Creating high quality bootable games from GD files

by TuxTheWise

Why would I want to do that?

- Make your own rips, so you can use files of a trusted source and be sure there are not corrupted files.
- Improve reading efficiency using dummy file and correct file order.
- Release a rare game that nobody have done yet.

1. Introduction

I've been making high quality releases lately, but unfortunatly my summer vacations will be finished soon. So, before I lose my free time, I decided to write this guide to help anyone who wants to make their own releases. Although it's a basic guide, if you fully understand it a door of possibilities will open to you.

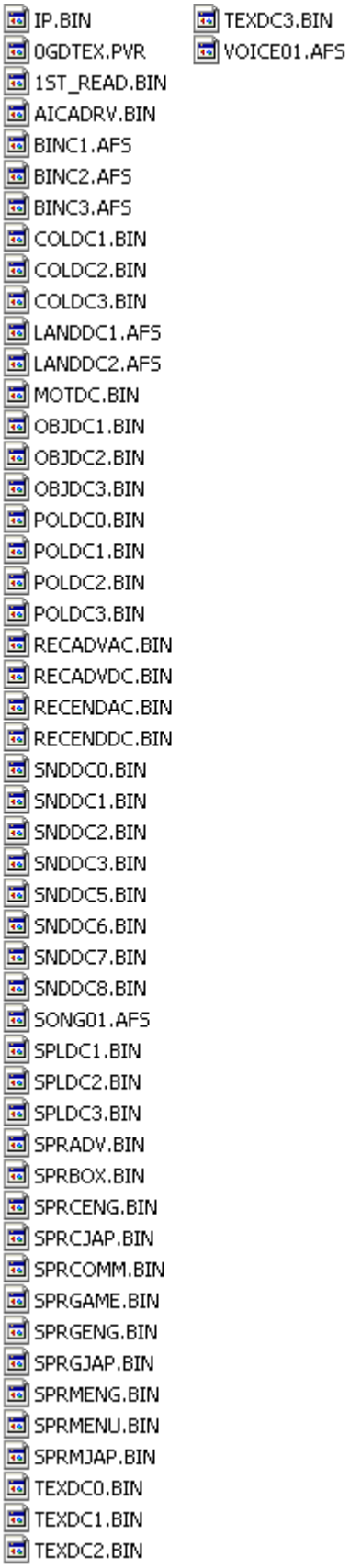
This guide will cover Crazy Taxi. I recommend you to test with this game, GDI is pretty easy to find, small, and you can easily notice the reading improvement techniques of this guide.

You need to extract the game files from track03.bin, extract IP.BIN and have extracted file order using IsoBuster. To do that, follow my other tutorial "Extracting GDI dump files" (included in this pack).

So, to start this guide you need the files extracted from a GD dump. This guide will only work on Katana games (no WinCE games, I'll make a tutorial for them later), and without protection. If your game has CDDA, CDDA won't play either (again, another tutorial for this will come later). A lot of games don't have any kind of protection and don't use CDDA, mostly the first ones releases. Examples without protection: Crazy Taxi, Tony Hawk's Pro Skater 1 and 2, Shadow Man, Legacy Of Kain: Soul Reaver, Resident Evil CODE: Veronica (and its japanese "Complete" bro). Examples of games with protection: Sonic Adventure 2, Fatal Fury: Mark Of The Wolves, Metropolis Street Racer. But even if a game is protected, you can use techniques of this tutorial to make already cracked releases (Echelon's, Kalisto's) better.

Another commentary, the files of the GD must fit on a CD (remember that some extra space is needed for selfboot session). If you use Crazy Taxi like this guide, it will fit easily. If you try a Tony Hawk, for example, files won't fit. Then you'll need to use downsampling techniques, that will be covered in another future guide.

Okay, so let's start with Crazy Taxi. Using "Extracting GDI dump files" tutorial files were extracted (including IP.BIN). Here are the files:



and the file order extracted with IsoBuster:

```
00548218 , >:\0GDTEX.PVR
00548283 , >:\1ST_READ.BIN
00497599 , >:\AICADRV.BIN
00526655 , >:\BINC1.AFS
00520773 , >:\BINC2.AFS
00512113 , >:\BINC3.AFS
00504386 , >:\COLDC1.BIN
00505952 , >:\COLDC2.BIN
00503496 , >:\COLDC3.BIN
00510321 , >:\LANDDC1.AFS
00511217 , >:\LANDDC2.AFS
00498484 , >:\MOTDC.BIN
00502810 , >:\OBJDC1.BIN
00502778 , >:\OBJDC2.BIN
00502746 , >:\OBJDC3.BIN
00499878 , >:\POLDC0.BIN
00502490 , >:\POLDC1.BIN}
00502671 , >:\POLDC2.BIN
00502479 , >:\POLDC3.BIN
00503045 , >:\RECADVAC.BIN
00503098 , >:\RECADVDC.BIN
00502939 , >:\RECENDAC.BIN
00502992 , >:\RECENDDC.BIN
00497627 , >:\SNDDC0.BIN
00498016 , >:\SNDDC1.BIN
00498250 , >:\SNDDC2.BIN
00498341 , >:\SNDDC3.BIN
00498360 , >:\SNDDC5.BIN
00502305 , >:\SNDDC6.BIN
00502407 , >:\SNDDC7.BIN
00498395 , >:\SNDDC8.BIN
00532894 , >:\SONG01.AFS
00503151 , >:\SPLDC1.BIN
00503310 , >:\SPLDC2.BIN
00502842 , >:\SPLDC3.BIN
00502159 , >:\SPRADV.BIN
00502267 , >:\SPRBOX.BIN
00501780 , >:\SPRCENG.BIN
00501828 , >:\SPRCJAP.BIN
00501872 , >:\SPRCOMM.BIN
00509697 , >:\SPRGAME.BIN
00509590 , >:\SPRGENG.BIN
00509650 , >:\SPRGJAP.BIN
00509042 , >:\SPRMENG.BIN
00509518 , >:\SPRMENU.BIN
00509251 , >:\SPRMJAP.BIN
00501030 , >:\TEXDC0.BIN
00508476 , >:\TEXDC1.BIN
00507940 , >:\TEXDC2.BIN
00507549 , >:\TEXDC3.BIN
00513870 , >:\VOICE01.AFS
```

2. Cleaning the garbage

Game is very small, but general rule is always make it the smallest we can, so useless files doesn't messes with file order (separating game files), and, if you need to downsample something (not Crazy Taxi case), you'll have more space and make it with better quality.

