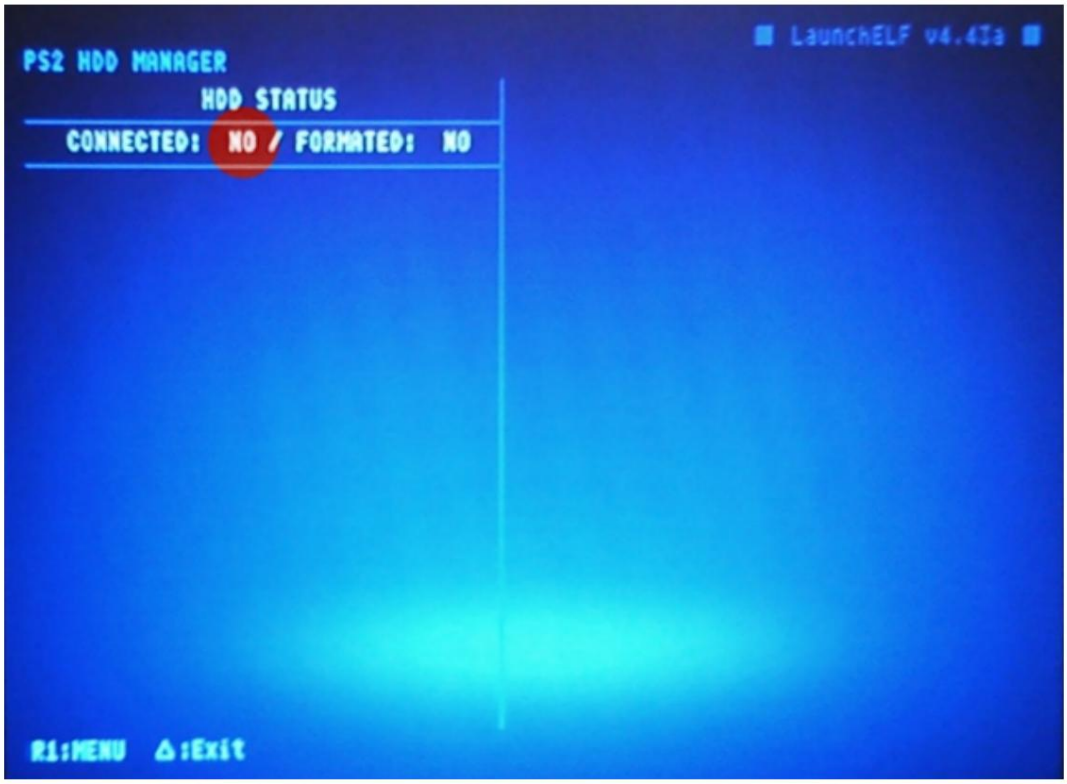
The console may not detect the hard drive if the jumpers are incorrectly set. As mentioned above, there must be at least one that sets the disk to Master (MA). It also happens that the lack of jumpers is MA. It all depends on the specific HDD model. Information about the configuration of PATA drives is always found on the label attached to the case.

The second possible cause is inaccurate insertion of the disk into the Network Adapter. Especially if the spacing of the Molex and IDE does not match the spacing of the plugs on the HDD side, and the user had to struggle to push it there. ;) Sometimes they do not fit at all and the user saved himself by knitting bridges to the torn out sockets, which do not withstand too many withdraw / insert cycles.

The third reason I can think of is a well-established Network Adapter slot in the console. It's fairly shallow and fragile, so the less you take it out of your PS2, the better.

The fourth reason is insufficient tightening of the Network Adapter to the console. Theoretically, NA does not even have to be screwed with screws, but it is better not to give up on it, because even the smallest looseness may result in the HDD being not detected.

The last cause may simply be a corrupt Network Adapter. Never remove it when the console is turned on (not even in standby mode).



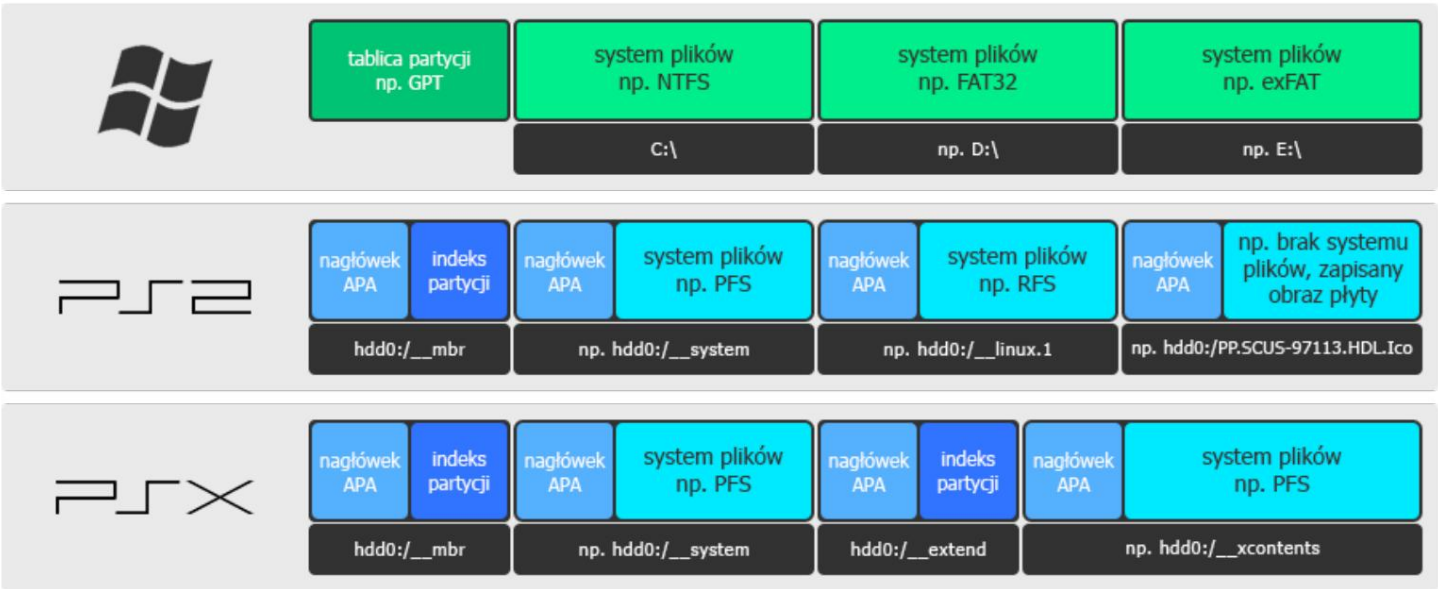
Logical structure

As Sony is used to, and this time it did not use one of the standard open formats, but designed its own, but at least training the imagination. ;)

Partition Table?

PS2 does not use any partition table. Instead, it uses an index with the partitions created, but encapsulated like all the others with the **APA** (Aligned Partition Allocation) header. This format allows you to save partitions in a rather extravagant way, because blocks where the smallest possible unit is **128MiB** (so the partition size must be exactly that much or a **multiple of it),** and the maximum is **128GiB.** In the header of each partition, in a special area called **Attribute ,** you can find information about the name, partition type (whether it is primary or sub-partition), you can also find icons to display in the console menu (PS2 Browser), and even an executable file.

To better understand the difference between the standard partitioning model and the freak that Sony developed, here is a small comparison. In the case of Windows, for example (green), the "partitionology" looks like there is one partition table with information about the space used by at least one partition. In the case of PS2 and PSX (in blue), you are dealing with a sequence of partitions where the index (or indexes) of the partitions created and each filesystem is encapsulated with an APA header.



Original Sony software was always designed for the official hard drive and theoretically it does not support any other. This is related not only to "invisible" HDDs other than the original one, but also to limiting the capacity to **LBA28** - that is, humanly speaking - of 128GiB (~ 137GB). Currently, all the scene software supports **LBA48** 2TiB (~ 2.2TB), including Sony firmware hacks, but formerly of course not and for a short period of time, the only way to get around this limitation was to use **APAEXT** (Toxic Extended APA). TeamToxic, the creators of the well-known series of modchips, implemented in their loader (ToxicOS) reading the cloned first partition just after the range for the standard index ends. In other words, in the case of APAEXT, there are two indices, one for the first 128GiB and the other for the space outside of it. The advantage of the extended APA is that it is compatible with everything that was created for LBA28, but the disadvantage is that it is incompatible with the software supporting LBA48 (you cannot format the HDD then, because you will lose partitions listed in the second index, or create partitions exceeding the 128GiB barrier).

It's funny that a few years later, Sony did exactly the same implementing hard drive support in **DESR-xxxx** models (i.e. special **multimedia PS2,** perversely named **PSX,** and colloquially PS2 **DVR). :)** As in APAEXT, also here the logical structure of the disk is divided into two areas, but this time one up to 40GiB, which covers the first index, and another above 40GiB, up to possible 250GiB, which covers the second. This variant is conventionally called **APADVR** to distinguish it from the standard and "toxic" version (officially there is no name and for Sony it is simply APA).

File systems

The main file system used by most PS2 software is **PFS** (PlayStation File System?). Today, this abbreviation is associated with a special encrypted container used on PlayStation Vita and PlayStation 4, but in the days of PS2 it was the official name of their hard drive file system, which has absolutely nothing to do with newer consoles. Apparently, its structure resembles the EXT2.

System partitions are all whose names begin with a double underscore, so according to the scheme: **"__ <label>"** or **"__. <label>".**

The first and only obligatory one is "__mbr" which, exceptionally , has no filesystem. This is where the signed loader is loaded (with the keys that we know since breaking the PlayStation 3), and which in turn can also run the signed next loader (e.g. Linux boot manager with **Linux Kit** or **PSBBN** (special distribution, but different than officially sold)). Its most important function is, of course, the index of the partitions created (equivalent to the partition table). In case of APAEXT application or on DVR models, additionally also the first and obligatory partition is "__mbr_ext__" (immediately after the first 128GiB) or "__extend" (here after the first 40GiB).

The remaining partitions are a mandatory set of, among others for **HDD OSD** (in a way it's a firmware upgrade) and **PSBBN:**

__net	contains information related to DRM, e.g. games downloaded from PSBB channels,
__system	contains firmware and loaders resources, includes firmware resources and configurations
__sysconf	
__common	contains user data, such as save files,
__contents	contains user data, used only by PSBBN

In the case of a DVR, you can find two more partitions. It is true that almost the entire disk is encrypted, but it takes place at the hardware level, so this encryption is completely transparent to the running software (thanks to which it is possible to make a sectoral, decrypted image from its level).

__xdata	contains temporary data, e.g. downloaded firmware updates
__xcontents	contain user data such as music, photos, videos, XMB thumbnails

Additional partitions with underscores are created by homebrew programs. For example, older versions of uLE, after formatting the disk, still created the "__boot" partition, which was used by modchips for the so-called DEV2 (Dev.olution Mode 2, ie starting the "BOOT.ELF" file from there). I met with "__main" yet, but I do not know what it is for because it does nothing that is covered in the guide you are reading.

Deleted partitions are flagged as empty and re-labeled with "__empty". Homebrew for PS2 ignores them, but various PC scene programs allow you to recover or restore data from them.
The "_tmp" partition is still possible , but it is only used by games for temporary operations and is removed at console startup.

Until now, all **homebrew partitions,** that is, those created by unlicensed programs, had a plus sign in the label preceding the chosen name. However, since uLE has had "forks" such as **double unofficial LaunchELF** (wLE for short - both of these names are unofficial) and **unofficial LaunchELF kHn,** then you can create partitions with any label. All other programs create partitions by. schema: **"+ <label>" (eg** "+ OPL", "+ ScummVM" etc.).

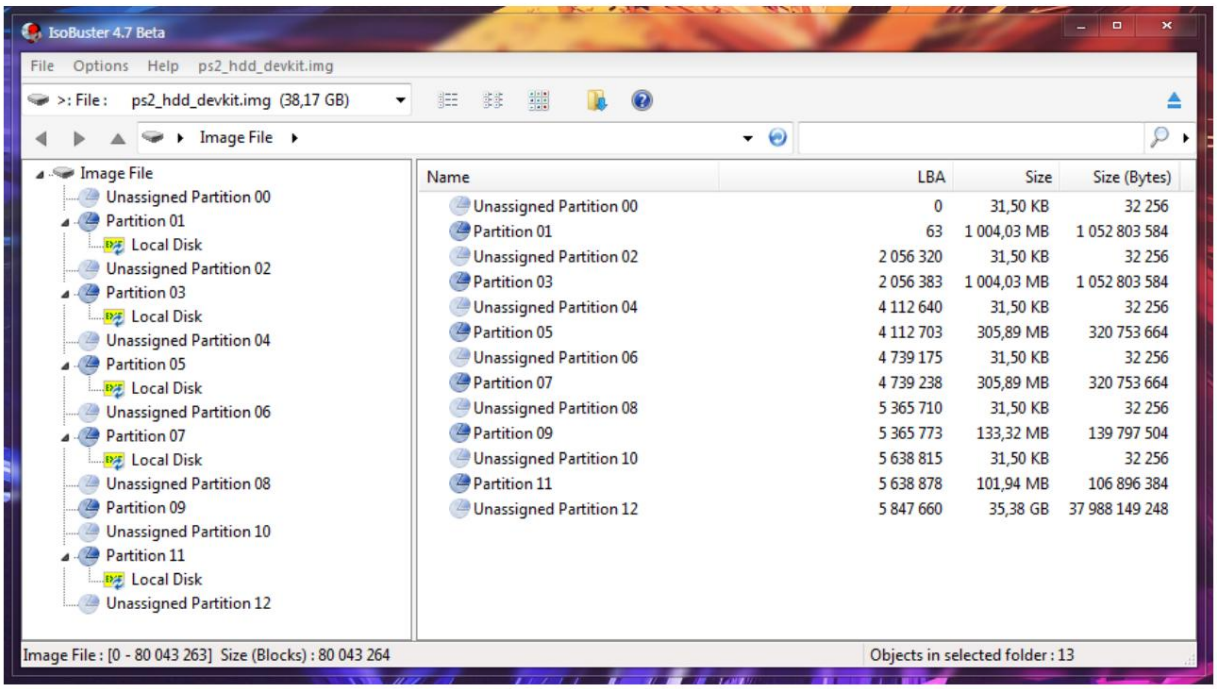
Game partitions, i.e. those created by official games (for their installation data and add-ons) or PSBBN, have their specific **PP prefixes.** and **PC.** (depending on whether they serve as **Partition Parent** or **Partition Child).** So it will always be **"PP. <label>"** and **"PC. <label>".** The HDD OSD treats all "children" as a whole and displays only the "parent". As you probably guessed, this naming system was used by developers to upload games and "link" DLC to them. Apart from "__common", these are the only partitions displayed by "Browser".

All the above-mentioned partitions (except "__mbr" and "__extend") use the PFS filesystem, so they can be browsed and freely managed with the file managers on the console side as well as with the **PFS Shell** on the computer side.

There is one more, special type of game partitions, i.e. unofficial ones, created with **HDLoader** and **Open PS2 Loader in mind.** They don't have any filesystem and contain sliced disc images uploaded as a whole partition (of course with zeros rounded according to APA requirements). Games released on CD must be converted from mode2 (blocks of 2352 bytes) to mode1 (2048), i.e. in the same form as those on DVD. Session / track information is also modified so it is not a perfect 1: 1 copy. Games uploaded in this way can be tweaked using, for example, **IsoBuster** (from version 4.7) or **WinHIIP,** and uploaded with the software that can be found in the chapter entitled "Copying PS2 Games to Hard Drive".

PSBBN partitions are exceptionally located on **RFS,** and although it is a **standard** file system, it is in its first, still then commercial version.

The Linux partitions (official except PSBBN) in turn lie on **EXT2,** which is also the **standard** filesystem and has full support for any Linux on any platform. Of course, mounting on a PC is possible, but after specifying the address where it starts from or after extraction and hooking it as a loop device. The exception here is **PS2Tool** (i.e. **devkits** DTL-Txxxxx) which do not use APA, but (almost;)) standard MBR.



