Different images formats

Disc images/ROMs come in a variety of formats, and over the years many formats have risen and fallen in popularity. This article focuses on the more common ones you may come across.

For information on compressed disk images take a look at Disc image compression.

What are disc images?

Disc images are a handy way of storing backup copies of your disks. For instance, you might have some Audio CDs which you play on a regular basis. You probably don't want them to wear out fast because you use them often. In that case, a good way to listen to them but also keep them in a pristine condition - because you will actually not use them - is to create disc images for them. The disk images can be stored on your computer and you can virtually mount them when you want to listen to your music. Obviously, you can rip their contents into another format such as `.mp3`, but that's another story.

Batocera can these disc images to load the contents of a game ROM into an emulator as it would have done so on the actual hardware. This is considered the most accurate way of loading the content. Fun fact: Batocera itself comes as a disc image; you're just using Etcher to burn that image onto a drive!

To sum it a bit up, here are some of the most prominent benefits of using disk images:

- Disk images are exact replicas of disk drives or disk volumes, so they faithfully preserve all details related not only to content but also to the original files and folders structure.
- A disk image of an optical disk can be very useful for backing up purposes.

Common file formats for disk images

As we know by now, a disk image is a file stored on your disk. Like any file, it must bear a name and an extension. In other words, a file must have a file format. The most common disk image file format today is “ .ISO ” , but there are many other types of file formats that can be used. Here are some of them:

- `.iso` - used for a variety of uncompressed disc image formats, refer to below for more info.
- `.nrg` (Nero CD/DVD Image File) - are CD or DVD disk images created with the Nero disc authoring software.
- `.bin` and `.cue` (Binary file and Cue sheet) - are CD or DVD disk images split into two different files. The `.bin` file that is a binary file that contains one data/audio track of the disc (there may be multiple `.bin` files). The complementary `.cue` file contains the details on how the data is structured on the original disk. When loading a disc like this, target the `.cue` file, not the `.bin` file.
- `.mdf` and `.mds` (Media disk Image File and Media Descriptor File) - the CD or DVD data is stored

inside the `.mdf` file, while the header and track information are stored in the `.mds` file.

This information has been surmised from https://www.digitalcitizen.life/simple-questions-what-disc-image-file-iso-nrg-bin and https://en.wikipedia.org/wiki/ISO_image

ISO and IMG

There is no standard definition for ISO image files. ISO disc images are uncompressed and do not use a particular container format; they are a sector-by-sector copy of the data on an optical disc, stored inside a binary file. ISO images are expected to contain the binary image of an optical media file system (usually ISO 9660 and its extensions or UDF), including the data in its files in binary format, copied exactly as they were stored on the disc. The data inside the ISO image will be structured according to the file system that was used on the optical disc from which it was created.

ISO files store only the user data from each sector on an optical disc, ignoring the control headers and error correction data, and are therefore slightly smaller than a raw disc image of optical media. Since the size of the user data portion of a sector (logical sector) in data optical discs is 2,048 bytes, the size of an ISO image will be a multiple of 2,048. The `.iso` file extension is the one most commonly used for this type of disc images. The `.img` extension can also be found on some ISO image files, such as in some images from Microsoft DreamSpark; however, IMG files, which also use the `.img` extension, tend to have slightly different contents. The `.udf` file extension is sometimes used to indicate that the file system inside the ISO image is actually UDF and not ISO 9660.

Can't I use just the .bin file then?

… or I can't see my PSX/Dreamcast/Saturn/Gamecube/Wii ROMs!