2.1. Deleting useless files

You can delete 0GDTEX.PVR. It's that picture shown when you insert GD and go to CD Player on Dreamcast main screen. Note that this picture is not shown on CDs, so

it's useless.

Some games have a file called WARN_ENG.DA, or something like that. I have no idea of what it is, but it can be deleted too.

Depending on the game you're working on, there may be more files that be deleted. Examples: In Tony Hawk's Pro Skater 2, some programmer forgot a .bat used in compiling a file of the game. In Tech Romancer japanese, there is a karaoke video. In American and European edition, video can't be seen in the game, but video file is still on GD. Anyway, it will depend on your ability to identify these files. Remember: if you have doubt, don't delete the file, you may mess the game. And last comment, if game fit on a CD, don't worry too much about that.

Well, in Crazy Taxi example, we can only delete 0GDTEX.PVR.

2.2. Shrinking AFS files

AFS (.afs) are packs of any kind of files. There are some useless space on them, and Echelon made a tool very useful that can delete it. It can be done safely on any AFS, because tool identify no standard AFS and won't touch them. This tool is called "dc_afsshrink.exe" and it's located somewhere on this pack. It will save ~500kb for each AFS file, and don't have any side effect on the game (not that I have seen). To use it, put it on the folder with GD files and use:

```
dc_afsshrink.exe *.afs
```

and it will "shrink" all AFS of the folder. Remember that it won't search on subdirectories, so if your GD has multiple directories you'll have to execute the tool in each one.

Crazy Taxi doesn't have subdirectories, so it only needs to be executed on the main folder. It will save about 3.5mb.

Don't forget to delete/move the program after use.

3. Hacking

Don't worry, explanations here are very simple, you don't need to be an expert to do it.

3.1. Making the game region-free and enabling VGA-Box support

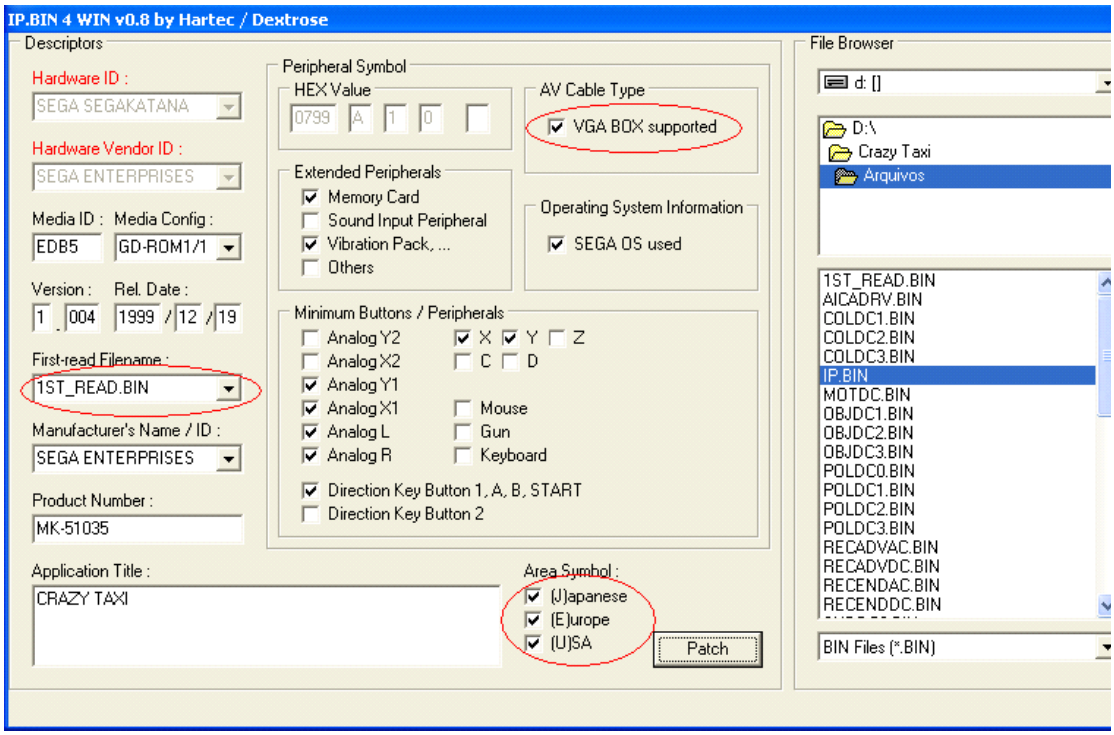
You'll use IPBIN4WIN.exe (located somewhere on this pack). Open the program and select IP.BIN (for the 9999th time, remember that you must have extracted it using "Extracting GDI dump files" tutorial, it doesn't come with game files). Details will be shown on left window. On "Area Symbol", mark all regions. Be sure that "VGA BOX supported" is marked too. After making necessary changes, click in "Patch".

I have no idea if Dreamcast checks region flags when booting CDs, so I don't know if marking regions will make any difference. Mark all anyway.

Games that have no VGA BOX supported by default, may or may not work with the accessory if you mark the flag. Some games work fine if you check it. Other games won't work with VGA even with the flag checked.

Another important thing to see here is "First-read Filename". Common name is "1ST_READ.BIN", but it can be any name. Take note of the name of your game.

Here is a screenshot of Crazy Taxi example with all important parts marked in read:



On this case, VGA Box is marked by default, and we just need to mark the Japanese and Europe regions. Don't forget to click "Patch" after modifing. Note that in this case the first-read filename is "1ST_READ.BIN".