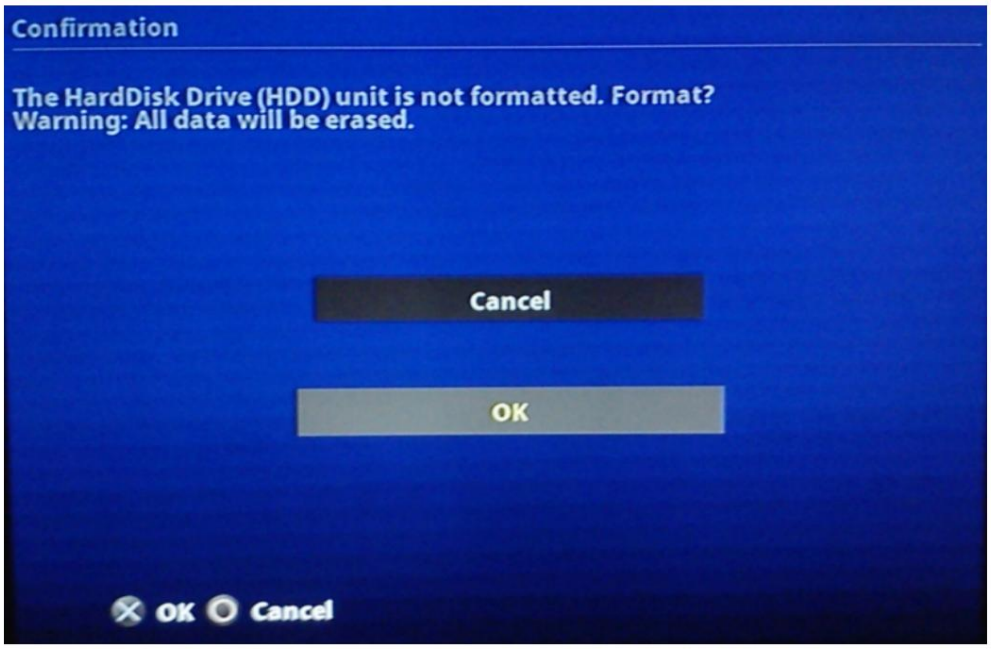
Partition mosaic from the DTL-T15000 console disk. For some mysterious reason the partition table is MBR, but it allows more than 4 primary partitions (no, it's not GPT), all intertwined with 31.5KiB blobs.

HDD formatting

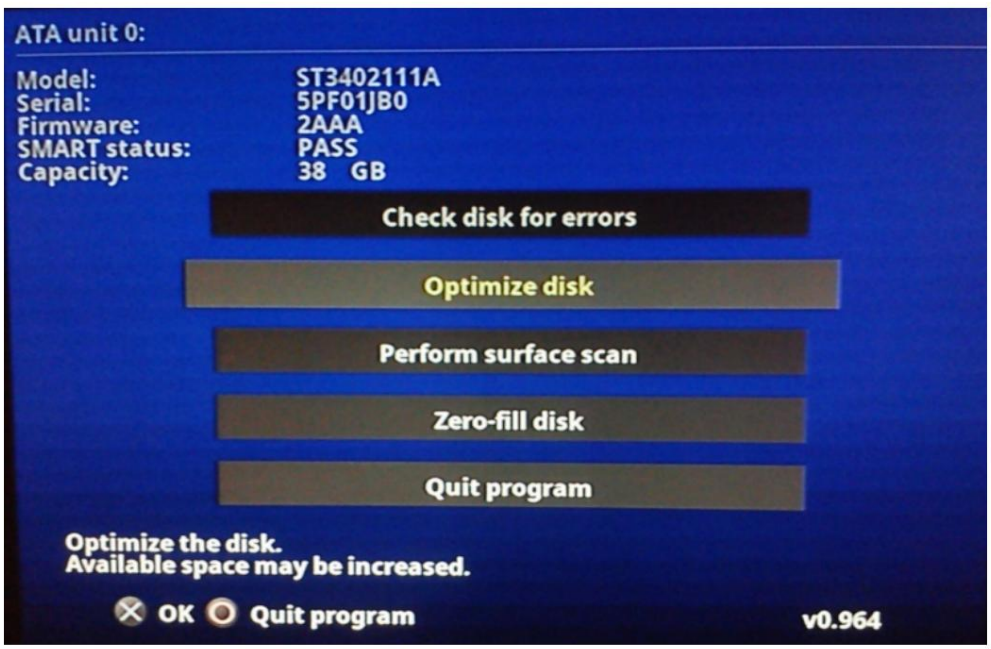
You can format the hard disk in many ways, such as using uLE (in the "PS2 HDD Manager" menu), and even on a PC (e.g. using **PFS Shell** and the "initialize yes" command), but in my opinion the easiest and fastest way is to use **HDDChecker ,** which has many more unique and useful functions.

Note: if you have any DESR model, you must not do it because you will irreversibly replace APADVR with ordinary APA, and thus you will lose the operating system without which the console will not start.
This can still be fixed if you have a copy of this data but you do not restore APADVR and all DVR functions will no longer work.

After starting the program, in case the disk does not have APA and primary partitions, you will immediately be asked if you want to format it.



But if you have already used this disk in the console and would like to clean it, it is best to select the "Zero-fill disk" option, thanks to which all sectors will be overwritten. If you only want to format, then after selecting this option, wait a few seconds and cancel (this is enough to paint over "hdd0: / __ mbr" to make the disk no longer recognized) and then you will be asked about formatting.



Free HDBoot

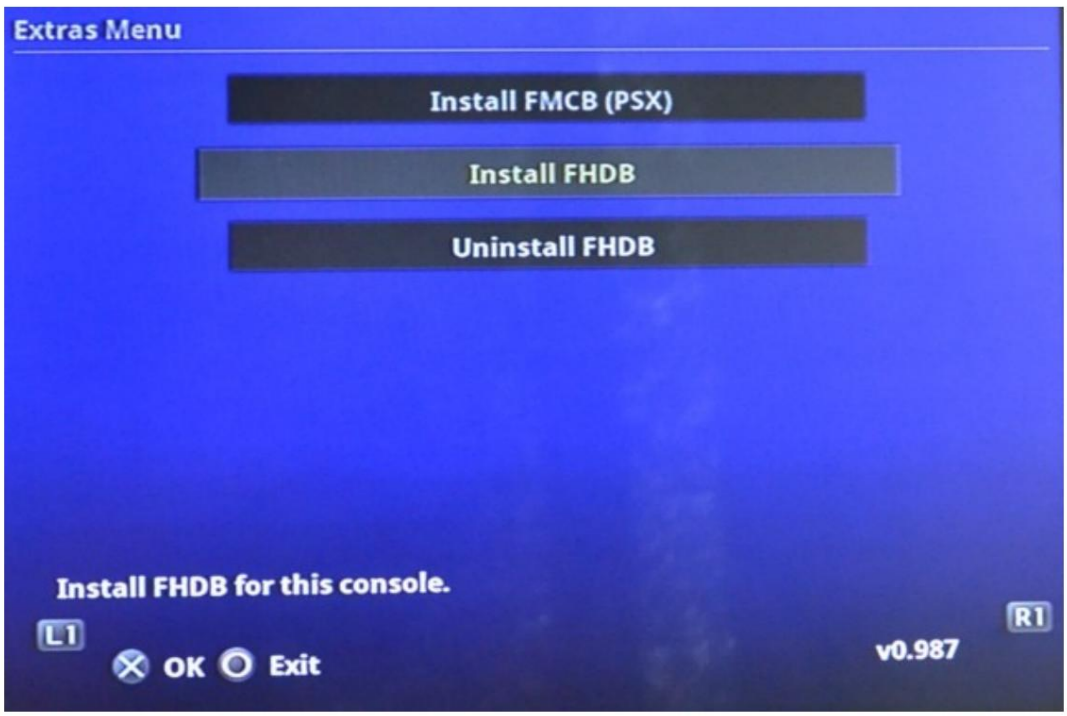
PS2 firmware is stored on ROMs, and therefore impossible to reprogram. Since you cannot update the firmware, you had to leave the door in the form of loading programs and their resources from the outside. And so decided the engineers designing the console, who left the ability to load the stuff from the memory card (which many years later exploited VAST, Memory Plus, Memor32 and finally Free McBoot) or the hard drive (which uses Free HDBoot). This hack is therefore the basis for the next optional (even if you are not interested in FHDB), because it automates the process of compulsory "__mbr" preparation, so that it searches for the next loader already on the system partition.

Note: if you have a **modchip** soldered in the console, the FMCB / FHDB after installing on the card / disk will either hang at startup or will not allow any program to be launched, displaying a black screen. Some modchips have the function of temporarily turning them off [1] [2] and many simply cannot be turned off. Apparently, some older versions more compatible with the chips (e.g. **1.965** and **1.963),** but in my case also with the above-mentioned chips. there are problems (last compatible version was **1.8b / 1.8c,** but these don't have HDD support). The best solution would be to simply disconnect the modchip's power supply or unsolder it completely.

1. Run **the Free McBoot & Free HDBoot installer** necessarily from a pendrive, because it will not work from a disc, memory card or hard drive (at least version **v1.966 (2019-04-13)).**

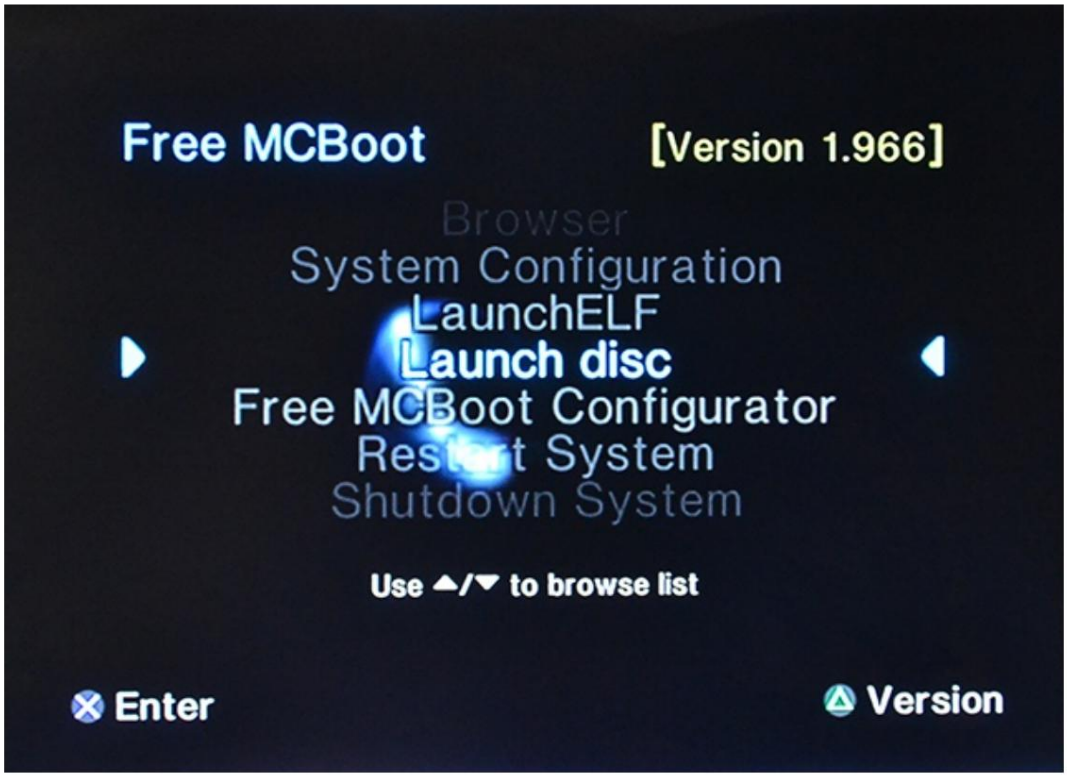
2. Press the R1 button to enter the HDD installation menu. Select the option "Install FHDB" and of course confirm with "Yes".

If you put any programs into the "\ INSTALL \ APPS-HDD \" folder , the FHDB will create an additional 128MiB partition "hdd0: /PP.FHDB.APPS/" to which it will copy these applications.



3. When the progress bar reaches the end, you will get a message saying that the installation is complete. Go back to the main menu and select "Exit". The console will turn off.

4. When you turn on your PS2, you should be greeted by a slightly modified menu.



On the SCPH-10000 and 15000 models, there will be a problem with the display of special symbols from SJIS (the left arrow will be in a different orientation and the lower right arrow will be in the background), similarly if you load the FMCB configuration file on the FHDB (and vice versa). The solution is either to play the HDD OSD, or simply to remove all these (in my opinion completely unnecessary because ugly) fountains, that is, make the menu similar to the original one or replace the appropriate characters in the "hdd0: / __ sysconf / FMCB / FREEHDB.CNF" file (but do it either by using FMCB Configurator or by editing the file on PC with a text editor; do not copy this from the tutorial you are reading now because the encoding is completely different here).

FROM:

```
OSDSYS_left_cursor = •
o006 OSDSYS_right_cursor =
• o005 OSDSYS_menu_top_delimiter = • y-99Free HDBoot • c1 [• r0.80Version% VER% • r0.00] •
y-00 OSDSYS_menu_bottom_delimiter0.se • o y009 • 99 o010 to browse list • y-00 • r0.00
```

On:

```
OSDSYS_left_cursor = •
o009 OSDSYS_right_cursor =
• o008 OSDSYS_menu_top_delimiter = • y-99Free HDBoot • c1 [• r0.80Version% VER% • r0.00] •
y-00 OSDSYS_menu_bottom_delimiter0.se • o y006 • 99 o007 to browse list • y-00 • r0.00
```

You can find all FHDB files in:

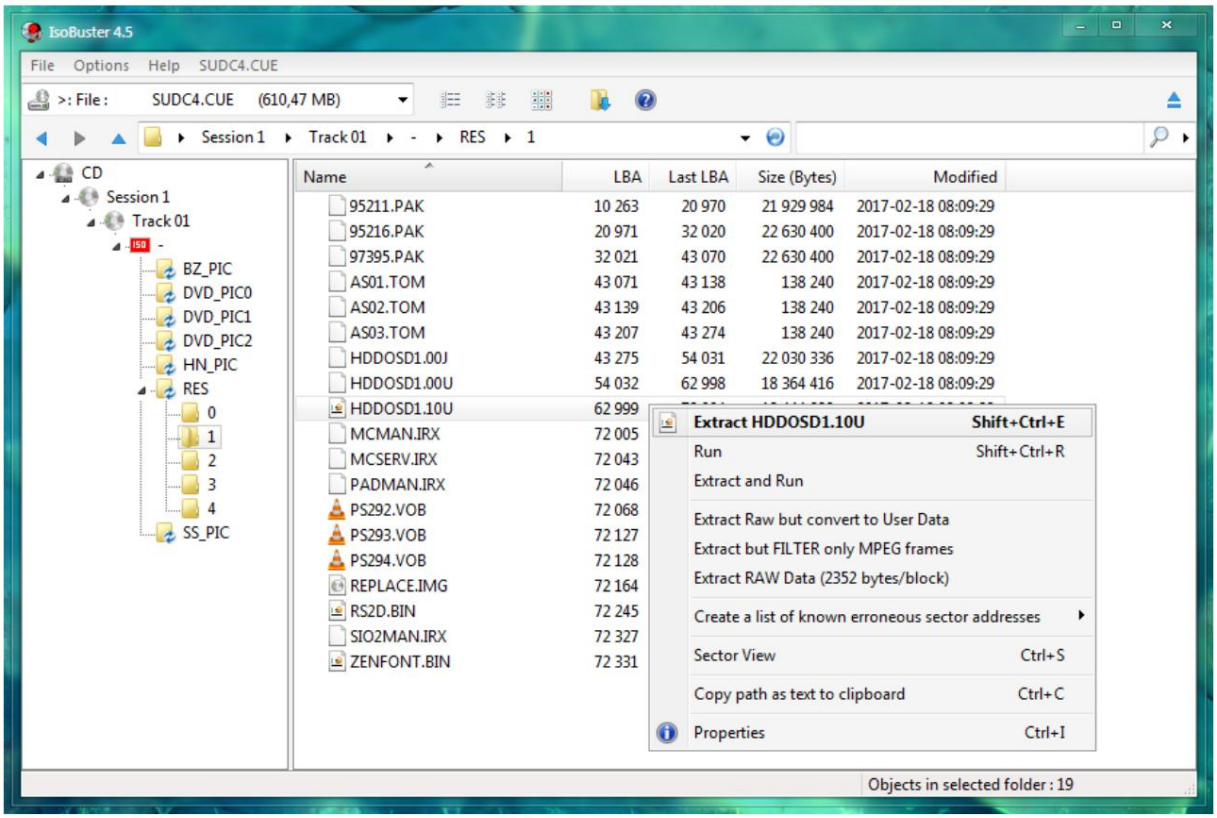
"hdd0: / __ sysconf / FMCB /"	program and configuration file plus additional modules file
"hdd0: / __ system / fsck /"	system checker
"hdd0: / __ system / osd /" "hdd0: / __ system / osd100 /" "hdd0: / __ system / osd110 /"	Free HDBoot, which is de facto "osdmain.elf" or "osd100.elf" or "osd110.elf" file

HDD OSD

PS2 Browser v2.00, commonly known as HDD OSD (or HOSD from the file name "hosdsys.elf") is a console firmware update for hard disk owners. It allows, for example, to manage "game partitions" or to copy saves from the card to the disk and vice versa. Unfortunately, the recognized partitions will be only those that are not systemic, and in addition they have the field with attributes filled (icon etc., which e.g. uLE does not do, so such partitions will be displayed as Corrupted Data).