Some emulators are capable of loading games from just the `.bin` file, however they might do so with glitches in the game (depending on how the data was stored) or with missing audio (tyically the BGM, but possibly other sounds too). You need the `.cue` sheet file that describes the `.bin` file(s) in order to properly load all the content on the disc. Sometimes game/audio data is spread out across multiple `.bin` tracks; a good example is *Rockman 8 - Metal Heroes (Japan)* for PSX which has the following file structure:

```
Rockman 8 - Metal Heroes (Japan) (Track 1).bin
Rockman 8 - Metal Heroes (Japan) (Track 2).bin
Rockman 8 - Metal Heroes (Japan) (Track 3).bin
Rockman 8 - Metal Heroes (Japan) (Track 4).bin
Rockman 8 - Metal Heroes (Japan).cue
```

This game contains 1 DATA-track, 3 AUDIO-tracks (tracks 2-4) and one CUE-Sheet file. In Batocera versions prior to v31, you would see 5 entries for this game in your game list. However, versions v31 and above will automatically ignore the extra files by default. If you have an old list generated by an older version of Batocera, you can remove the `gamelist.xml` file in the appropriate system's rom directory to force Batocera to regenerate it (you will lose any custom edits you have made to it, of course!)

CUE/SBI/GDI sheet recovery

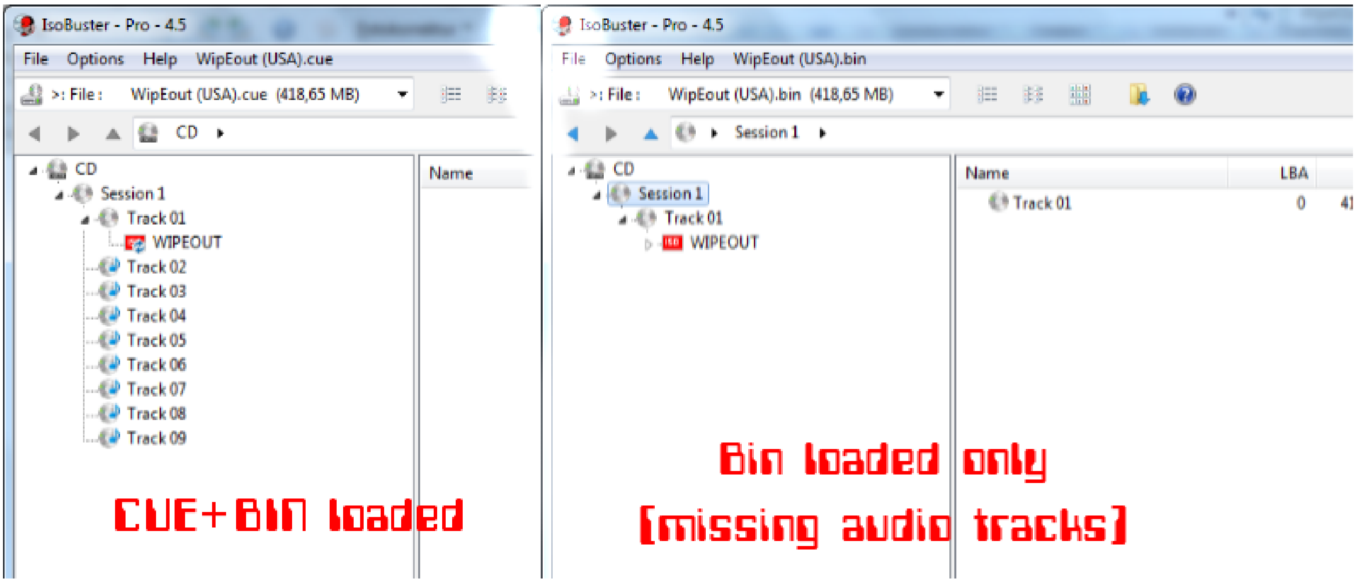
If you have only the `.bin` file and not the `.cue` file, you can do the following:

- Re-dump the disc, making sure to retain the generated `.cue` file.
- Visit http://redump.org/, search for your image (make sure it's the right region!) and download the `.cue` file. This also includes `.sbi` files if needed.
- Generate the `.cue` file online with CUEMAKER (multi-bin).
- Another online generator PSX Cue File Maker (multi-bin).
- Pick up some premade ones (Github). This also includes `.sbi` files if needed.
- Use a dedicated tool, like Liors CUE Maker (for Windows).