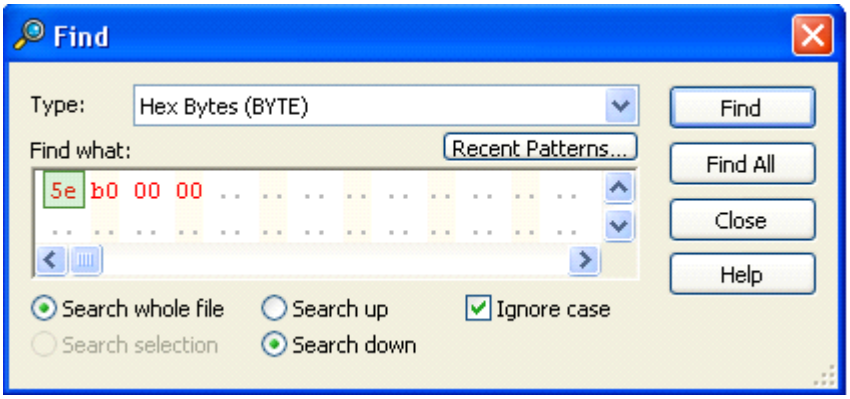
3.2. Hacking first-read filename

This is the more important part of the job. You'll need to modify 4 bytes in first-read file name that I have no idea what they are for. The only thing I know, is that if you don't modify them, game won't boot. Now you can see you're not learning from a very wise guy after all.

Anyway, you'll need to hex edit first-read file name. I recommend Hex Editor Neo, but any other hex editor will work because we're using basic functions only. Anyway, open the file found on last section (example: 1ST_READ.BIN) on hex editor.

You'll have to find this sequence "5E B0 00 00" and modify to "96 00 00 00". You have to modify 2 of this sequences only. If you find more than 2, remember that the ones you want to modify will be close to each other.

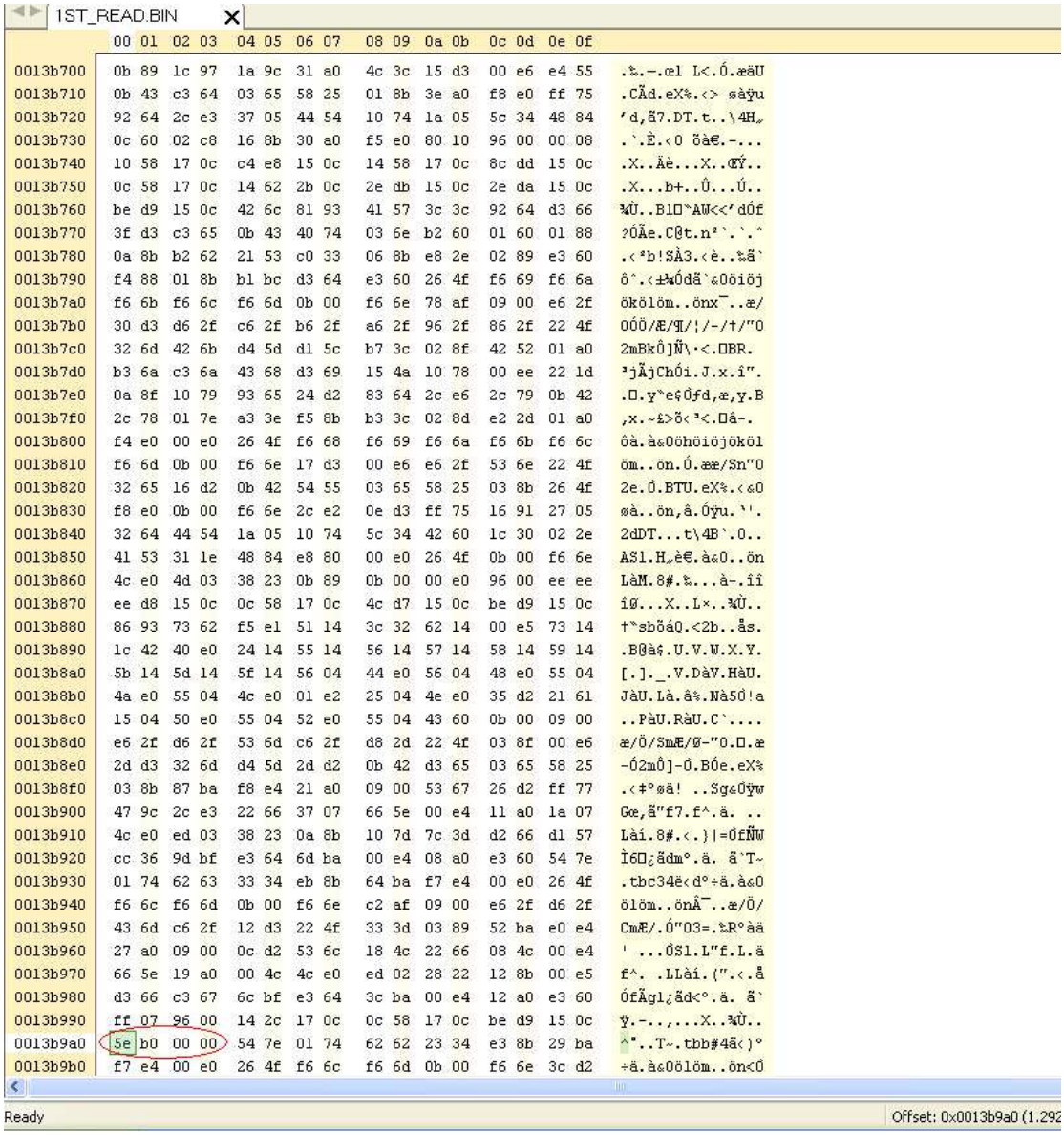
Okay, so let's work on Crazy Taxi. Open 1ST_READ.BIN on Hex Editor Neo. Press CTRL+F (or Edit, Find). In type, select "Hex Bytes (BYTE)". Now type the sequence I told you, like this picture:



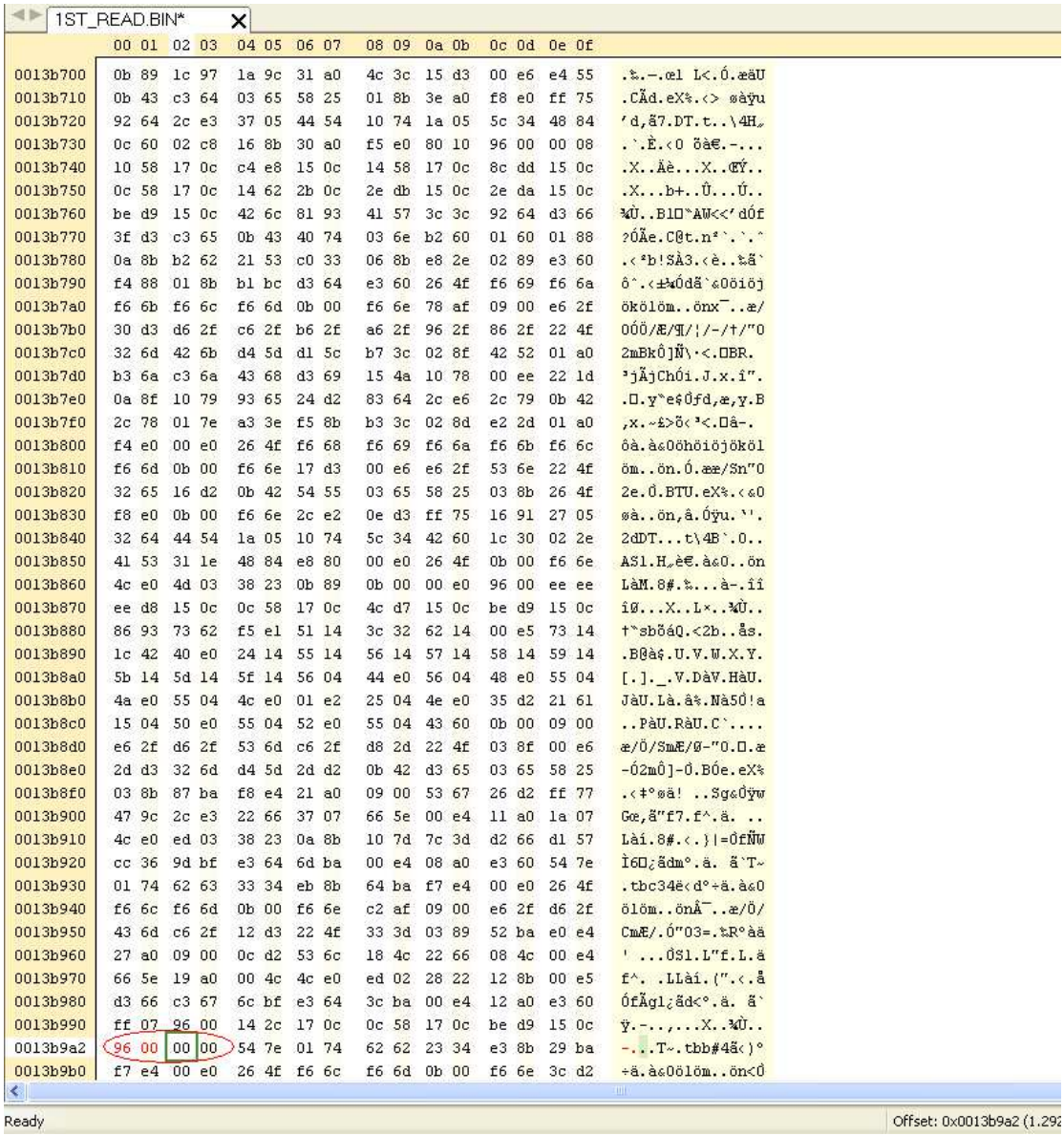
and click Find. For the first time, we're only looking for the sequence and taking note of the offsets (bottom of screen), because we'll find all and see the two closer to each other to know which one to modify. Okay, in Crazy Taxi we find 2 sequences in the positions:

```
1.292.704
1.293.676
```

We only found two copies, so of course they are the ones we have to modify. You can go to them again using "Find" or using CTRL+G (or Edit, Go to Offset) and typing the offset you took note. So, we modify the "5E B0" to "96 00" of both entries (you can alternate between characters and hex windows using key TAB). For the first entry, before we modify:


1ST_READ.BIN
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
0013b700 0b 89 1c 97 1a 9c 31 a0 4c 3c 15 d3 00 e6 e4 55 .‰.—.œ1 L<.Ó.æäU
0013b710 0b 43 c3 64 03 65 58 25 01 8b 3e a0 f8 e0 ff 75 .CÃd.eX%.‹> øàÿu
0013b720 92 64 2c e3 37 05 44 54 10 74 1a 05 5c 34 48 84 'd,ã7.DT.t..\4H„
0013b730 0c 60 02 c8 16 8b 30 a0 f5 e0 80 10 96 00 00 08 .`.È.‹0 õà€.–...
0013b740 10 58 17 0c c4 e8 15 0c 14 58 17 0c 8c dd 15 0c .X..Äè...X..ŒÝ..
0013b750 0c 58 17 0c 14 62 2b 0c 2e db 15 0c 2e da 15 0c .X...b+..Û...Ú..
0013b760 be d9 15 0c 42 6c 81 93 41 57 3c 3c 92 64 d3 66 ¾Ù..Bl□"AW<<'dÓf
0013b770 3f d3 c3 65 0b 43 40 74 03 6e b2 60 01 60 01 88 ?ÓÃe.C@t.n²`.`.ˆ
0013b780 0a 8b b2 62 21 53 c0 33 06 8b e8 2e 02 89 e3 60 .‹²b!SÀ3.‹è..‰ã`
0013b790 f4 88 01 8b b1 bc d3 64 e3 60 26 4f f6 69 f6 6a ôˆ.‹±¼Ódã`&Oöiöj
0013b7a0 f6 6b f6 6c f6 6d 0b 00 f6 6e 78 af 09 00 e6 2f ökölöm..önx¯..æ/
0013b7b0 30 d3 d6 2f c6 2f b6 2f a6 2f 96 2f 86 2f 22 4f 0ÓÖ/Æ/¶/¦/–/†/"O
0013b7c0 32 6d 42 6b d4 5d d1 5c b7 3c 02 8f 42 52 01 a0 2mBkÔ]Ñ\·<.BR.
0013b7d0 b3 6a c3 6a 43 68 d3 69 15 4a 10 78 00 ee 22 1d ³jÃjChÓi.J.x.î".
0013b7e0 0a 8f 10 79 93 65 24 d2 83 64 2c e6 2c 79 0b 42 .□.y"e$Ò‚d,æ,y.B
0013b7f0 2c 78 01 7e a3 3e f5 8b b3 3c 02 8d e2 2d 01 a0 ,x.~£>õ‹³<.□â-.
0013b800 f4 e0 00 e0 26 4f f6 68 f6 69 f6 6a f6 6b f6 6c ôà.à&Oöhöiöjököl
0013b810 f6 6d 0b 00 f6 6e 17 d3 00 e6 e6 2f 53 6e 22 4f öm..ön.Ó.ææ/Sn"O
0013b820 32 65 16 d2 0b 42 54 55 03 65 58 25 03 8b 26 4f 2e.Ò.BTU.eX%.‹&O
0013b830 f8 e0 0b 00 f6 6e 2c e2 0e d3 ff 75 16 91 27 05 øà..ön,â.Óÿu.'.
0013b840 32 64 44 54 1a 05 10 74 5c 34 42 60 1c 30 02 2e 2dDT...t\4B`.0..
0013b850 41 53 31 1e 48 84 e8 80 00 e0 26 4f 0b 00 f6 6e AS1.H„è€.à&O..ön
0013b860 4c e0 4d 03 38 23 0b 89 0b 00 00 e0 96 00 ee ee Là M.8#.‰...à–.îî
0013b870 ee d8 15 0c 0c 58 17 0c 4c d7 15 0c be d9 15 0c îØ...X..L×..¾Ù..
0013b880 86 93 73 62 f5 e1 51 14 3c 32 62 14 00 e5 73 14 †"sbõáQ.<2b..ås.
0013b890 1c 42 40 e0 24 14 55 14 56 14 57 14 58 14 59 14 .B@à$.U.V.W.X.Y.
0013b8a0 5b 14 5d 14 5f 14 56 04 44 e0 56 04 48 e0 55 04 [.]._.V.DàV.HàU.
0013b8b0 4a e0 55 04 4c e0 01 e2 25 04 4e e0 35 d2 21 61 JàU.Là.â%.Nà5Ò!a
0013b8c0 15 04 50 e0 55 04 52 e0 55 04 43 60 0b 00 09 00 ..PàU.RàU.C`....
0013b8d0 e6 2f d6 2f 53 6d c6 2f d8 2d 22 4f 03 8f 00 e6 æ/Ö/SmÆ/Ø-"O.□.æ
0013b8e0 2d d3 32 6d d4 5d 2d d2 0b 42 d3 65 03 65 58 25 -Ó2mÔ]-Ò.BÓe.eX%
0013b8f0 03 8b 87 ba f8 e4 21 a0 09 00 53 67 26 d2 ff 77 .‹‡°øä! ..Sg&Òÿw
0013b900 47 9c 2c e3 22 66 37 07 66 5e 00 e4 11 a0 1a 07 Gœ,ã"f7.f^.ä. ..
0013b910 4c e0 ed 03 38 23 0a 8b 10 7d 7c 3d d2 66 d1 57 Làí.8#.‹.}|=ÒfÑW
0013b920 cc 36 9d bf e3 64 6d ba 00 e4 08 a0 e3 60 54 7e Ì6□¿ãdm°.ä. ã`T~
0013b930 01 74 62 63 33 34 eb 8b 64 ba f7 e4 00 e0 26 4f .tbc34ë‹d°÷ä.à&O
0013b940 f6 6c f6 6d 0b 00 f6 6e c2 af 09 00 e6 2f d6 2f ölöm..önÂ¯..æ/Ö/
0013b950 43 6d c6 2f 12 d3 22 4f 33 3d 03 89 52 ba e0 e4 CmÆ/.Ó"O3=.‰R°àä
0013b960 27 a0 09 00 0c d2 53 6c 18 4c 22 66 08 4c 00 e4 ' ...ÒSl.L"f.L.ä
0013b970 66 5e 19 a0 00 4c 4c e0 ed 02 28 22 12 8b 00 e5 f^. .LLàí.("..‹.å
0013b980 d3 66 c3 67 6c bf e3 64 3c ba 00 e4 12 a0 e3 60 ÓfÃgl¿ãd<°.ä. ã`
0013b990 ff 07 96 00 14 2c 17 0c 0c 58 17 0c be d9 15 0c ÿ.–..,...X..¾Ù..
0013b9a0 5e b0 00 00 54 7e 01 74 62 62 23 34 e3 8b 29 ba ^°..T~.tbb#4ã‹)°
0013b9b0 f7 e4 00 e0 26 4f f6 6c f6 6d 0b 00 f6 6e 3c d2 ÷ä.à&Oölöm..ön<Ò
Ready
Offset: 0x0013b9a0 (1.292

and after:



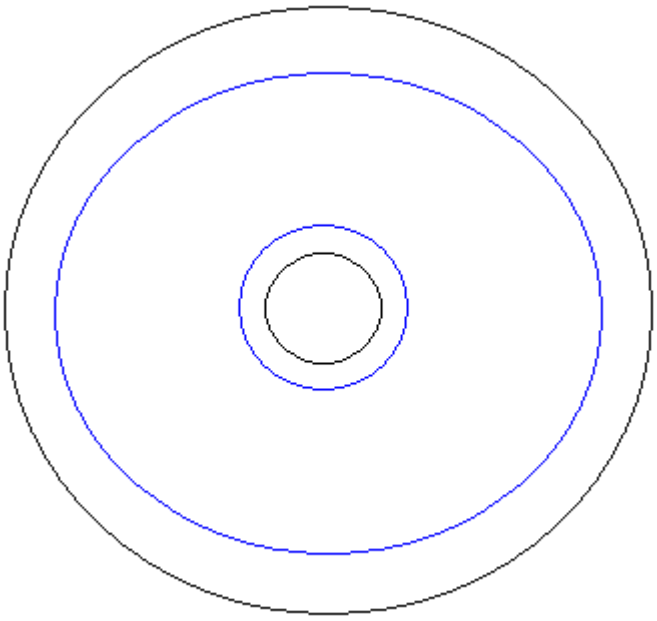
and we do the same for second entry. Save the file and you're hacking is done.