The HDD OSD was included with the **HDD Utility Disc.** The problem, however, is that these discs formatted the entire medium, and the update itself worked only on the original HDD and in addition were without LBA48 support (i.e. capacities above 128GiB). That is why I base the following description with heartache on the pirated version of "everything-having", but most of all patched, that is, without the above-mentioned idiotic limitations. Unfortunately, the compilation of these tools under the name **SUDC4** (Sony Utility Discs Compilation) you have to find yourself in the dark corners of the Internet. ;) The MD5 checksum for the disc image is: A7901D0EEFBA48960825A55F157F97EC.

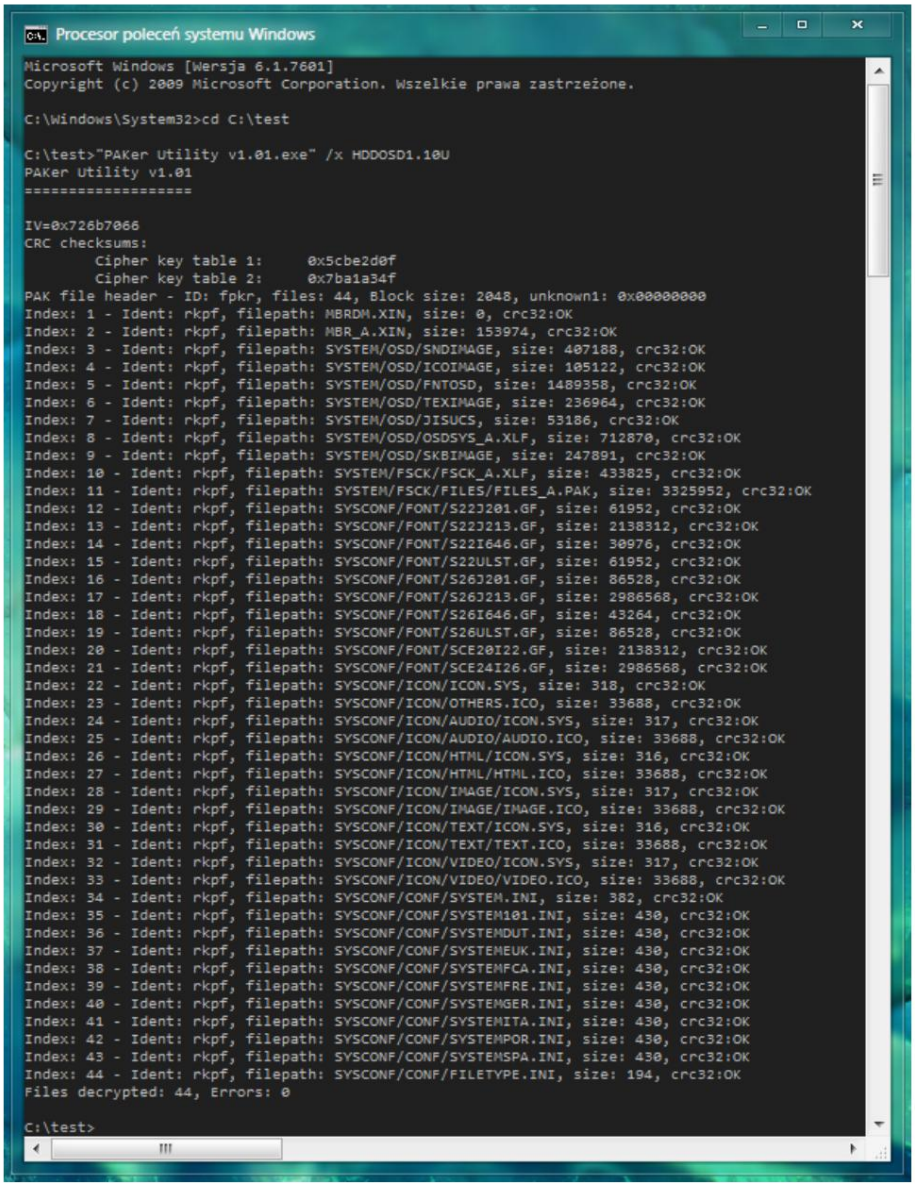
1. Open an image with eg **IsoBuster** and extract the "HDDOSD1.10U" file.



2. The package copied from the CD image is in the PAK format, used by Sony's PS2 utilities.
So you have to unpack its contents to be able to manually transfer the update to the HDD.

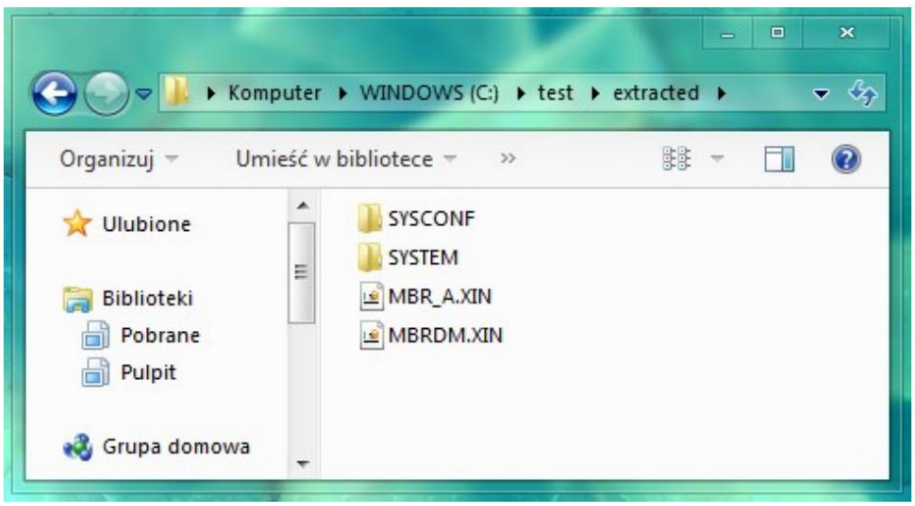
This is what the **PAKer Utility** program is for , which does not have a graphical interface, so either run CMD or PowerShell first or write an appropriate * .bat.

The syntax is very simple: "<program name> / x <file>".

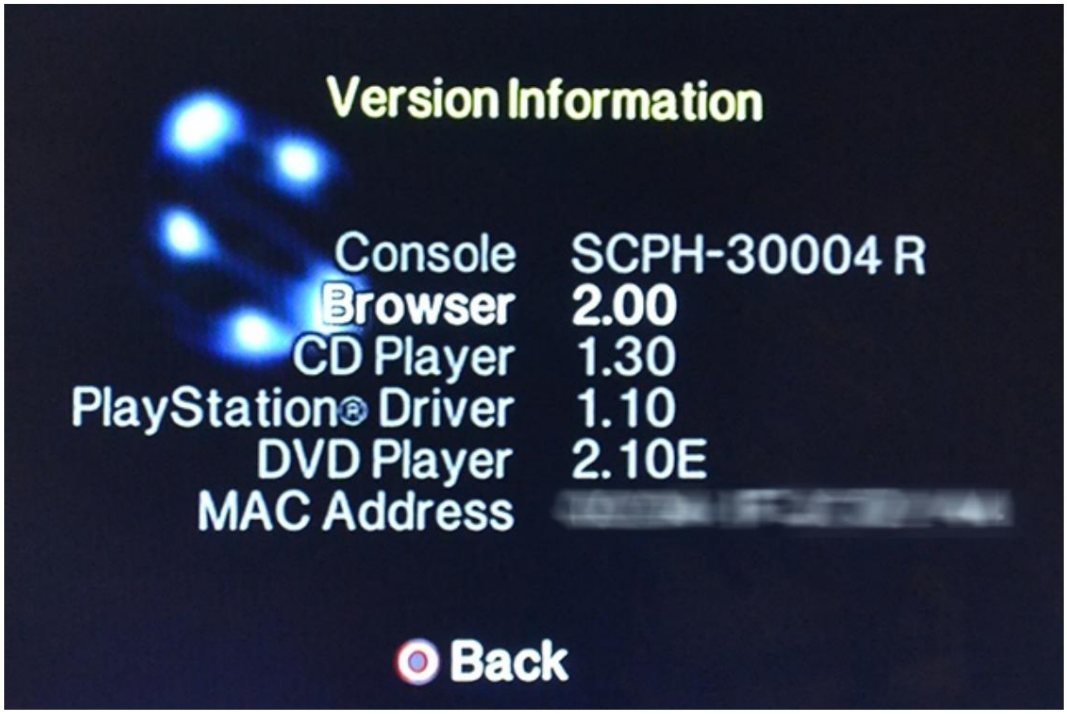


3. In the place where you started the program, the "extracted" folder will be created, and in it files and folders that should be transferred to the PS2 hard drive (using uLE, for example, from a USB flash drive (i.e. from "mass: /") or via with **PFS Shell** on Windows after connecting the drive to the computer).

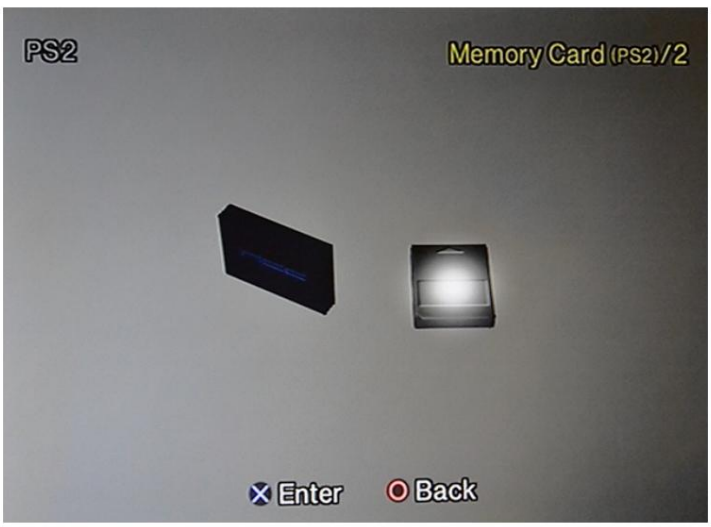
"MBR_A_XIN" and "MBRDM.XIN" can be thrown away, **the contents of** "\ extracted \ SYSCONF \" should be copied to "hdd0: / __ sysconf", and "\ extracted \ SYSTEM \" to "hdd0: / __ system". Additionally, the file name from "OSDSYS_A.XLF" should be changed to "osdmain.elf", "osd100.elf" or "osd110.elf" (if you want to replace FHDB) or "hosdsys.elf" (if you want FHDB to adopt HDD OSD) .



4. If you did everything correctly, after starting the console, you will see Browser version 2.00 in the properties, not earlier.



If you do not intend to upload programs and games to prepared partitions (more on that later), so that they imitate the original partitions and thus run them directly from the "browser", the sense of installing HDD OSD is negligible. Even saves, instead of copying them to HDD, it is much safer to back them up to pendrives ("PSU Paste" option in uLE).



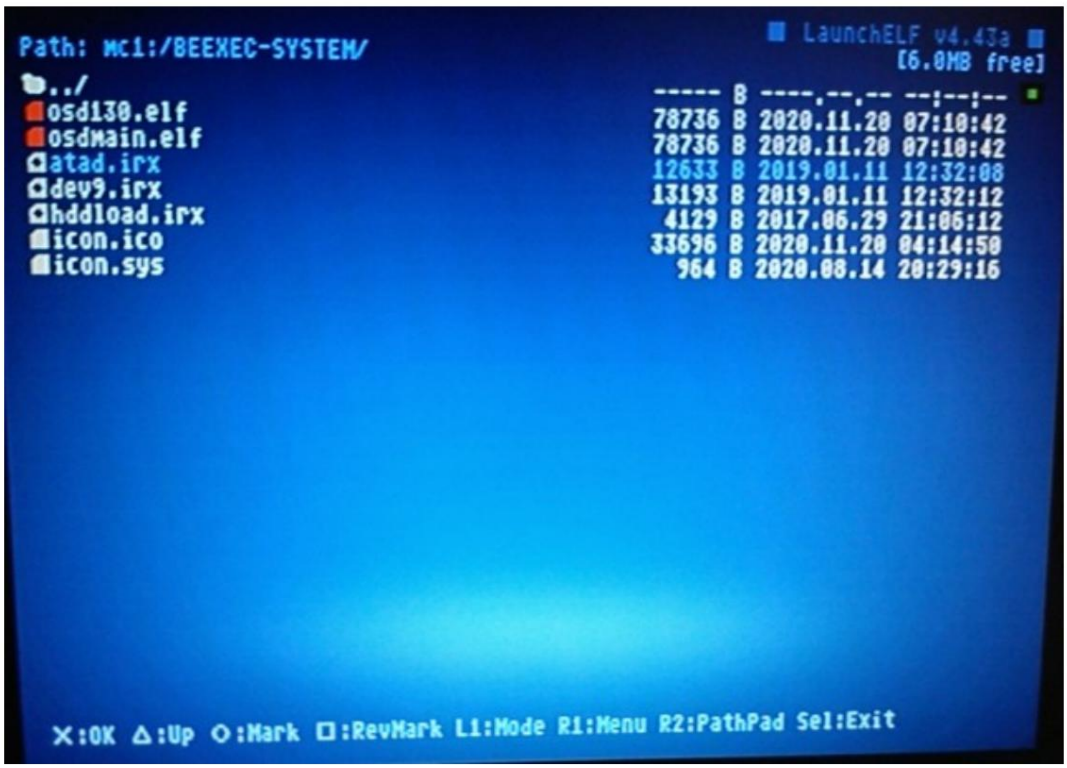
The hard drive is represented by the NA icon, while the folder in "hdd0: / __ common /" suitcase.

HDD OSD on SCPH-7K

On the SCPH-70xxx and 75xxx models, it is also possible to run the HDD OSD, but only if you have gone through the tedious process of soldering the tape to the motherboard (otherwise, of course, you will not connect the hard drive to act as the internal one). As if the problems were not enough, Sony turned off the OSD Update for HDD in this series of consoles - and humanly speaking - it is no longer possible to automatically start FHDB and / or HDD OSD only from the disk itself. Fortunately, you can get a similar effect with the FMCB (version 1.966) installed on the memory card, which has its own implementation of this function and will fire the FHDB for you from the disk, while adapting the HDD OSD.

The procedure is the same as for regular fats, with the difference that you will need to install an FMCB card with additional modules: "atad.irx", "dev9.irx" and "hddload.irx". So after uploading the FMCB, run uLE and copy the above-mentioned necessary files from, for example, a USB flash drive, which you will find in the **installer** (in the directory "\ INSTALL \ SYSTEM \", remember that their names have lowercase letters, they will not be found in uppercase), and you must add them to the "B * EXEC-SYSTEM" folder on "memorce" (the console region is hidden under the asterisk) .
If you have installed the multi-regional version, you will find these modules in "BIEXEC-SYSTEM" and from there you can copy them elsewhere.