If you'd like to simplify your collection in your file manager, you can use the CHD compressed image format (it will consolidate both the `.cue` and `.bin` files into a single `.chd` file). However, this format is not compatible with all emulators, check the `_info.txt` file in the rom directory to see which formats your emulator supports.

I still don't understand....? C.U.E.??

Even you merged all tracks to a big huge single `.bin` file, you still need the `.cue` file. I'll show you :) This is the same for the `.img` format! (I think CDRWin used this)



Multi-disc games

Some games included multiple discs, which you had to switch between at certain points in the game. One example of this is Final Fantasy VII on PSX. Let's use that.

To automatically load the next disc of a game, you can use a `.m3u` playlist file.

Please note that even if `_info.txt` says `.m3u` is a supported format, not all standalone emulators/Libretro cores support loading discs in this way.

To make one, simply create a text file with the same filename as your intended game name (this is what will appear in your game list. It could be anything, really). Within that text file, write the names of the `.cue`/`.gdi`/`.mdf` sheets or `.chd` files for your game discs.

If creating the playlist file with Windows, be aware of the line ending limitations. For most systems, this does not matter, but some will only read playlist files if they use Linux-style line terminators.

Batocera will automatically hide duplicated entries showing in the gamelist when using playlists to define which discs are a part of a single game. However, this will only work when the playlists are referencing just the filenames; relative pathing will override this behaviour and show all discs. To put it simply:

- `filename.cue` → hide this disc
- `./filename.cue` → don't hide this disc

All the files inside of the applicable ROM folder

For instance, if your game's .cue sheets were structured like:

```
roms/
└─ psx/
   ├─ Final Fantasy VII (Disc 1).bin
   ├─ Final Fantasy VII (Disc 1).cue
   ├─ Final Fantasy VII (Disc 2).bin
   ├─ Final Fantasy VII (Disc 2).cue
   ├─ Final Fantasy VII (Disc 3).bin
   └─ Final Fantasy VII (Disc 3).cue
```

you would put the following as text into the `Final Fantasy VII.m3u` text file:

Final Fantasy VII.m3u

```
Final Fantasy VII (Disc 1).cue
Final Fantasy VII (Disc 2).cue
Final Fantasy VII (Disc 3).cue
```

Save the text file with the file extension `.m3u` and place it in the same folder as the game's discs. Your final folder structure should look like this:

```
roms/
└─ psx/
   ├─ Final Fantasy VII.m3u
   ├─ Final Fantasy VII (Disc 1).bin
   ├─ Final Fantasy VII (Disc 1).cue
   ├─ Final Fantasy VII (Disc 2).bin
   ├─ Final Fantasy VII (Disc 2).cue
   ├─ Final Fantasy VII (Disc 3).bin
   └─ Final Fantasy VII (Disc 3).cue
```

A subfolder for each multi-disc game

You can opt to put all of your multi-disc games in their own subfolders instead for organization purposes. Like so:

```
roms/
└─ psx/
   └─ Final Fantasty VII/
      ├─ Final Fantasy VII (Disc 1).bin
      ├─ Final Fantasy VII (Disc 1).cue
      ├─ Final Fantasy VII (Disc 2).bin
      ├─ Final Fantasy VII (Disc 2).cue
      ├─ Final Fantasy VII (Disc 3).bin
      └─ Final Fantasy VII (Disc 3).cue
```

In the case that you've organized your ROMs like this, you can use the following scripts/SSH commands to automatically generate the M3U playlists and put them into the appropriate subfolders:

Click to expand

- **via SSH** **TENTATIVE (needs some more testing, run at your own risk!)**First, change the current directory to the system's ROM folder with `cd <path/to/folder>`. For example:

  ```
  cd /userdata/roms/psx
  ```

  - For `.cue` files:
    - Run the following:

      ```
      for DIR in *; do echo "$(for file in "$DIR"/*.cue; do echo "$(basename "$file")"; done)" > "$DIR"/"$DIR".m3u; done
      ```

  - For `.chd` files:
    - Just run the following:

```
for DIR in *; do echo "$(for file in
"$DIR"/*.chd; do echo "$(basename "$file")";
done)" > "$DIR"/"$DIR".m3u; done
```

