Disc image compression

Systems from the 5th and 6th console generation often use optical discs like CD or DVD to store game data. Disc image sizes range from 700 MB (CD), 1.4 GB (GC Mini-DVD), 4.7 GB (single-layered DVD) and even 25 GB (Blu-Ray) for newer systems. Often times only a fraction of the actual disc size is used for game data and the rest is filled with dummy data to be optimized for read access by the optical drive laser.

The Super Mario 25th Anniversary Wii disc itself is a 4.7GB, when really the actual game data is only a single SNES ROM (12 MB of useful data). Disc compression helps to save more than 99% of hard disk space and can speed up loading time on emulators.

On the other side systems like the Sony PlayStation often use multi-track images (data and audio tracks) containing several *.bin and *.cue files which can become cumbersome in daily usage with Batocera or other emulation frontends.

Choosing the right disc image compression can help in both cases and should be the preferred format for most of the disc based systems. This page will help you to find the suitable compression method and toolset.

If you are not familiar with using command line tools there are is also a GUI-based helper tool available:

- **game-converter**: a GUI for Linux

CHD

CHD is short for (C)ompressed (H)unks of (D)ata. It is a lossless compression format originally developed for MAME, for the hard-drive contents of certain arcade machines. It has since been used in several other emulators as a means of storing disc based game data. It compresses the contents of a disc image to a handy single .chd file. Here's a comparison between 7z file compression and CHD (originally an 700 MB image):

### Supported systems

CHD is supported by many systems and the recommended format for the following systems:
• 3do
• dreamcast
• segacd
• neogeocd
• pcenginecd
• psx
• ps2 (available with Batocera v31)
• saturn

If you have one of the PAL PlayStation games that features LibCrypt copy protection, you have a .sbi file in addition to the .bin/.cue file. The CHD creation process doesn't process the .sbi file. Therefore you will need the .sbi file in the same directory as the .chd file for the game to run.

Creating CHDs

CHD files can be created using the chdman program, developed by the MAME project.

chdman is a commmand line application. You can create a .chd file by targeting *.cue, *.iso or *.gdi files.

If there is a *.cue file available, always use this for creating the .chd. Otherwise choose the *.iso (Sony PlayStation 2) or *.gdi (Dreamcast) file. Note that GameCube/Wii have their own compressed format.

Open the command line in either Windows with cmd.exe or in Linux with your favorite terminal emulator. Put both chdman and the game file in the same folder, navigate to that folder in your terminal and run:

```bash
chdman createcd -i <game.cue> -o <game.chd>
```

For example, if the file structure was like so:

```
/totally-rad-folder-name/
├── chdman
│   ├── Spiderman.cue
│   └── Spiderman.bin
```

then first cd /totally-rad-folder-name then run:

```bash
chdman createcd -i Spiderman.cue -o Spiderman.chd
```

Then make yourself a cup of tea. This can take a while. After that you'll have <game.chd> and you can delete the *.cue/*.bin, *.iso or *.gdi/*.bin file(s).

The compression is fully lossless so it is possible to redo the compression and get the original files back.
CHD batch converting

You can batch convert all files in a directory like following (replace .cue with the correct file type you want to convert):

**Linux/macOS:**

On Linux you can install the mame-tools package on most distributions. On macOS use brew to install the package rom-tools. Then you can navigate to the folder that has all your game ROMs and run the following for loop to convert all the files in one go:

```
for i in *.cue; do chdman createcd -i "${i}" -o "${i%.*}.chd"; done
```

replacing *.cue as needed. If you're converting a lot of files, practice some yoga poses. This may take a while.

**Windows:**

Just place the chdman.exe into the directory where you have your disc images. The official MAME package contains a copy of the executable file. You can download the EXE file and open it up like an archive with 7-zip, then extract chdman.exe.

Here's some handy BAT scripts you can just double-click to instantly convert all the games in the current folder to CHD:

For *.cue:

```
cue-to-chd.bat

for /r %%i in (*.cue) do chdman createcd -i "%%i" -o "%%~ni.chd"
```

For *.iso:

```
iso-to-chd.bat

for /r %%i in (*.iso) do chdman createcd -i "%%i" -o "%%~ni.chd"
```

For *.gdi:

```
gdi-to-chd.bat

for /r %%i in (*.gdi) do chdman createcd -i "%%i" -o "%%~ni.chd"
```

If you're converting a lot of files, play some tunes on your guitar. This may take a while.
CSO

CSO is a compression method for the ISO image format and is the short form for Compressed ISO. It also called CISO.