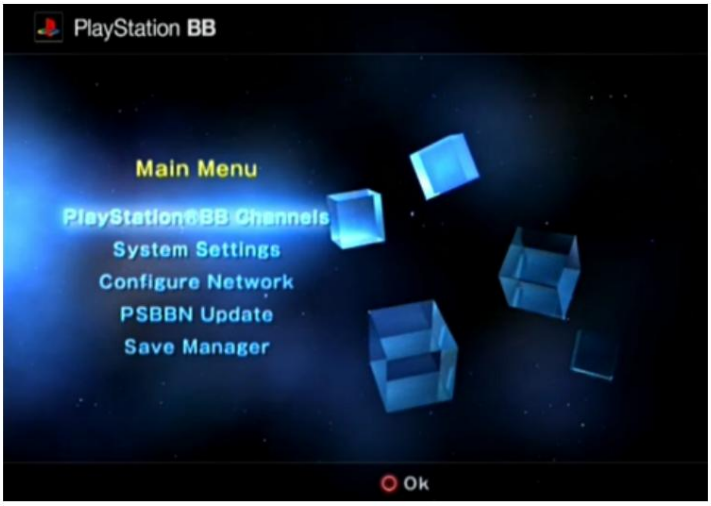
Europe	mc0: / BEEXEC-SYSTEM /
North America and part of Asia	mc0: / BAEXEC-SYSTEM /
China	mc0: / BCEXEC-SYSTEM /
Japan	mc0: / BIEXEC-SYSTEM /



In such a constellation, the PS2 will run the FMCB from the card, and the latter will run the rest from the HDD (don't be surprised when you see the FMCB logo twice, and the boot itself is slightly longer).

PlayStation BB Navigator

PlayStation BroadBand Navigator , or **PSBBN** for short, is a special Linux distro, released exclusively in Japan. With a clear conscience, the entire PSBB can be called the progenitor of the PlayStation Store and PSBBN a client on the user's side. Today, long after all game publisher / developer channels have been turned off, this exotic invention is only a curiosity and part of the console's history. It's slow to act and basically useless except for an interesting interface and the ability to watch movies and listen to music.



Unfortunately, despite the promises of Sony, the "navigator" never left the borders of the country of the blooming cherry and, as in the case of the Utility Disc, security forces the Japanese console without a modchip (installation from a copy via ESR or a disc image via HDL / OPL is not possible) and the original hard drive . Patched installers have never seen anything else patched PSBBN (and translated into English) in ready-to-upload HDD images. Unfortunately, I will not be able to guide you through the installation process, because I do not have the appropriate equipment, a transplant is also not very possible because not all data is uploaded to PFS, EXT2 and RFS partitions (one is raw or with an unknown file system). Nobody has broken it completely so far, so you are left with other sectoral HDD images that you can paint on any disk, for example with the **HDD Raw Copy Tool** or **DMDE program.**

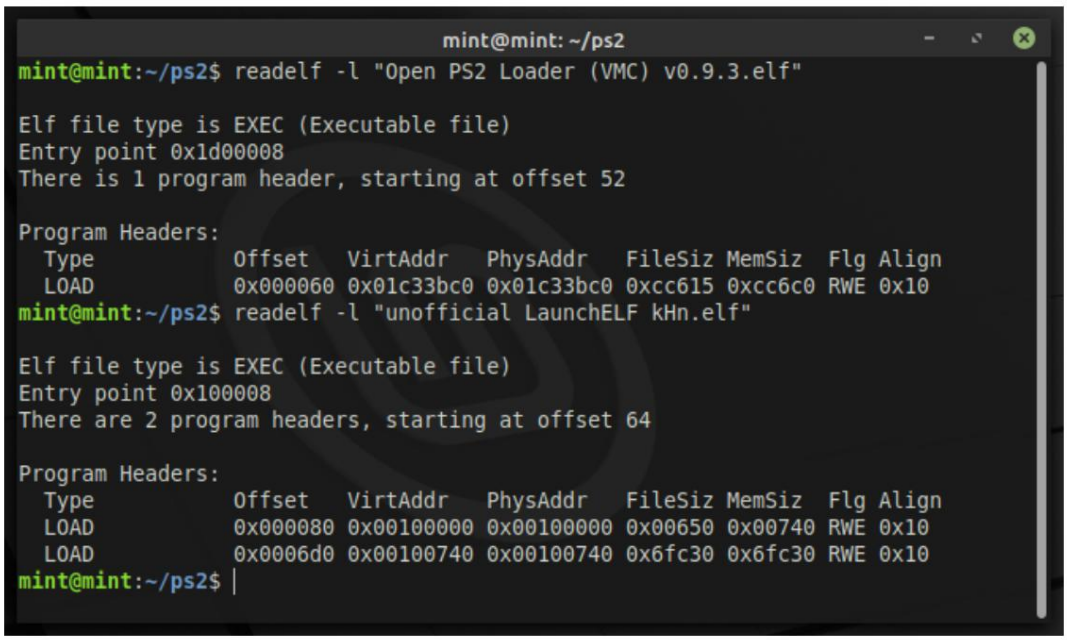
MBR Boot

The " hdd0 : / __ mbr" partition, like all the others, has a special place for the "crypto elf".
While to run something from the "PP partition" you need to have a HDD OSD loaded and cut through several menus, in the case of "MBR" not, and there the uploaded application will be automatically launched when the console starts.
It is the equivalent of OSD Update (ie FHDB).

The address **0x130** contains the sector number on which OSDSYS is looking for the file. Sectors are represented by the LBA system, where the unit in this case is always 512 bytes. That it would not be too easy, it is in the reverse order of the bits. ;) So on the example of FHDB, you will find **0020h there.** So you have to swap these values to get 2000h. Now, by changing this number from hexadecimal to a friendlier system for normal people, that is, to decimal, it turns out to be 8192d. Since this is a sector number and the magnitude of each is 512B, you have to do a multiplication of these numbers and you will find 4194304d. Now it is enough to use the conversion of the number system again, this time from decimal to hexadecimal, to find out the address on which the KELF is located, which in this case is **400000h.** Any address can be set, of course, as long as it does not conflict with any other important information (e.g. part of an index). This is specifically recommended because Sony also uses it.

The same is with the file size you will find at **0x134.** You also need to make the above-mentioned transformations. In the example of the FHDB, the size is 9800h, which will be 77824B after the mathematical adventures. And that's exactly what the loader FHDB weighs. If this file size is not correct, OSDSYS will not start KELF, the console will take a long time to turn on and hang in the menu. The maximum size that can be used is **883200B.**

The program cannot be any program either, only one that will be loaded on the physical address in memory **0x00100000.** This can be checked for example on Linux on a computer using "readelf -l <application name>". Later, the header should be truncated and converted to KELF (a description of how to do this can be found in the chapter entitled "Creating PP Partition").

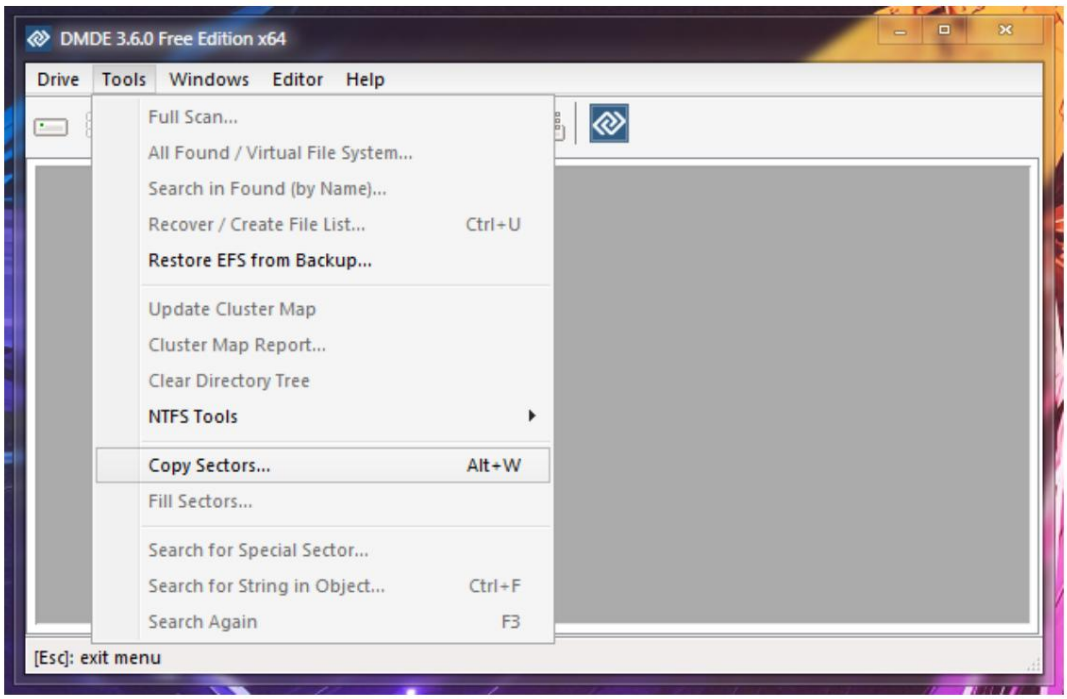


Incompatible OPL and compatible uLE kHn.

I honestly admit that I have not found any application that meets all the above conditions (i.e. the size and address limit in operating memory), so if you need such functionality and in the form of a file manager, I recommend that you simply upload **the unofficial LaunchELF kHn,** which has already been adapted by the author .

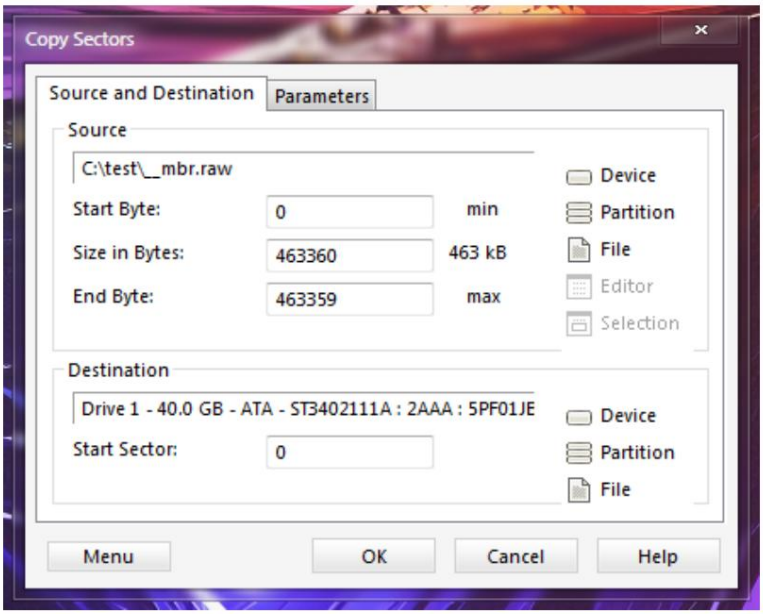
And here there are two possible ways: either you are trying to inject a file ("MBR.KELF") to a specific address (i.e. by exercising with the calculator, but keeping all partitions), or you overwrite "MBR" with a specially crafted piece of this partition ("__mbr.raw" "), but losing all others at the same time.
The following description is based on the second variant, because it is the simplest and does not require familiarity with the hex editor.

1. Run the **DMDE** program under administrator rights and from the "Tools" menu, select the "Copy Sectors ..." option.

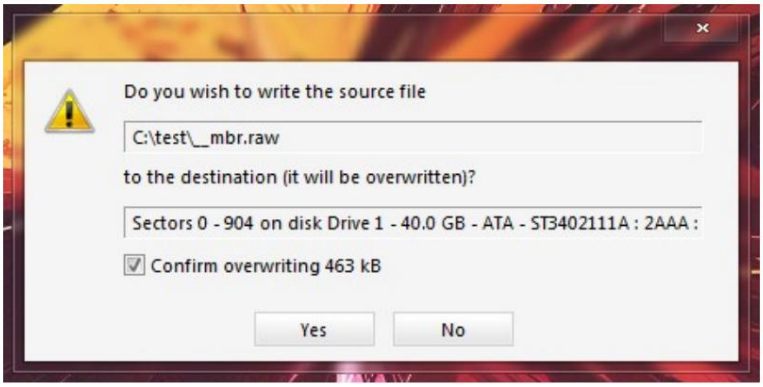


2. In the "Source" category, click the "File" button and select the "__mbr.raw" file. Start Byte should be zero. In turn, in the "Destination" category, click the "Device" button and select the PS2 hard drive connected to the computer from "Physical Devices". "Start Sector" should also be zero.

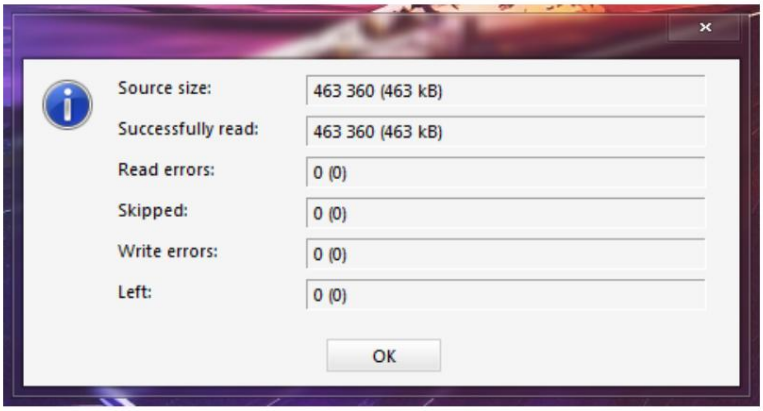
After making sure that the selected HDD is correct,;) click the "OK" button.



You will be asked for confirmation again, so check the appropriate box and of course click "Yes".



3. After the operation is completed, you should not see any errors in the summary.



Copying games for PSX to your hard drive

POPS is the official PlayStation emulator that was downloaded together with the game entitled "Bishi Bashi Stepchamp 3", from the Konami Channel. Other games were never released in this way, and the "emu" itself, as well as the rest of the content for PSBBN, was encrypted and assigned to that particular HDD and that particular console on which the game was downloaded. Due to this absurdly murderous DRM, the lack of keys and the popularity of the "navigator" outside of Japan close to absolute zero, it has not been broken for a long time. The first public decrypted SLBB-0001 (i.e. just POPS) were tested and modified in various ways. The "clean version" on which I base the guide consists of two files (basically three, but "POPS_IOX.PAK" in the case of HDD is not needed), which you have to find on your own in the abyss of the Internet (the other versions will not be compatible):

POPS.ELF emulator MD5: 355A892A8CE4E4A105469D4EF6F39A42

IOPRP252.IMG MD5 IOP Memory Image: 1DB9C6020A2CD445A7BB176A1A3DD418

Additionally, you need the **POPStarter program.** It is a loader that patches the virgin version on the fly, enriching it with various additional functionalities, but above all significantly increasing compatibility with games.

Preparing the Environment

There are three ways to run PSX games from disc images using the official Sony emulator. They differ in the location and naming of the loader. In all cases, the emulator and the IOP image should be dumped in "hdd0: / __ common / POPS /".

Variant A

If you want to keep all PSX games on one partition, you have to put emulator and POPStarter files under the name "POPSTARTER.ELF" into "hdd0: / __ common / POPS / ". Then you put the games into "hdd0: / __. POPS /" (they will be loaded only from here) and you run them simply by selecting the * .vcd disc image in the **uLE kHn** file manager (at the moment, this method will **not** work with any version of uLE other than kHn).

Scheme:

"hdd0: / __ common / POPS / POPSTARTER.ELF"
"hdd0: / __. POPS / *. VCD"

Variant B

Alternatively, you can also name POPStarter exactly like the disc image and put it in "hdd0: / __. POPS /", but then you unnecessarily duplicate the loader for each game separately, although then you are no longer tied to uLE kHn but any hive , because you start the games by selecting a loader named for that game.

Scheme:

"hdd0: / __. POPS / <game title> .VCD"
"hdd0: / __. POPS / <game title> .ELF"

Variant C

If you prefer to have a bootable game, but each partition is separate, then you will waste a lot of disk space (not many games will play multiple 128MiB) and some time to prepare the PP partition (more on this in a separate chapter). However, you get an attractive form of starting the game, directly from the HDD OSD menu.
The disc image must be named "IMAGE0.VCD", and the POPStarter must have the name that you enter in the CNF file (but you can also read about it in the section on creating PP partitions) and it must be a signed version, i.e. KELF (because the HDD OSD only such starts).