Note: There is a program called dahack.exe that I read that do this process we just made. You can try it. I didn't use it here because I've never tested it before, and hex editing is very easy anyway.

4. Creating a dummy file

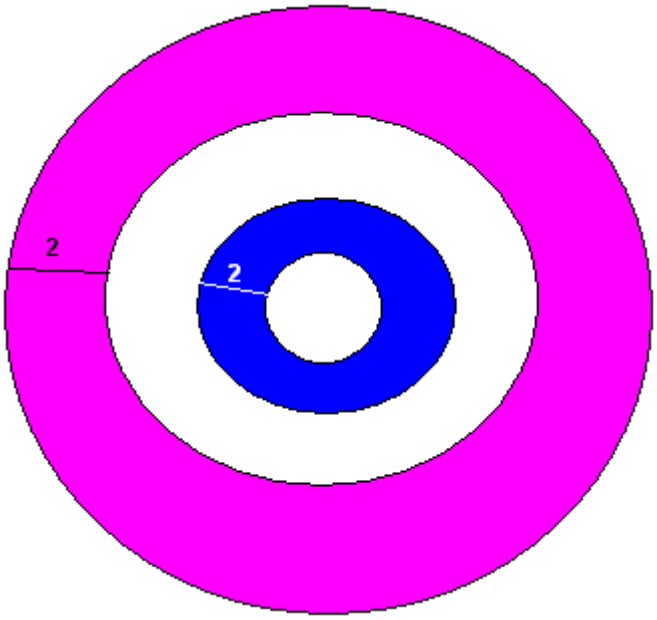
Before we create a dummy file, let's discuss why that's important.

Dreamcast reads discs in a constant angular speed. It means that the disc does a fixed N spins per minute. Another important point is that data has the same density around all the disc (I guess). Well, now imagine the disc turning, and the lens fixed on a position reading data. Let's imagine that data is distributed in 2 lines, marked in blue in this picture:



Inner line is smaller than the other. We considered the same data density, so bigger circle has more data than the other. Let's put some absurd numbers, just to help the understanding. Smaller line has 10 bytes, and bigger one has 100 bytes. Let's suppose that disc spins one time per second (another absurd), that's a constant angular speed. Well, if Dreamcast reads smaller circle, then it will read 10 bytes/second. If it reads bigger circle, then it will read 100 bytes/second. The conclusion is: Dreamcast reads faster the more far you're from center of the disc.

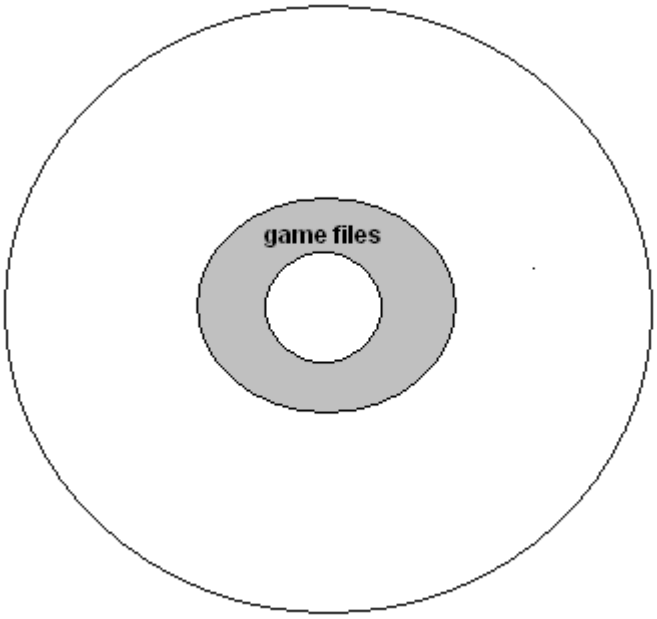
There is another advantage of having the files far of center. Let's imagine the area like in this picture:



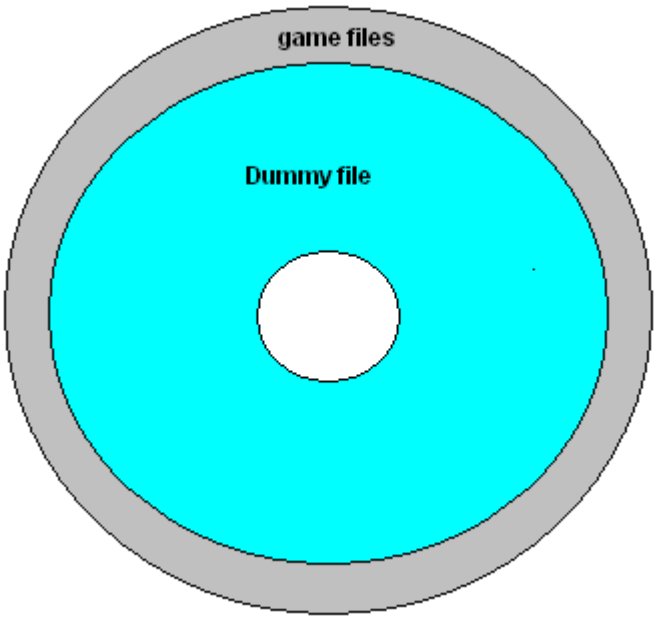
The measures marked are 2 something, let's consider centimeters. It's obvious that purple area is much bigger than blue area. So, considering the hypothesis of same data density, purple area has more data than blue area. For numbers, consider that blue area has 20mb and purple area has 100mb.