It was originally used to compress PlayStation Portable UMD disc images.

Supported systems

CSO/CISO is supported by multiple systems and the recommended format for the following systems:

- ps2 (prior to Batocera v31)
- psp

The Wii/Gamecube also support a CSO/CISO format, that is unrelated to the compression method explained here. More information can be found at the wit: Wiimms ISO Tool.

Creating CSOs

There are different tools to create CSO files. maxcso is a free command line application that is available to Linux, macOS and Windows.

To compress an ISO file just use:

```
maxcso <game.iso>
```

This will create a `<game.cso>` file in the same folder. Afterwards you can delete the `.iso` files to free up space.

The compression is fully lossless so it is possible to redo the compression and get the original files back:

```
maxcso --decompress <game.cso>
```

CSO batch converting

Batch converting can be done with a simple for loop.

**Linux/macOS:**

```
for i in *.iso; do maxcso "${i}"; done
```

**Windows:**

...
for %i in (*.iso) do maxcso.exe "%i"

RVZ

The Dolphin team developed a new compression format based on WIA called RVZ. Unlike all the previous formats, RVZ is lossless and can preserve the padding data on Wii discs as well as the necessary files needed by the Wii's IOS. Usage is very similar to GCZ in Dolphin itself, but it only works on newer Dolphin versions.

Supported systems

RVZ is only supported by the Dolphin emulator and therefore the recommended format for the following systems:

- gamecube
- wii

Creating RVZs

You need the Dolphin emulator (minimum version 5.0-12188+) to compress ISO files. At the moment the compression has to be done via the User interface:

Find your game in Dolphins game list, right click and select “Convert File…”

The default settings should be fine. If you want you can raise the compression level.
You can convert your game image back to ISO from RVZ when you want to restore it back to an exact replica of your game ISO the same way.

**RVZ batch converting**

Just multi-select with pressing the *CTRL* key to choose multiple games and select “Convert File...”

**SquashFS**

SquashFS allows you to losslessly compress entire folders and all their contained items. This image can be extracted to be identical to the files that were compressed. Unlike ZIP/7Z/GZ files, SquashFS allows Batocera to present the contents of the compressed image to the system as if though it weren't compressed at all by automatically mounting them upon launch. This has been a latent feature but full integration with the tools necessary to handle SquashFS images was introduced in Batocera v33.

There are some limitations to keep in mind:

- **SquashFS is read-only.** Obviously, this causes issues with games that attempt to write to the directory it has been installed to, which is typical for many PC games to do. Some are exceptions, such as basic DOS titles. There are workarounds for this, read below.
- **Only systems that specify in their `_info.txt` as supporting *squashfs* will be able to launch SquashFS images.

SquashFS is particularly good for PS3 games, which are stored as folders which are only read by the system.
Creating a SquashFS file

This is currently only possible via SSH. Batocera utilizes mksquashfs to create SquashFS images. Syntax: mksquashfs <game> <game>.squashfs.

For example:

```
cd /userdata/roms/dos
mksquashfs skweek.pc skweek.pc.squashfs
```

Or to compress a single file:

```
cd /userdata/roms/xbox
mksquashfs crazytaxi3.iso crazytaxi3.iso.squashfs
```

Or to compress a single file with the xz compression option:

```
mksquashfs crazytaxi3.iso crazytaxi3.iso.squashfs -comp xz
```

After confirming that the game still launches and runs correctly, the old folder can be safely removed.

If you use the btrfs filesystem, you can enable compression, and you'll get the same compression result without any action.

Technical details

First, to be listed in EmulationStation, SquashFS images must be added technically by a developer to the list of eligible systems. (aka, dos and ps3 are added for the moment, but it very easy to add any system while it supports folder as rom). Squashfs images are mounted under /var/run/squashfs/<rom> and this is this directory that is given to the emulator.

From:
https://wiki.batocera.org/ - Batocera.linux - Wiki

Permanent link:
https://wiki.batocera.org/disk_image_compression

Last update: 2022/02/14 07:48