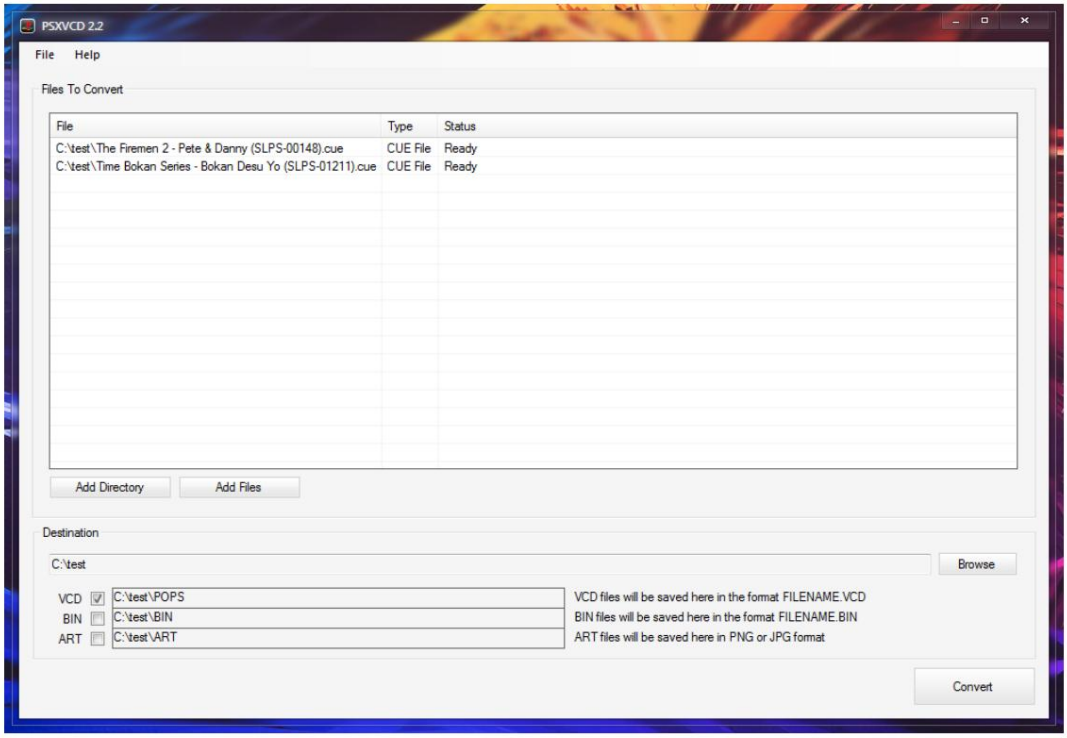
Scheme:

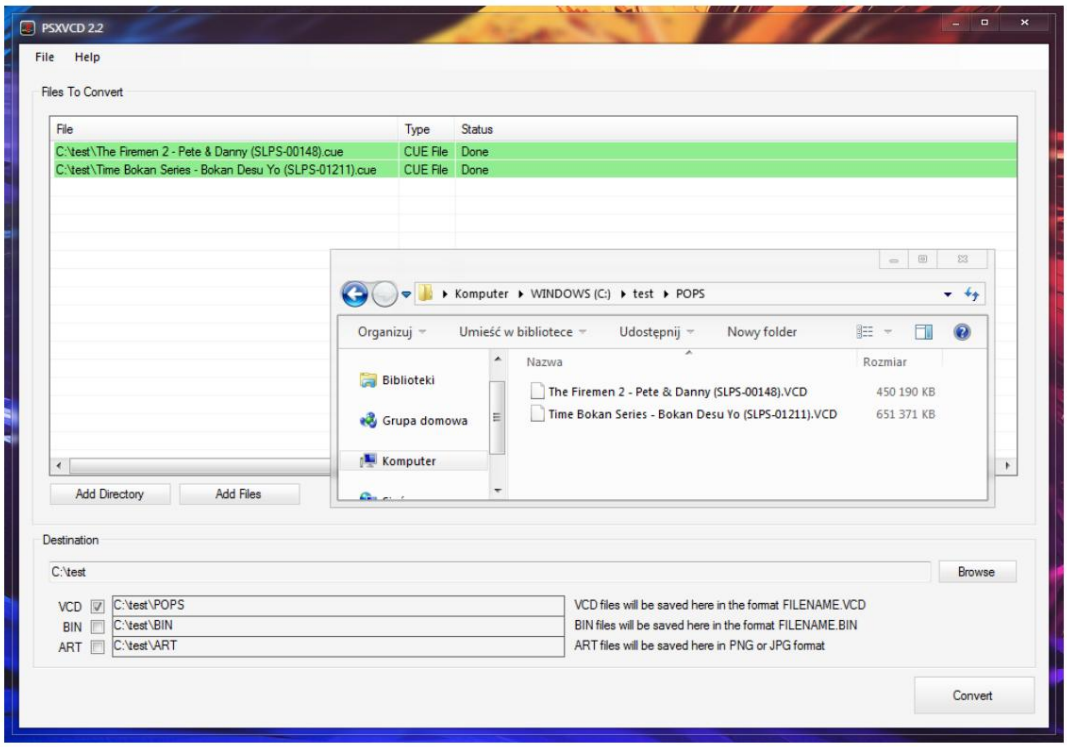
"hdd0: / PP. <Game title> /IMAGE0.VCD" (in case of multi-disc games, for consecutive discs instead of zero: 1, 2 and 3) "hdd0: / PP. <Game title> / <name defined in "system.cnf">. KELF "

Converting

games Disc images must have all CD-Audio tracks merged (if the game has one) and contain TOC. Fortunately, there is a tool that automates this process, which is **PSXVCD.**

Run the program, add CD images ("Add Files") or a directory with images ("Add Directory"), select a destination for the * .vcd images ("Browse") and click the "Convert" button.





Be sure to leave the VCD extension in capital letters. Also remember not to exceed 33 characters in the file name, because more will not be displayed in the HDD OSD in the game folder preview.

Game Folder

For each running game (at least once), POPStarter in its latest version (i.e. R13 Beta from 2019-06-05) creates a folder where resources are stored per title ("hdd0: / __ common / POPS / <image name plates> / "), but you might as well anticipate him and personalize the angle a bit.

Virtual memory cards So
images of real "meme" should already be formatted and have exactly **128KiB** (so they should be without junk headers that are added by eg DexDrive (* .gme) or POPS from PSP / PSV / PSTV (* .vmp)).
Corresponding to the slot in a real console, the files "SLOT0.VMC" and "SLOT1.VMC" correspond to them. So you can successfully throw in virtual PSX cards from e.g. PS3 (* .VM1) or ePSXe emulator (* .mcr), simply changing their names to the required ones (i.e. from "epsxe000.mcr" to "SLOT0.VMC" ").

The problem occurs with multi-disc games, because a separate folder will be created for each disc image, incl. with separate "memorki". And while with titles that allow you to save between media changes (eg "Parasite Eve"), this problem is limited to saving the "last save", shutting down the game and copying VMC to the next folder, there are those that do not provide such a possibility (eg, "Chrono Cross", "Metal Gear Solid"). There are two solutions: either with special **ripkits** you will remove the code responsible for changing the medium and merge all disc images into one, creating a hybrid that is incompatible with the art and standard, but a working hybrid - or you will make a text file "VMCDIR.TXT", in which you will enter the name of the folder for common cards (e.g. "Clipboard" and of course you must create such a folder as well; do not enter the paths). If this is not yet clear then take a look at an example below:

"hdd0: / __ common / POPS / Metal Gear Solid (CD 1) /VMCDIR.TXT" "hdd0: / __
common / POPS / Metal Gear Solid (CD 2) /VMCDIR.TXT" "hdd0: / __ common /
POPS / Clipboard / SLOT0.VMC "" hdd0: / __ common / POPS / Clipboard /
SLOT1.VMC "

Multi-disc games And
while we are talking about games released on more than one disc, a similar problem arises with the selection of the next disc. So create a file "DISCS.TXT" in each of the folders corresponding to each separate disc. For example for MGS:

"hdd0: / __ common / POPS / Metal Gear Solid (CD 1) /DISCS.TXT" "hdd0: / __
common / POPS / Metal Gear Solid (CD 2) /DISCS.TXT"

And enter the names of the disc images there (maximum four).

```
Metal Gear Solid (CD1). VCD Metal
Gear Solid (CD2). VCD
```

When the game asks you to change the medium, you will use a combination of buttons:

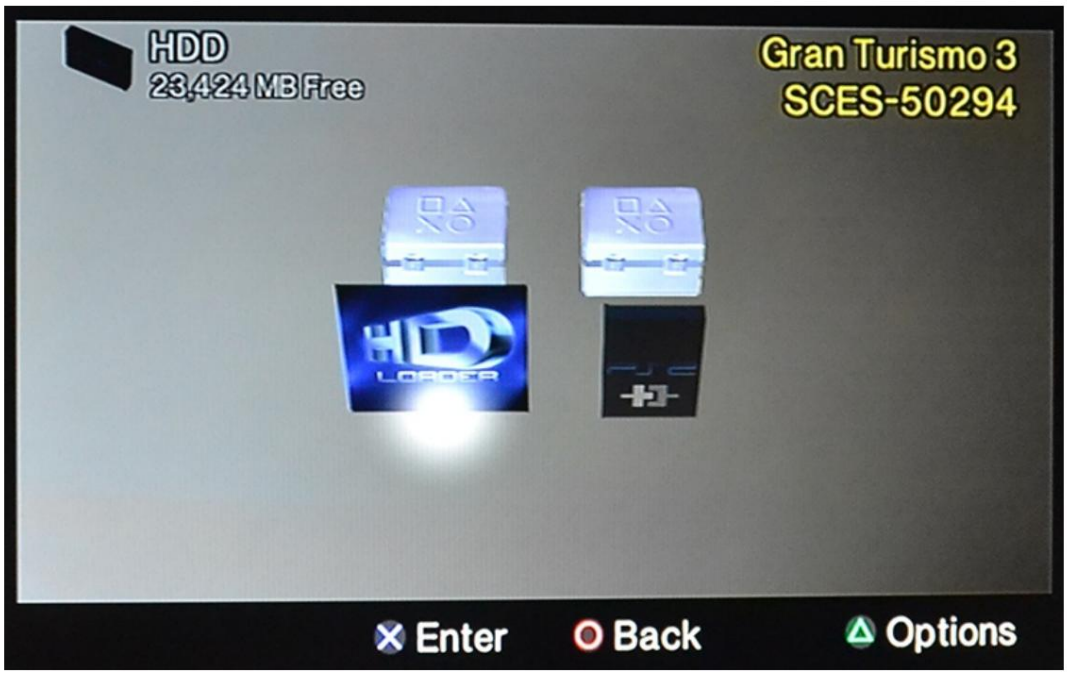
Opening the flap	Select + L2 + R2 + Triangle
Inserting plate no. 1 Insert	Select + L2 + R2 + Up
plate no. 2	Select + L2 + R2 + Right
Inserting disc no. 3 Insert	Select + L2 + R2 + Down
plate no. 4 Closing the	Select + L2 + R2 + Left
hatch	Select + L2 + R2 + Square

Copying PS2 games to your hard drive

Hard drive games for PlayStation 2 outside of PSBB channels - that is, all those loaded by various stage applications - are disc images. If necessary, sliced like a cucumber (in terms of partition fragmentation) and rounded up in size to fit within the limit imposed by the unfortunate APA.

There are different methods of uploading games and each has different nuances. You can copy games using **WinHIIP** to a disk physically connected to the computer (it cannot cope with large HDDs, but for example, it is the only one that supports APAEXT). Similarly, the same can be done with **HDL Dump** or **HDL Dumb** (also via the network). You can also use the best program in my opinion - **HDL Game Installer.** It is an application from which games can be installed directly from the console's optical drive, but it is also a server for the client on the Windows side. If your PS2 disc reader is barely misfired or dirty, or if the game you want to copy has been released on DVD9, a two-layer disc, use the network method. If you want to install a game released on DVD9, but necessarily from the disc inserted in the console, then use **HDLoader.**

HDLGame Installer and HDL Dump / Dumb immediately load the icon, description and mini version of the Open PS2 Loader into the attributes field, so theoretically all games can be launched from the PS2 menu without running additional programs (of course only if you have the HDD OSD installed).



HDL Game Installer - disc variant **Note:** no support for games on double-layer DVDs (commonly known as DVD9 or DVD DL).

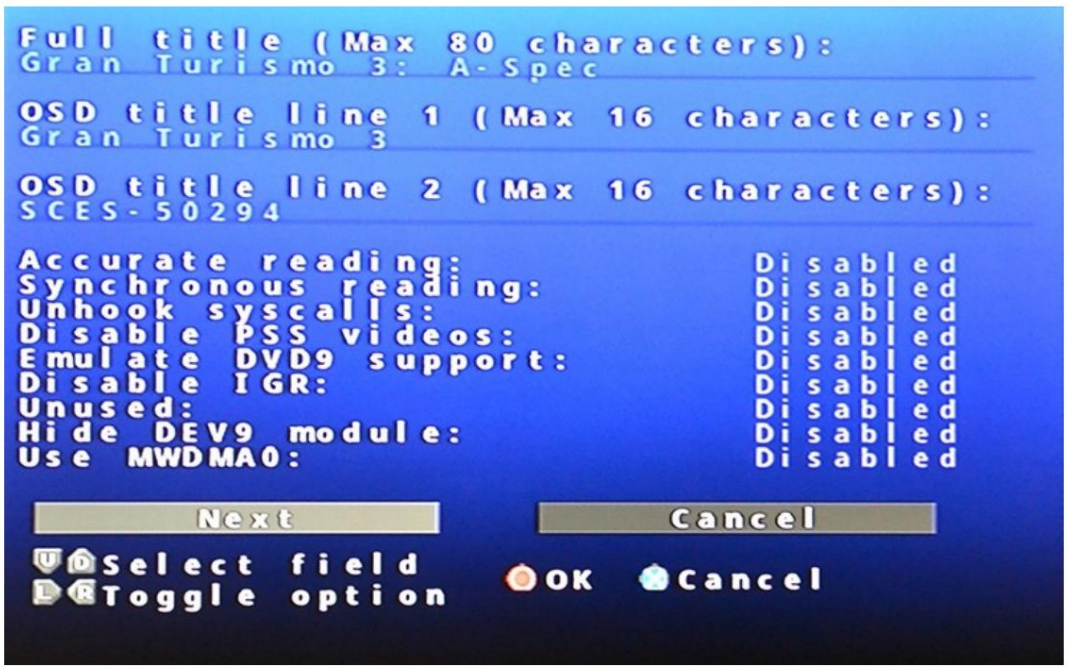
1. Run **HDL Game Installer** and wait a moment for the drive to recognize the disc. When the whistling stops, press the Start button.

The order here is important only for people with a weak laser or dirty lens and / or mirrors, because for some reason "browser" when he cannot wait;) until the Mechacon allows him to access the medium, displays "red clouds" and you have to eject again / insert the tray ("monolith models") or open / close the flap (slim models).

2. Give the game a title displayed in the HDLoader and Open PS2 Loader menus (you can use up to 80 characters). Do the same with the title and subtitle displayed in the browser (16 characters each).

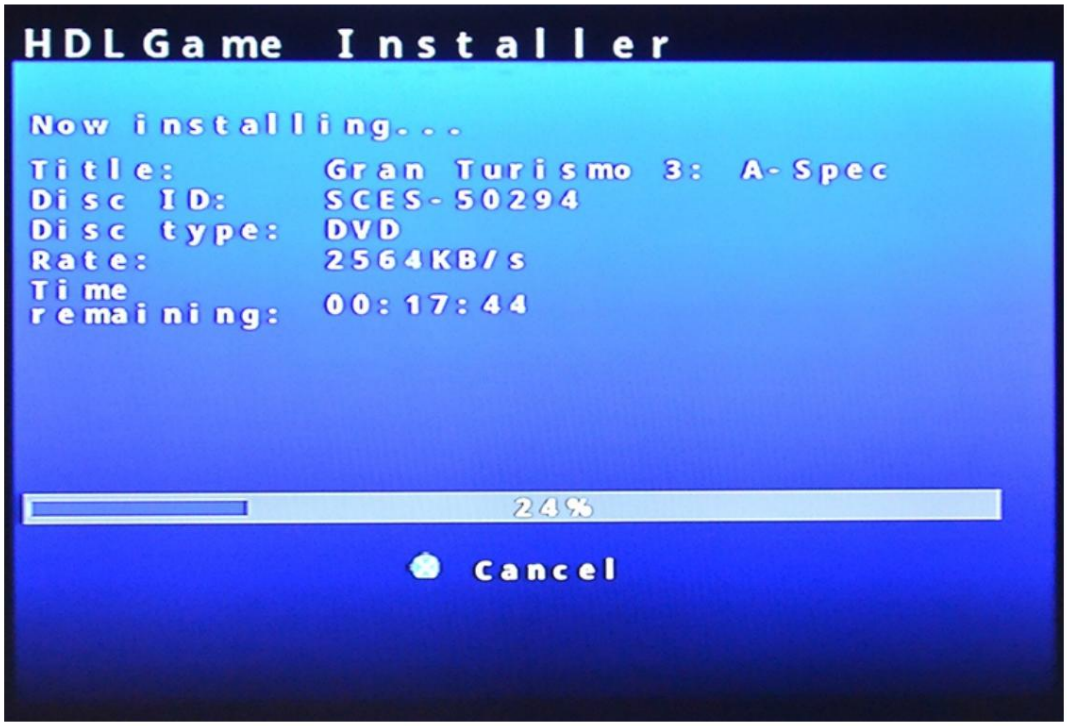
Additionally, you can mark special work modes for the loader, but if you don't know what and why, leave them with the default values. You can change these options at any time, also after installing the game.