Okay, now imagine your reader. You know that noise it does? It's the reader moving far and near the disc center, to get in the right data "line". Well, if data is recorded on blue area, the reader can cover 20mb moving 2cm. For purple area, reader can cover 100mb moving 2cm. It means that if the game is recorded on the purple area, reader will actually move less, because more files will be available with the same move. Imagine a 100mb game, where it needs to load 2 files, one contained in inner part of recording, other in the distant part of the recording. If we record them in purple area, reader will read first file, move 2cm and then read the other. We can record in blue area, but since it only holds 20mb, recording will pass the 2cm, let's consider 4cm. So, for this case, reader will read first file, moves 4cm, and then reads the other. That's worse, because moving more means bigger loading times and more noise.

So, our objective is to record files far from center. We can't just make that, because recoding must start from inner part of disc. So what we do is create a dummy file and force it to be put in inner part of the disc, this way data will be "pushed" far. This picture before dummy file:



and after dummy file:



We'll analyse how to put it first later. Before that, we need to create the dummy file with an appropriate size. Well, obviously, the best results are obtained when dummy file + games files = all space available in disc. Well, selfboot session occupies some space of the disc, so how to find out the right size? Well, I've calculated that with a selfboot session, you have available in disc 709.505.843 bytes. Maybe it's not the exact number (now I can't remember how I calculated it), but it's very next to it. So you can use dummy.exe (included somewhere else in this pack) this way to create the dummy file:

```
dummy.exe [name of file] [size in bytes]
```

Note: Dummy file is a file full of bytes "00", so it's extreme compactable in .rar, .7z or whatever. It practically won't increase the image of your disc in a compacted format.

Well, let's go back to Crazy Taxi. Pressing right-button on the folder of Crazy Taxi we see that the folder has 101.483.508 bytes.

Note: When we record the game, IP.BIN doesn't need to be present as a file in game folder. So, if you don't want to have a copy of it, you can see the size of the files without IP.BIN. But don't erase it of the folder yet, we'll use it soon. In Crazy Taxi example I'll keep a copy in game files.

If our disc has 709.505.843 bytes available, we make 709.505.843 - 101.483.508 = 608.022.335 bytes, that's our dummy file size. So we use dummy.exe this way

```
dummy.exe DUMMY.DAT 608022335
```

and a huge file called DUMMY.DAT will be created. Of course you can name it whatever you want, it's not important. Now move it together with your game files. To see if size of files are okay, you can right click folder and it must have exactly 709.505.843 bytes.

Note: Use upper-case in filename. The file order list that we'll create soon is case sensitive, so pay attention and have all the files in upper-case or you can have a hard to discover problem in future.

5. Preparing file order

We need to create a file order list to a program that we'll use later. If we want that files are in this order (from inner part of the disc):

```
DUMMY.DAT
C.BIN
B.BIN
A.BIN
```

We'll have to create a .txt file containing this information:

```
data/DUMMY.DAT 4
data/C.BIN 3
data/B.BIN 2
data/A.BIN 1
```

Pay attention: 1) We use inverted bars (/), not regular ones (\); 2) List is case sensitive, pay attention on the example: "data" is lowercase, filename (as the names you have in your game files folder) is uppercase; 3) The program that will use this file needs everything perfect, you can make your list not to work correctly if you put a space after the number for example.

The numbers in front of the file are it's "weight". The higher the number is, closer to the center it will be. This number is only used to compare with other files weight, it's not an absolute value. For example, if instead of the last file list you used that one:

```
data/DUMMY.DAT 60000
data/C.BIN 1512
data/B.BIN 100
data/A.BIN 0
```

The result would be the same. Just be careful to respect the range limit (0-65535).

Only weight is respected when program uses the list, so you don't need to respect the order, but it's good so you don't get confused. For example:

```
data/B.BIN 2
data/DUMMY.DAT 4
data/C.BIN 3
data/A.BIN 1
```

would work exactly the same way, but if you want to change anything, it will be harder to see.

Last comment, there is no problem if you deleted some file and forgot it in file list, it won't make any bad. For example:

```
data/DUMMY.DAT 4
data/C.BIN 3
data/B.BIN 2
data/A.BIN 1
data/NOEXIST.BIN 10
data/LALALA.EXE 12
```

If "NOEXIST.BIN" and "LALALA.EXE" don't exist, result will be the same of the first file list shown.

As you could see, dummy file order is the most important, because it must come before everything else. But you should use file order of GD, because usually games are organized to increase performance. A simple example. Imagine a game where the first level is loaded. It's a 400mb game, and for first level it loads "AMBIENT1.BIN" and "ZTEXTURES1.AFS". If you don't respect original file order, the two files will be organized by alphabetic order and files will be far away from each other (one begins with A, other with Z). But if you use GD file order, probably these two files will be put together because it was optimized, so Dreamcast will load level faster because it needs to read first file, and the other one is very close, it doesn't need to move reader far away to read it. But how to use original file order?