- **Accessible the drive directly on Windows** Place the script in the `roms/<system>` folder and double-click/execute it. If you want to apply this to *all* CUE and/or CHD multi-disc games in all systems, run it from `roms/` instead.
    - For NTFS userdata filesystems:
        - For `.cue` files:

          ntfs-generate-m3u-for-cue-in-subfolders.bat

          ```
          @echo off
          for /R "%~dp0" %%I in (*.cue) do for %%J in
          ("%%~dpI.") do echo
          %%~nxI>>"%%~dpI%%~nxJ.m3u"
          ```

        - For `.chd` files:

          ntfs-generate-m3u-for-chd-in-subfolders.bat

          ```
          @echo off
          for /R "%~dp0" %%I in (*.chd) do for %%J in
          ("%%~dpI.") do echo
          %%~nxI>>"%%~dpI%%~nxJ.m3u"
          ```

    - For FAT userdata filesystems:
        - For `.cue` files:

          fat-generate-m3u-for-cue-in-subfolders.bat

          ```
          @echo off
          for /F "eol=| delims=" %%I in ('dir
          "%~dp0*.cue" /A-D-H /B /ON /S 2^>nul') do
          for %%J in ("%%~dpI.") do echo
          %%~nxI>>"%%~dpI%%~nxJ.m3u"
          ```

        - For `.chd` files:

          fat-generate-m3u-for-chd-in-subfolders.bat

          ```
          @echo off
          for /F "eol=| delims=" %%I in ('dir
          "%~dp0*.chd" /A-D-H /B /ON /S 2^>nul') do
          for %%J in ("%%~dpI.") do echo
          ```

```
%%~nxI>>"%%~dpI%%~nxJ.m3u"
```

From:
https://forums.launchbox-app.com/topic/53467-batch-file-to-auto-generate-m3us/

The M3U `Final Fantasy VII.m3u` playlist should contain the following:

Final Fantasy VII.m3u

```
Final Fantasy VII (Disc 1).cue
Final Fantasy VII (Disc 2).cue
Final Fantasy VII (Disc 3).cue
```

If doing this by hand, save the text with the file extension `.m3u` and **place it in that multi-disc game's specific subfolder**. If using the automated BAT script, the files are automatically placed in the correct subfolders. Your final folder structure should look like this:

```
roms/
└─ psx/
   └─ Final Fantasty VII/
      ├─ Final Fantasy VII.m3u
      ├─ Final Fantasy VII (Disc 1).bin
      ├─ Final Fantasy VII (Disc 1).cue
      ├─ Final Fantasy VII (Disc 2).bin
      ├─ Final Fantasy VII (Disc 2).cue
      ├─ Final Fantasy VII (Disc 3).bin
      └─ Final Fantasy VII (Disc 3).cue
```

A subfolder for each disc

In case you want to have your disc images inside of separate subfolders, and you'd like your `.m3u` to be one level above, you can append a period (.) followed by the folder directory relative to the `.m3u`'s location for each line in the `.m3u` file. For instance if you wanted your folder structure to look like this:

```
roms/
└─ psx/
   ├─ Final Fantasy VII (Disc 1)/
   │  ├─ disc.bin
   │  └─ sheet.cue
   ├─ Final Fantasy VII (Disc 2)/
   │  ├─ disc.bin
   │  └─ sheet.cue
   ├─ Final Fantasy VII (Disc 3)/
   │  ├─ disc.bin
```

```
    │   └─ sheet.cue
    └─ Final Fantasy VII.m3u
```

you would put the following as text into the `Final Fantasy VII.m3u` file:

Final Fantasy VII.m3u

```
./Final Fantasy VII (Disc 1)/sheet.cue
./Final Fantasy VII (Disc 2)/sheet.cue
./Final Fantasy VII (Disc 3)/sheet.cue
```

If you set it up with separate subfolders for each disc, Batocera will display duplicate entries! It can only recognize and remove extra discs for a single game if they are in the same folder. Alternatively, you can just have a single folder for all that game's discs along with the `.m3u` file and only the `.m3u` file will appear.