Select "Next" and confirm your choice.



3. The program will ask for confirmation, so select "OK". The next question will be about the icon used. If you have a save from this game on a memory card (because only it will be loaded from there), I recommend that you select "Game save". It will certainly be prettier than the default one, i.e. from the HDLoader, which is ancient today. Confirm your choice of course "OKAY".

4. The process of copying the game to HDD will start, which is represented by the percentage of progress bar, and when finished it will display the message "Game installed successfully!".

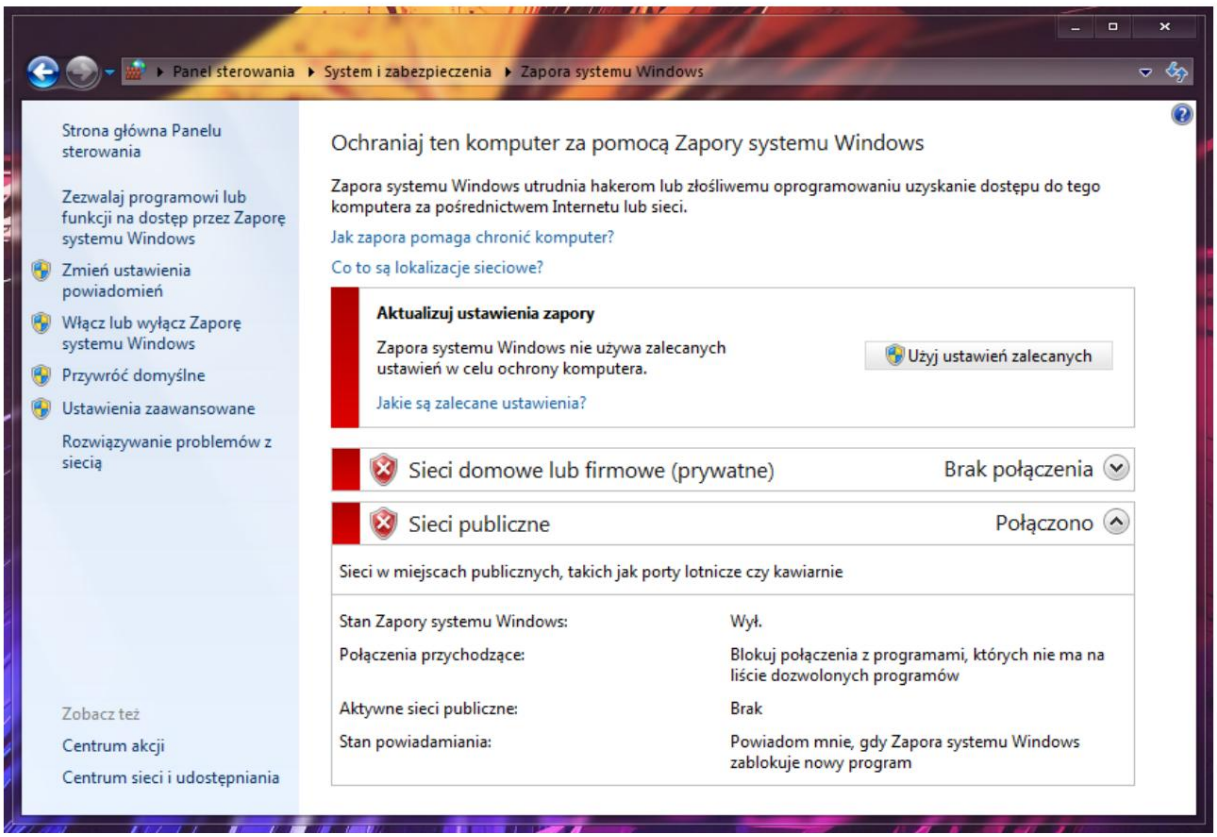


5. After returning to the main menu of the application, you should see the game on the list. This is where you can change "Game options", which you will learn more about in the section on the meaning of options.

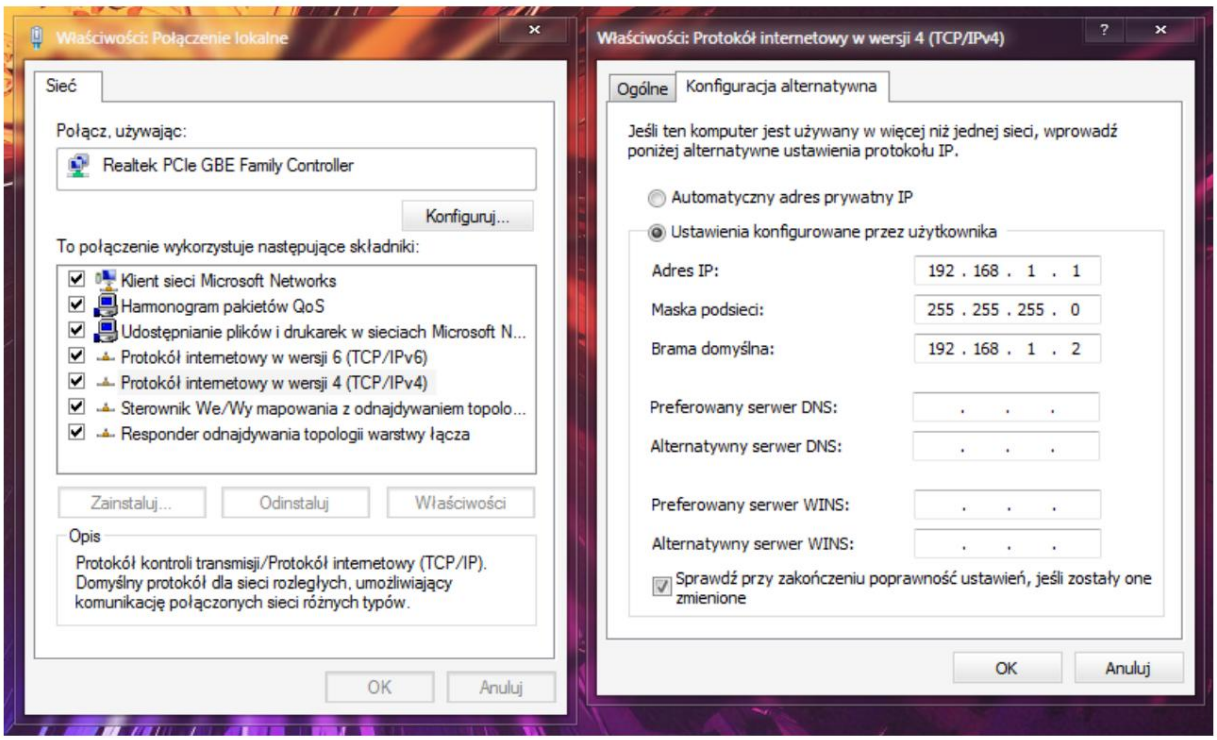
HDL Game Installer - network variant

1. Go to the security settings in Windows and turn off the firewall (or unblock TCP ports 45061 and 45062 in it, or let the program that will talk to the console pass).

I do not use this solution on a daily basis, I only have one Ethernet socket in my computer, and the console is right next to the PC. So, out of laziness, instead of breaking through the router, firewalls and other stick miracles, I followed the path of least resistance. Which, of course, I do not recommend for security reasons.

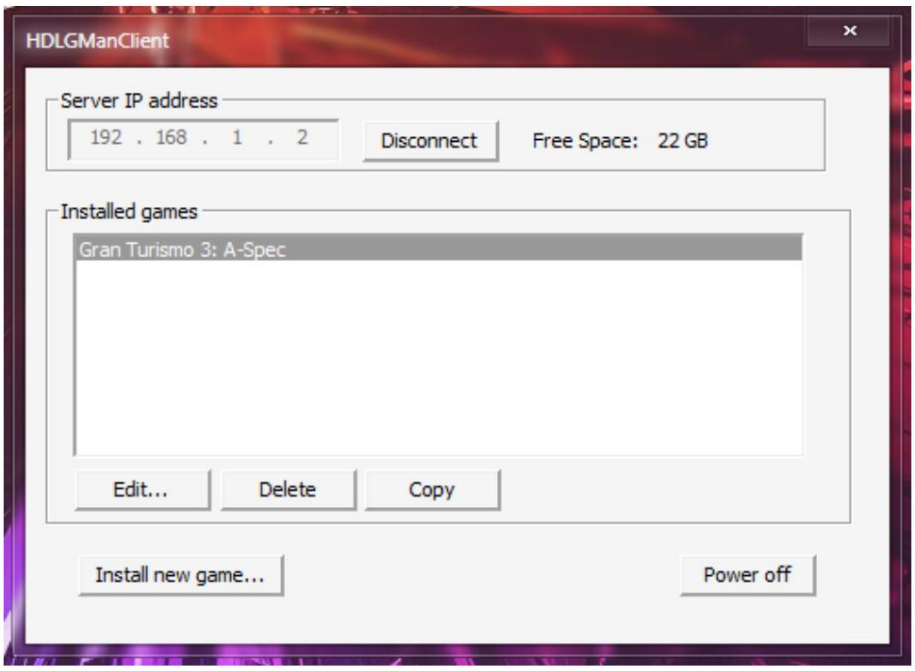


2. Go to network card settings and set hard your local network.



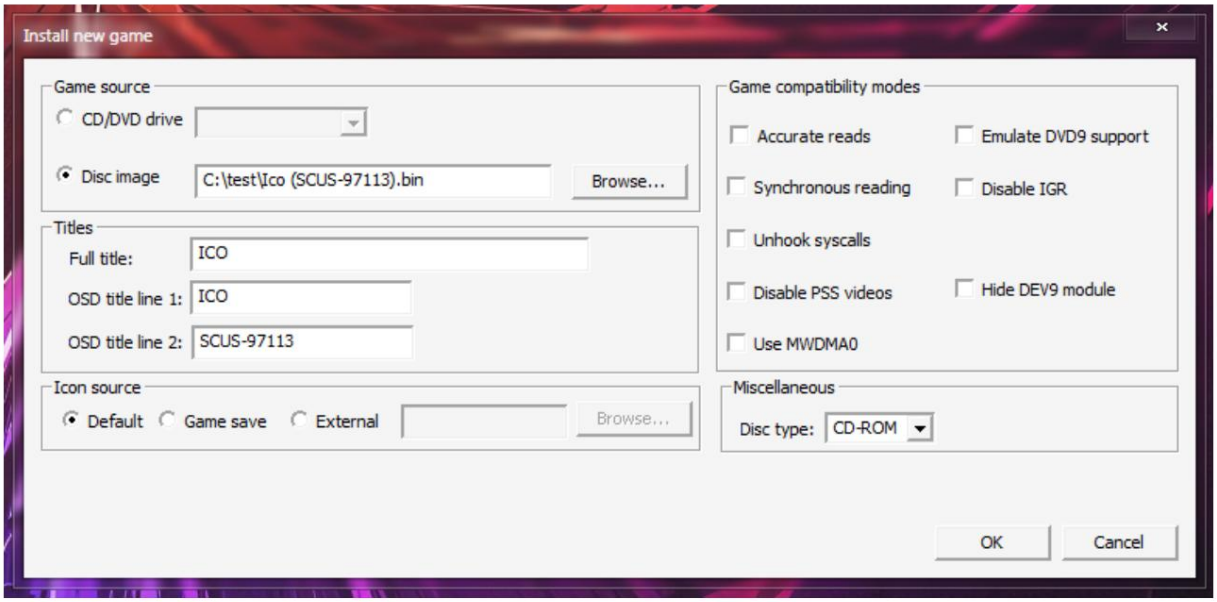
3. Run the **HDL Game Installer** program and wait for a while for it to try to connect to the computer. Which is likely to fail. Enter the settings by pressing Select, turn off DHCP ("Disabled") and set the address, mask and gateway according to the settings in Windows.

4. Go back to PC and **run HDLoader Game Manager Client for HDL Game Installer,** enter the correct address and click the "Connect" button. A list of games installed on the disc should appear.

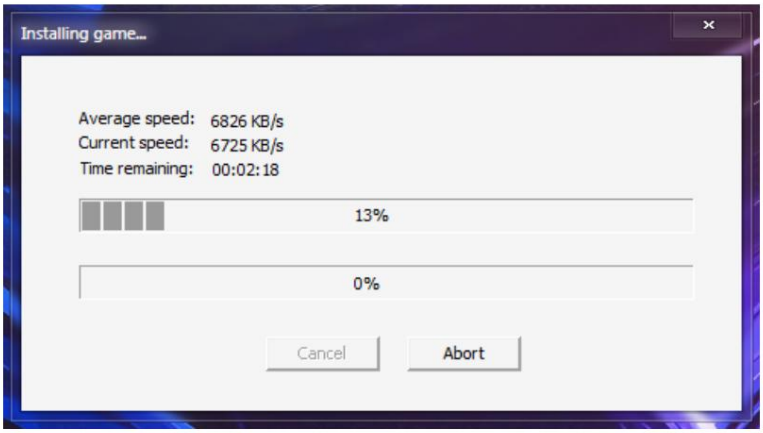


5. Click on the button "Install new game ..." and then on "Add Game". In the next window, choose the installation method, whether from an optical drive ("CD / DVD drive") or from a disc image ("Disc image"). In the case of a disc image, * .iso for DVD and * .bin for CD are supported (do not point to * .cue). As with the console CD installation, you need to specify the titles displayed in homebrew (80-character limit) and the console menu (16-character limit). You can also select an icon, e.g. from a computer, and set the game launch options. Also select the type of media ("CD-ROM" or "DVD-ROM").

Finally, click on the "OK" button



6. You will go back to the "Job List" menu and if you do not want to add more games, press the "Start" button (in the program window, of course, not on the joypad). You will be asked for confirmation and the process of installing the game on the console will begin.

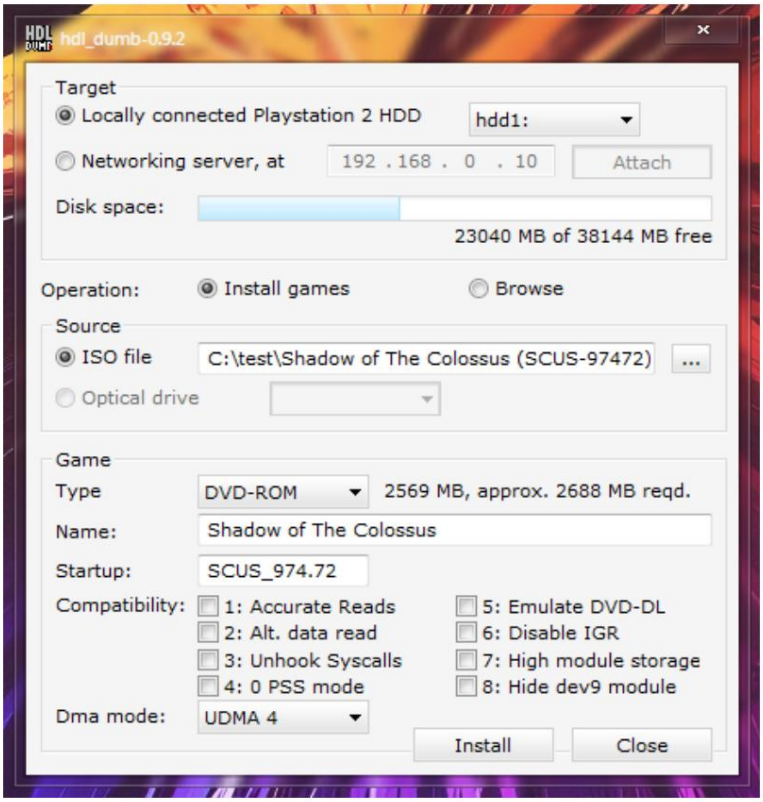


After all, the message "Games installed successfully" will be displayed. And in the console, after exiting HDLGI, the game is ready to run.

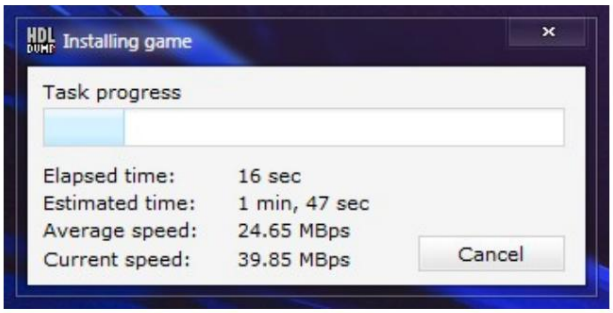
1. Connect the hard drive from the console to the computer.

2. Launch **HDL Dumb.** Check "Locally connected PlayStation 2 HDD" which should be detected immediately. Select the location of the disc image (supports the following formats: * .iso, * .gi, * .iml / *, * .img, * .nrg, * .bin / *. Cue). Specify the media type, title, additional options, and data transfer mode.

Finally, click on the "Install" button.



3. The game installation process will start. When finished, the window will disappear and the game will be waiting for you on the HDD.



The meaning of the

HDL Game Installer option is loaded into the attribute field by the Open PS2 Loader program in the version without a graphical interface. It is an older version than the current GUI version, therefore some options may differ. However, you can always run games via external OPL, not the one from the header and configure them in the same way.

- **Accurate reading** (mode 1)
  Emulates reading speed from a disc. Some games have problems running faster than the drive speed.

- **Synchronous reading** (mode 2)
  Changes the read mode from asynchronous to synchronous. Some games need to be read immediately after a specific sector or file access is requested, otherwise they crash.

- **Unhook syscalls** (mode 3)
  With this option enabled, OPL will not stay in RAM if the game resets (overwrites) the IOP memory by loading the IOPRP image. Then neither IGR (InGameReset), nor GSM (Graphics Synthesizer Mode Selector), nor PS2RD (PS2 Remote Debuger (formerly Artemis)) can be used.

- **Disable PSS videos** (mode 4)
  Disables * .pss format movies. A useful option for people who do not want to watch "cutscenes" in the game or simply have very slow flash drives (some games can hang because of this).

- **Emulate DVD9 support** (mode 5)
  According to the descriptions, this option emulates a two-layer disc for games ripped into a single-layer one, but I must admit that I do not really understand what is going on here, since the so-called ripkits take care of changing media check to DVD5 and possibly patching OTP to PTP.

- **Disable IGR** (mode 6)
  Disables the ability to reset the game using a combination of joypad buttons. Some games are incompatible with this and must be opted out of IGR.

- **High module storage** (mode 7, no longer used)
  Changes the storage address of modules in memory for games that conflict with them.

- **Hide DEV9 module** (mode 8, no longer used)
  Hides the network module as "plate".

- **Use MWDMA0** (shifted in the new OPL to a separate category of data transfer mode from HDD)
  HDD reading mode set to MDMA-0. Approximately reading speed from the disc. How does this relate to mode 1, I do not know, I suspect that instead of this option, you should use UDMA-4, and in case of potential problems related to it, turn on mode 1 (ie "Accurate reading").

Creating a PP partition

The HDD OSD will not display any other partitions than PP, PC and the contents of "__common" in the "browser" if not it will have the name consistent with the nomenclature, the correct icon will be loaded and a metadata file describing it in the field attributes. Such a partition will either be ignored (system) or displayed as Corrupted Data
(any other). The only advantage of "PP partition" is the ability to automatically start programs from within "Browser".

Preparation of the icon

The icon format is the same as for PS2 memory cards (so you can take any one of some program or **tweak your own,** name it "list.ico"), but the file describing it for some mysterious reason is not binary versus text.

Create a simple ANSII text file with Windows line breaks, complete according to the following formula and save as "icon.sys".

PS2X	A header that tells the PS2 the type of content (D: folder, X: executable).
title0 =	For example, the title of the game. The maximum number of characters is 16.
title1 =	For example, the game ID. The character limit is the same as above.
bgcola = 0	Background transparency value.
bgcol0 = 0.0.0	The background color in the upper left corner. RGB.
bgcol1 = 0.0.0	The background color in the upper right corner. RGB.
bgcol2 = 0.0.0	The background color in the lower left corner. RGB.
bgcol3 = 0.0.0	The background color in the lower right corner. RGB.
lightdir0 = 1.0, -1.0,1.0	The direction of the incident first light. X, Y, Z.
lightdir1 = -1.0,1.0, -1.0	Direction of incident second light. X, Y, Z.
lightdir2 = 0.0,0.0,0.0	Direction of incident third light. X, Y, Z.
lightcolamb = 64.64.64	Color of general lighting. RGB.
lightcol0 = 64.64.64	The color of the first illumination source. RGB.
lightcol1 = 16.16.16	The color of the second illumination source. RGB.
lightcol2 = 0.0.0	The color of the third illumination source. RGB.
uninstallmes0 =	Displayed message while deleting a partition.
uninstallmes1 =	As above, only the second.
uninstallmes2 =	As above, only the third.

Preparation of the CNF

The CNF file format is similar to that of optical discs. Create a simple ANSII text file from Windows line breaking, complete according to the following formula and save it as "system.cnf":

BOOT2 = pfs: /BOOT.KELF	Absolute path to the executable on the target partition.
VER = 1.00	Program version. Irrelevant.
VMODE = NTSC	Region. Also irrelevant.
HDDUNITPOWER = NICHDD	Responsible for not disabling the Network Adapter and HDD.

If you enter "PATINFO" instead of the path to * .kelf, then the file will be uploaded to the attribute field, not to PFS space. Only really useful if you don't intend to change it in the future
file (because it is easier to replace a file with a file manager, for example, than to fight with the command line). The same is true "IOPRP =" (if you need it, it should go under "BOOT2 =").

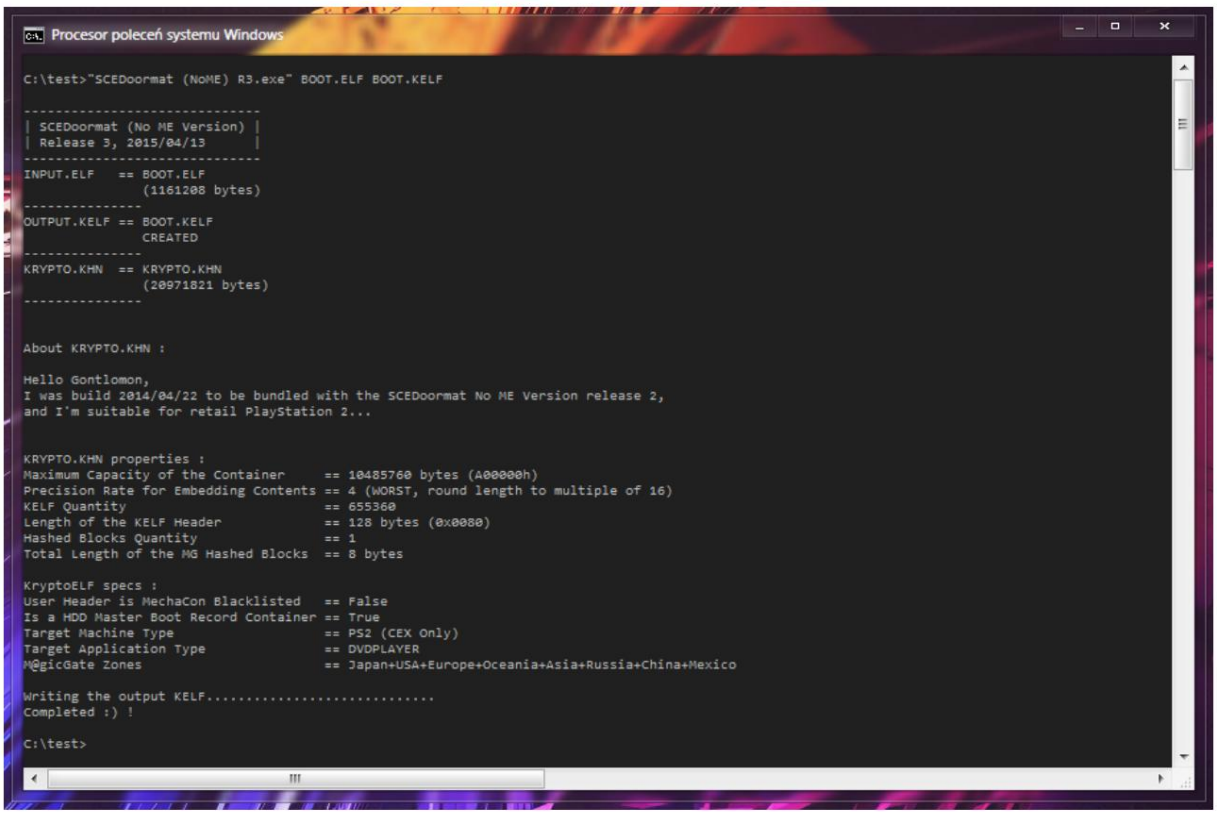
Program preparation

The executable files for PS2 are in *** .elf format,** the problem is that the HDD OSD will only run *** .kelf,** i.e. also "elves", but signed and encrypted (commonly known as **Krypto ELF,** they can also have the *** .xlf extension) .** To do this correctly, you would have to use the **kelftool application.** Unfortunately, in the current version the program allows you to create "kelfów" only for OSD Update for DESR-xxxx models. So it remains to use **SCEDoormat,** which, although not as elegant as the aforementioned kelftool, at least allows you to sign the file so that the HDD OSD is not fussed about. ;)

1. Download the application SCEDoormat (NoME) R3, along with the necessary file "KRYPTO.KHN".

2. To the program directory, put the "elf" you want to run after selecting the partition on PS2 in the "browser".

The syntax is simple: "<program name> <input file> <output file>". In my case it is just a hive named "BOOT.ELF", and the spit file "BOOT.KELF" (in line with what I wrote earlier in "system.cnf").



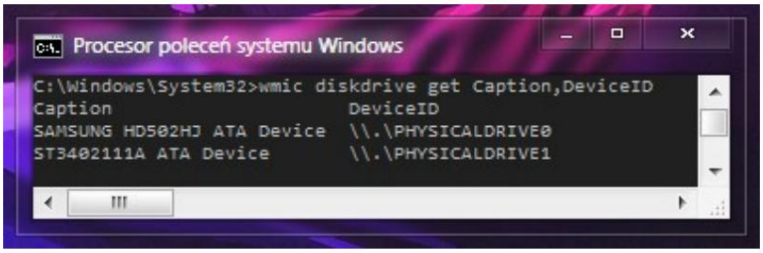
Attribute field modification

Now that you have all the files ready, put them in the same place as **PFS Shell** and **HDL Dump programs.**

Connect the drive to the computer under the PATA controller (i.e. traditionally with 80-core ATA and Molex tape). As always in such cases, I recommend avoiding the USB pockets because their quality and manufacturer support vary (sometimes the ATA standard is not fully implemented, and even that the LBA48 support is in some awkward way).

1. At the very beginning, run CMD or PowerShell as administrator, as one of the programs you are about to use requires such permissions.

2. Enter "wmic diskdrive get Caption, DeviceID" to get the UNC path to the disk. In the example below it is "\\ PHYSICALDRIVE1" because my HDD model used by PS2 is ST3402111A this time.
Remember this number because you will need it again.



3. Now **start PFS Shell** and tap "device \\. \ PhysicalDrive1". The size of the letters does not matter, I chose this notation because it is easier to read in my opinion).

Unfortunately, on Windows this is the best method to access the physical HDD, i.e. using UNC.
On Linux it is enough to list the mass devices with "lsblk" and refer to "/ dev / **sdx"** or "/ dev / loopx" by default.
The progenitor used the nomenclature "disk1:", "disk2:" etc. but on this version it wo **n't** work.

If HDD is detected, blinking ">" will change to "#".

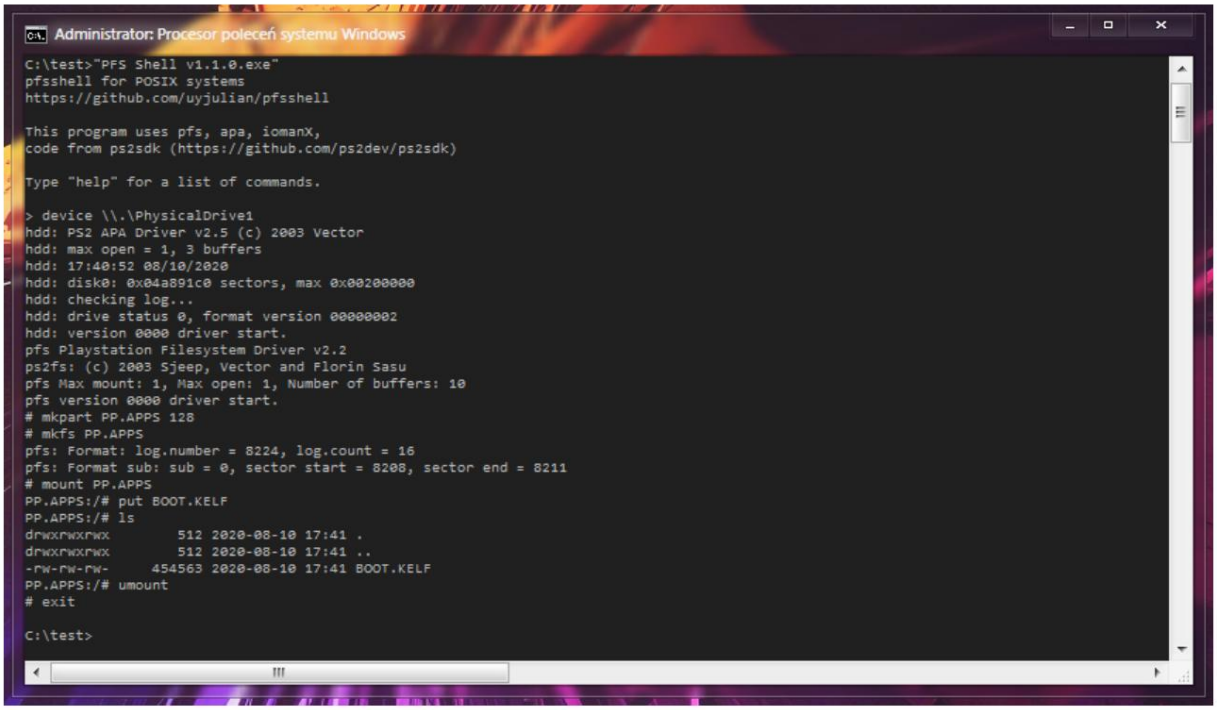
4. To create a new partition, type "mkpart PP. <label> <size>", so in my example it will be "mkpart PP.APPS 128".

5. You also need to format this partition, which you do with the command "mkfs <partition>". According to the example above, it will be "mkfs PP.APPS".