After setting up your M3U playlists correctly

Batocera should show you the `.m3u` as a single game entry (in **v31** and up) which will contain all discs. When you get to the end of the disc you are playing, the next disc will be automatically loaded.

M3U playlist automatic disc control is confirmed to be working with the following emulators:

- PSX
    - Libretro/PCSX-ReARMed
    - DuckStation
    - Libretro/DuckStation
- SegaCD
- SEGA Saturn
- SEGA Dreamcast

Manual disc swapping

In case the automatic disc swapping fails, refer to these manual disc swapping instructions below.

RetroArch (most libretro cores)

You can utilize RetroArch's **Disc Control** menu in the Quick Menu (`[HOTKEY]` + ● button in-game). Select **Eject Disc**, then go to **Current Disc Index** and change your disc to the next one. Then, **Insert Disc** to continue the game.

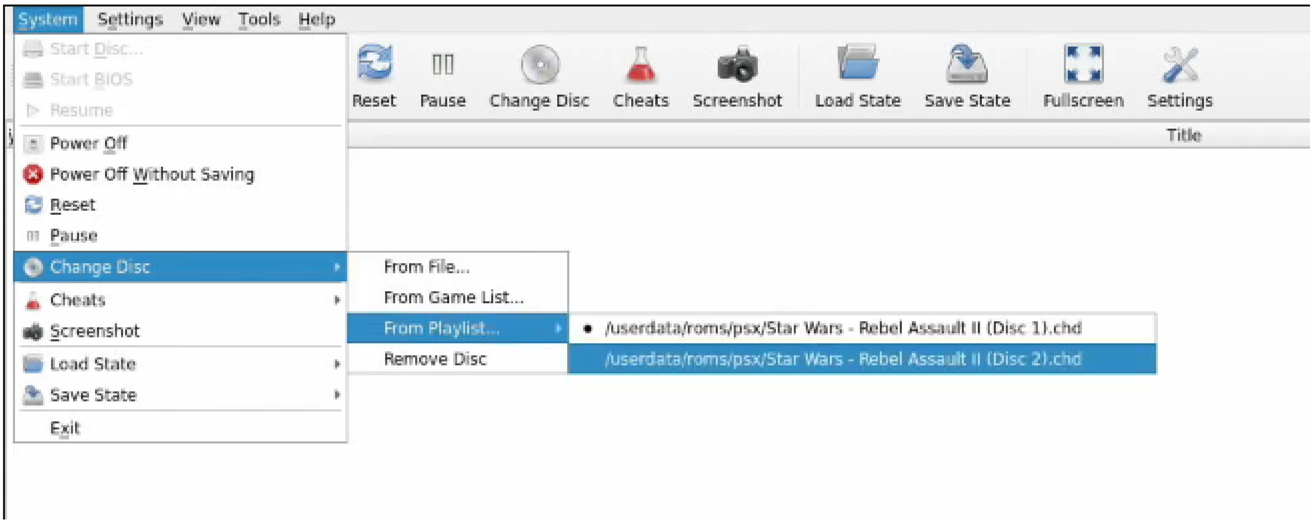
Swap Disc is for legacy purposes and should not be used.

The following cores are confirmed to not be working with the disc control menu:

- SEGA Saturn
  - Libretro/Yabause

DuckStation (standalone)

You'll need a keyboard for this method. While playing the game, press `[Alt]+[Tab]` to switch focus to the standalone emulator config. Then, using your mouse, click on “Change Disc” or navigate to **System** > **Change Disc** > **From Playlist…** and choose the next disc.



libretro: Opera (3DO)

From Batocera **v34**, enable the **NVRAM STORAGE** option from the advanced system settings (`[SELECT]` in the game list) for the multi-disc game's discs to allow them to automatically use the same save file. Then load the second disc from ES after completing the first one. Do note that this shared storage has the real-world limitations of the 3DO's drive space.

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

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

Last update: **2022/10/01 06:15**