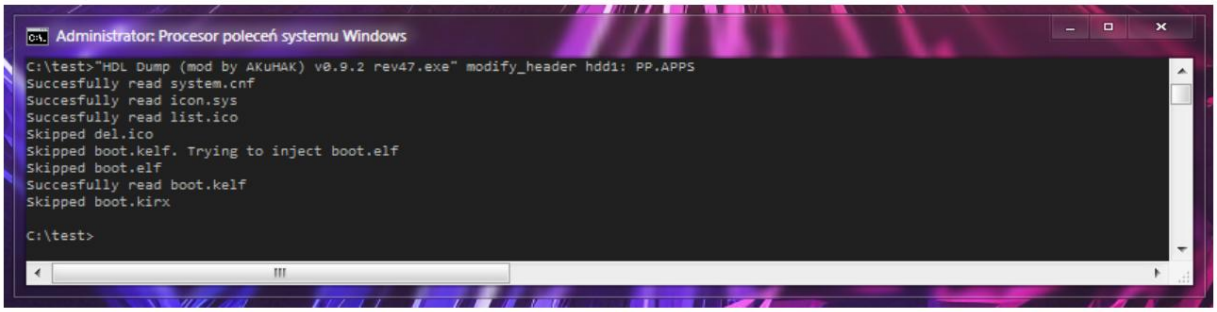
6. Mount the filesystem from the recently created partition by typing "mount <partition>", ie "mount PP.APPS".

7. Copy the "BOOT.KELF" file to the console's hard disk by typing "put <file>", in this case "put **BOOT.KELF".**

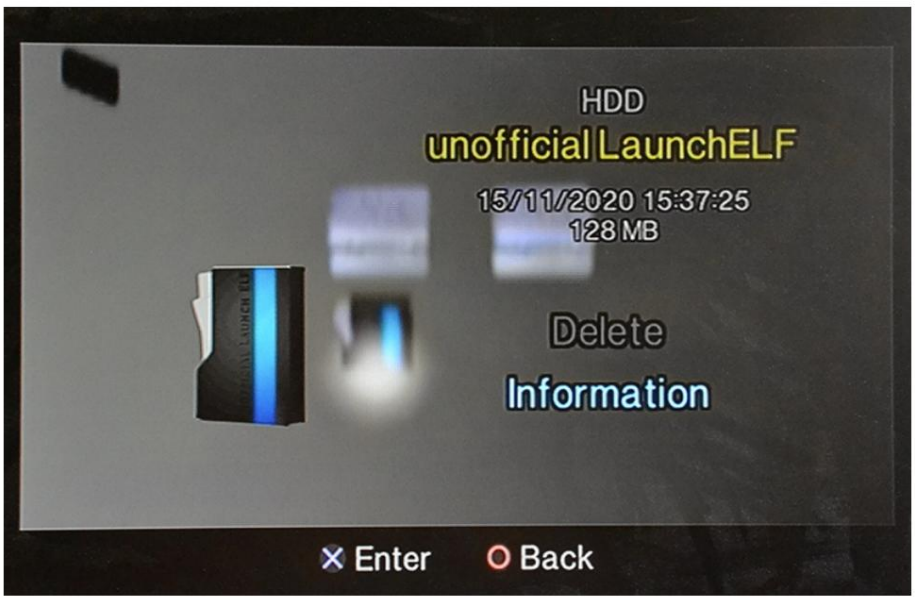
8. Finally, type "umount" to unmount the partition, then "exit" to exit the shell.



9. Now run **HDL Dump** with the parameters "modify_header hdd <disk number>: <partition>", so in my example it will be "modify_header hdd1: PP.APPS"



Theoretically, HDL Dump will throw BOOT.KELF on the indicated partition by itself, as it has entered into BOOT2 in the file "system.cnf", but it did not do it on the version I tested. So I decided that since I run PFS Shell anyway and create a partition with it, I will also throw in a bootable program. Alternatively, you might as well create a partition with the uLE or just use some existing "PP". The bootable file can also be transferred via the network or pendrive in uLE. Whatever is most comfortable for you.

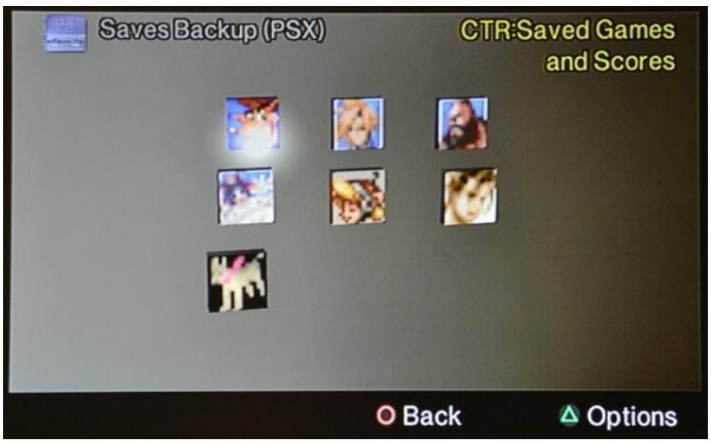


Custom folder icons on the "Common partition"

The default icon for folders created on "__common" is a suitcase. The default for files and subfolders is white sheet. But if the parent folder attribute changes to **C4A7h ,** the PS2 will start, inter alia, parse inside "icon.sys" and display the icon (same as on "memorce"). Sony has reserved this feature for "hdd0: / __ common / Your Saves /", in which some games allow you to write, just like on a memory stick. However, there are no contraindications to use this trick for completely different directories. And don't get me wrong, it will not unlock the possibility of saving to such a folder, and even more so in games that do not support HDD - the point is that you can personalize the appearance of a given corner on this particular partition. Personally, I found the use of this function in the form of separate folders for save files from PSX and PS2, something like a storage where you can copy them back to real memory cards if needed and of course as a form of an additional backup.

Editing is best done in a hex editor, because no program I know allows you to change the attributes of selected abstracts, but then you need to understand how the PFS filesystem is built, otherwise you risk damaging the edited partition and data loss.

Fortunately, there is a tedious but completely safe way to do this. HDLGame Installer saves the settings in "Your Saves", and if this folder is not present, it creates it with the appropriate attributes in the array. It is enough to enter the settings in this program (under the "Select" button), select "OK" to save the changes to the disk and turn on uLE. In it, rename the folder to the desired one, and delete the directory with HDLGI settings. Repeat the operation until you succeed, that is as many times as you want to have such "special folders". Here are two examples:



hdd0: / __ common / Saves Backup (PSX) /

hdd0: / __ common / Saves Backup (PS2) /

Spring cleaning

Homebrew programs, after formatting the disk, create several partitions, which I briefly described in the chapter on basics. Due to the user's needs, some are redundant (you can remove them with **uLE kHn** (because neither uLE, nor "wLE" in the current versions will not allow it), and if necessary, you always have the opportunity to create them anew). And if this guide gave you a headache;) and you decided to upload any of the HDD images shared by the community, then you will find out what you can get rid of with a clear conscience.

- **If you are** satisfied with FMCB on the memory card, and you are interested in nothing more than uploading PS2 games to HDD, you can delete all partitions except "__mbr". You will recover all the free disk space, minus 128MiB .

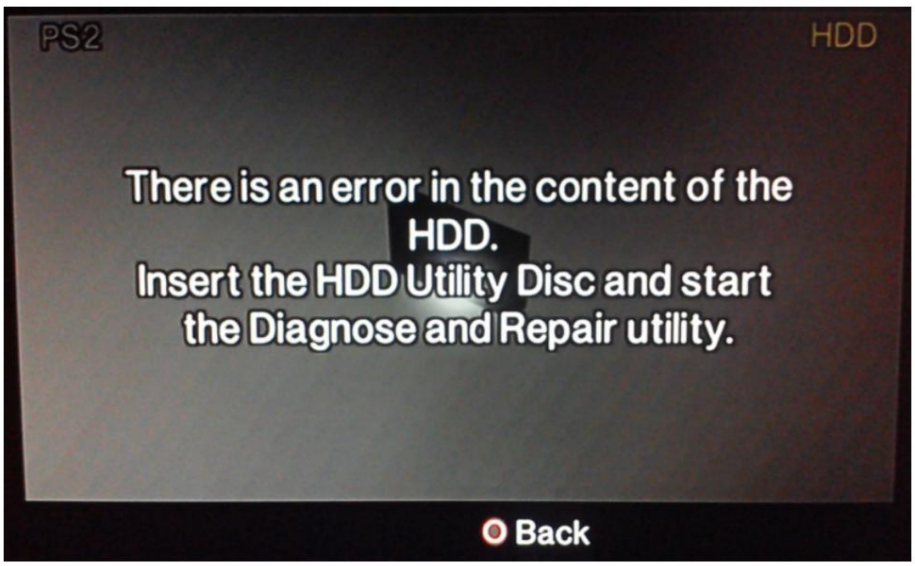
- **If** you have the "__boot" partition, but you are not using the DEV2 modchip mode, you can delete it and free it all 128MiB.

- **If you are** not going to install PSBBN , you can delete the "__net" partition. You will gain 128MiB .

- **If you** already have PSBBN installed but you don't want it and you don't want to format the media, then get rid of "__net", "__contents", "PP.BB", all "PP. <PSBB games>" and all " linux. <number> ". You release the dizzying 10240MiB + (n * 128). Additionally, you can also delete "hdd0: / __ system / p2lboot /".

- **If you** don't want to upload FHDB or HDD OSD then "__system", "__sysconf" and "__common" won't be needed for anything. You will release 1792MiB . However, if you have already installed the FHDB, after removing them, the console will fall into the so-called **boot loop** until you format the disk (or overwrite "__mbr").

- **If you are** not going to emulate PSX games with the official emulator then the "__common" partition is basically redundant. Like "__. POPS" and all "PP. <Game title for PSX>". However, if you are using the HDD OSD, after removing "__common", "browser" will display an error message every time (can be ignored). You release 128MiB + (n * 128).

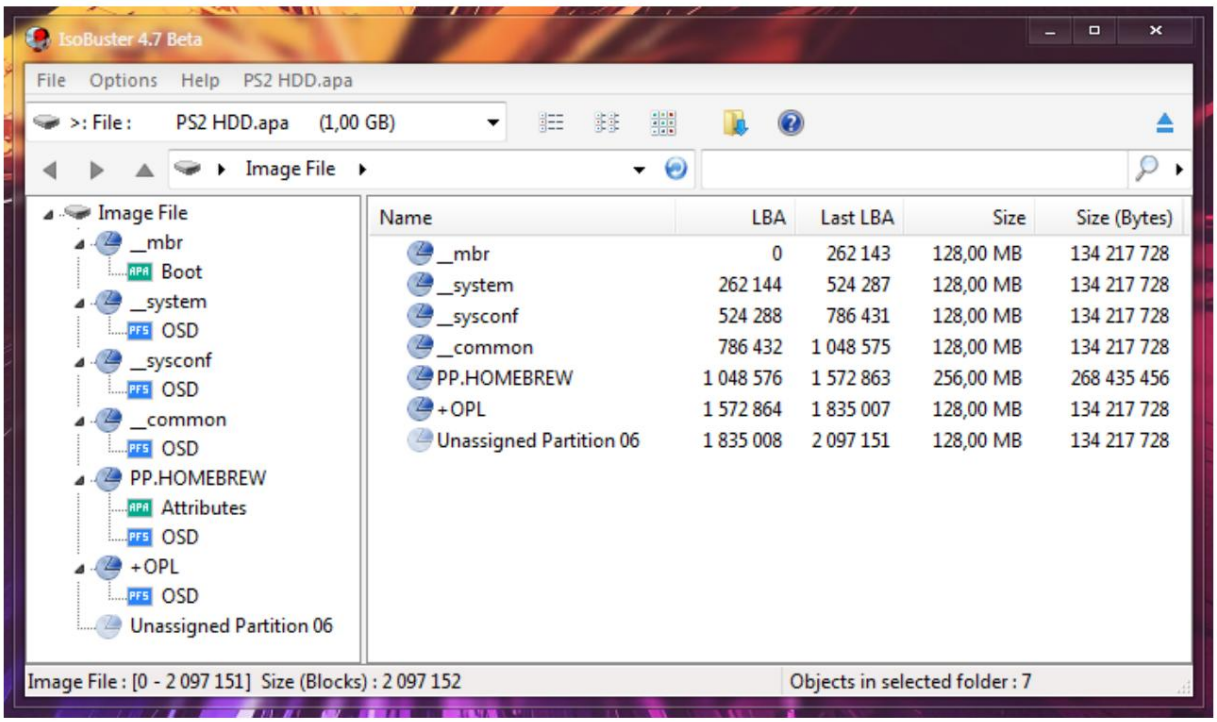


- **If you have** put any programs into the "APPS-HDD" directory before installing FHDB, and each megabyte is worth its weight in gold, move the programs to another partition, correct the paths to them in the FHDB configuration file and delete the "PP.FHDB.APPS" partition . You will gain 128MiB .

- **If** you want to defragment partitions and get rid of empty spaces, use the HDD program Checker, in which select "Optimize disk".

- **If you** would like to keep all system partitions but keep their size to a minimum, this is not possible at the moment. Instead, you can reformat the hard disk, delete all partitions (except "__mbr") and recreate it by defining your own sizes. Only in this way is it possible to keep all partitions required by various Sony software while saving a maximum of 1536MiB.

  Note: the minimum size of the hard disk cannot be less than 16GiB. Otherwise, all variants of uLE and PFS Shell will create partitions smaller than APA specification for "__net" (even 8MiB! It depends on the maximum HDD capacity), but it will not be possible to create any new partitions, therefore this size it is borderline at the moment.

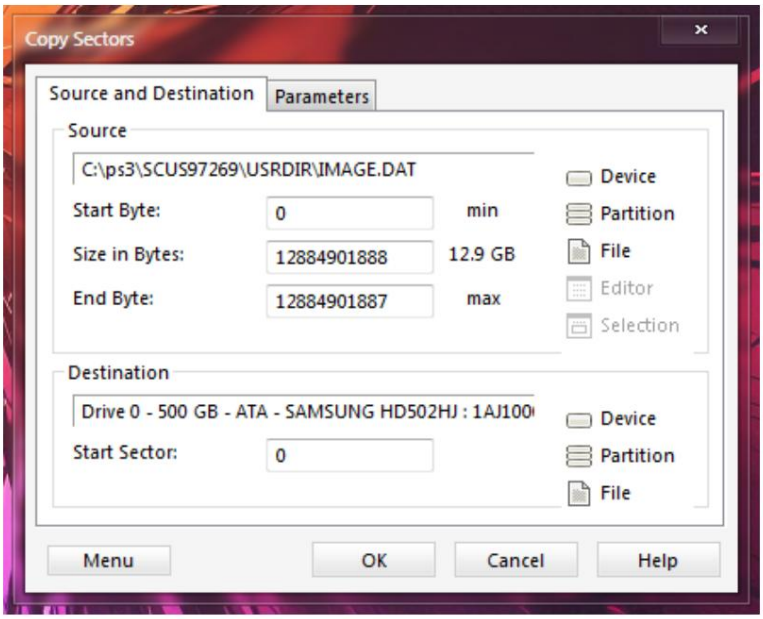


An example of how it looks in my size.

IMAGE.DAT

On backwards compatible **PlayStation** 3s that had Emotion Engine and Graphics Synthesizer soldered on board (or only one of them), Sony made it possible to **install a few selected games** on the PS2 virtual hard drive, if the user had previously installed a **PS2 System Data** package from PlayStation Store containing the template used through these games to install your data.

The said virtual disk is an "IMAGE.DAT" file which is a sectoral image of the PS2 hard disk.
You can simply upload it to a real HDD (eg using **DMDE** or **HDD Raw Copy Tool)** and probably after installing the HDD OSD, run the game from the partition displayed in the Browser.



The file is located in "dev_hdd0 / game / <GameID> /USRDIR/IMAGE.DAT" and can be sent to a PC via FTP (or if it is smaller than 4GiB, then to a USB flash drive using a file manager, e.g. built into the **multiMAN).**

* * *

Remaining issues to be discussed:

• Preparation of Linux Black Rhino partition in APA space. • Installation of PlayStation BroadBand Navigator

Thanks to:

• **Jimmikaelkael, Neme,** and other souls involved in the FMCB v1.8x project • **SP193** for titanic work on FMCB / FHDB 1.9xx, comments and **documentation** • **krHACKen** for SUDC, POPStarter, SCEDoormat, dumpy and tips • row of good souls, constantly working hard on OPL • **ffgriever** for ESR • **Shaolin Assassin** for excellent **documentation** on POPS and POPStarter • **Zabuza** for the Network Adapter photo • **KaiQ** for giving me the Network Adapter • **Haker120** (aka **A youkai of love)** for dumpy, additional comments on the guide and help with testing • **Sandungas** for IMAGE.DAT samples • **Darkshadow**_ for explaining the calculation algorithm partition header checksum • **Pinky1** for writing a program to correct partition header checksums

Berion
2020-10-29