Let's do it in Crazy Taxi, and you'll be able to do with every game. Well, using "Extracting GDI dump files" tutorial mentioned thousand of times, you must have a file list exported by IsoBuster. Here is the one I got for Crazy Taxi:

```
00548218 , >:\0GDTEX.PVR
00548283 , >:\1ST_READ.BIN
00497599 , >:\AICADRV.BIN
00526655 , >:\BINC1.AFS
00520773 , >:\BINC2.AFS
00512113 , >:\BINC3.AFS
00504386 , >:\COLDC1.BIN
00505952 , >:\COLDC2.BIN
00503496 , >:\COLDC3.BIN
00510321 , >:\LANDDC1.AFS
00511217 , >:\LANDDC2.AFS
00498484 , >:\MOTDC.BIN
00502810 , >:\OBJDC1.BIN
00502778 , >:\OBJDC2.BIN
00502746 , >:\OBJDC3.BIN
00499878 , >:\POLDC0.BIN
00502490 , >:\POLDC1.BIN
00502671 , >:\POLDC2.BIN
00502479 , >:\POLDC3.BIN
00503045 , >:\RECADVAC.BIN
00503098 , >:\RECADVDC.BIN
00502939 , >:\RECENDAC.BIN
```

```
00502992 , >:\RECENDDC.BIN
00497627 , >:\SNDDC0.BIN
00498016 , >:\SNDDC1.BIN
00498250 , >:\SNDDC2.BIN
00498341 , >:\SNDDC3.BIN
00498360 , >:\SNDDC5.BIN
00502305 , >:\SNDDC6.BIN
00502407 , >:\SNDDC7.BIN
00498395 , >:\SNDDC8.BIN
00532894 , >:\SONG01.AFS
00503151 , >:\SPLDC1.BIN
00503310 , >:\SPLDC2.BIN
00502842 , >:\SPLDC3.BIN
00502159 , >:\SPRADV.BIN
00502267 , >:\SPRBOX.BIN
00501780 , >:\SPRCENG.BIN
00501828 , >:\SPRCJAP.BIN
00501872 , >:\SPRCOMM.BIN
00509697 , >:\SPRGAME.BIN
00509590 , >:\SPRGENG.BIN
00509650 , >:\SPRGJAP.BIN
00509042 , >:\SPRMENG.BIN
00509518 , >:\SPRMENU.BIN
00509251 , >:\SPRMJAP.BIN
00501030 , >:\TEXDC0.BIN
00508476 , >:\TEXDC1.BIN
00507940 , >:\TEXDC2.BIN
00507549 , >:\TEXDC3.BIN
00513870 , >:\VOICE01.AFS
```

I made a script to handle this kind of file. It's the listorganizer.au3 (included on this pack). It will open a IsoBuster exported file list, sort it by LBA order (LBA is sector count I guess, so the higher the LBA, more far from center) and write the results that you'll see soon. I haven't included any kind of detection for invalid format, so have the list in this format before using it (8 numbers, space, comma, any string).

To convert the list, open the program, select the file list created with IsoBuster and "output.txt" will be created in the script folder (not in the source folder!). If your game has hundreds of files, the process may take a while because of sorting algorithm. For Crazy Taxi, that's the output:

```
>:\AICADRV.BIN 30000
>:\SNDDC0.BIN 29990
>:\SNDDC1.BIN 29980
>:\SNDDC2.BIN 29970
>:\SNDDC3.BIN 29960
>:\SNDDC5.BIN 29950
>:\SNDDC8.BIN 29940
>:\MOTDC.BIN 29930
>:\POLDC0.BIN 29920
>:\TEXDC0.BIN 29910
>:\SPRCENG.BIN 29900
>:\SPRCJAP.BIN 29890
>:\SPRCOMM.BIN 29880
>:\SPRADV.BIN 29870
>:\SPRBOX.BIN 29860
>:\SNDDC6.BIN 29850
>:\SNDDC7.BIN 29840
>:\POLDC3.BIN 29830
>:\POLDC1.BIN 29820
>:\POLDC2.BIN 29810
>:\OBJDC3.BIN 29800
```

```
 >:\OBJDC2.BIN 29790
 >:\OBJDC1.BIN 29780
 >:\SPLDC3.BIN 29770
 >:\RECENDAC.BIN 29760
 >:\RECENDDC.BIN 29750
 >:\RECADVAC.BIN 29740
 >:\RECADVDC.BIN 29730
 >:\SPLDC1.BIN 29720
 >:\SPLDC2.BIN 29710
 >:\COLDC3.BIN 29700
 >:\COLDC1.BIN 29690
 >:\COLDC2.BIN 29680
 >:\TEXDC3.BIN 29670
 >:\TEXDC2.BIN 29660
 >:\TEXDC1.BIN 29650
 >:\SPRMENG.BIN 29640
 >:\SPRMJAP.BIN 29630
 >:\SPRMENU.BIN 29620
 >:\SPRGENG.BIN 29610
 >:\SPRGJAP.BIN 29600
 >:\SPRGAME.BIN 29590
 >:\LANDDC1.AFS 29580
 >:\LANDDC2.AFS 29570
 >:\BINC3.AFS 29560
 >:\VOICE01.AFS 29550
 >:\BINC2.AFS 29540
 >:\BINC1.AFS 29530
 >:\SONG01.AFS 29520
 >:\0GDTEX.PVR 29510
 >:\1ST_READ.BIN 29500
```

It's not in the correct layout yet, we need to make some adjustments. A nice way to do it open the file in notepad and use "replace" tool to replace all " >:\" (without quotes!) for "data/". Again, note the space (standard file list can't have lines starting with a blank space"), the case and the inverted bar. So, doing it, that's the result:

```
data/AICADRV.BIN 30000
data/SNDDC0.BIN 29990
data/SNDDC1.BIN 29980
data/SNDDC2.BIN 29970
data/SNDDC3.BIN 29960
data/SNDDC5.BIN 29950
data/SNDDC8.BIN 29940
data/MOTDC.BIN 29930
data/POLDC0.BIN 29920
data/TEXDC0.BIN 29910
data/SPRCENG.BIN 29900
data/SPRCJAP.BIN 29890
data/SPRCOMM.BIN 29880
data/SPRADV.BIN 29870
data/SPRBOX.BIN 29860
data/SNDDC6.BIN 29850
data/SNDDC7.BIN 29840
data/POLDC3.BIN 29830
data/POLDC1.BIN 29820
data/POLDC2.BIN 29810
data/OBJDC3.BIN 29800
data/OBJDC2.BIN 29790
data/OBJDC1.BIN 29780
data/SPLDC3.BIN 29770
data/RECENDAC.BIN 29760
data/RECENDDC.BIN 29750
data/RECADVAC.BIN 29740
data/RECADVDC.BIN 29730
data/SPLDC1.BIN 29720
```

```
data/SPLDC2.BIN 29710
data/COLDC3.BIN 29700
data/COLDC1.BIN 29690
data/COLDC2.BIN 29680
data/TEXDC3.BIN 29670
data/TEXDC2.BIN 29660
data/TEXDC1.BIN 29650
data/SPRMENG.BIN 29640
data/SPRMJAP.BIN 29630
data/SPRMENU.BIN 29620
data/SPRGENG.BIN 29610
data/SPRGJAP.BIN 29600
data/SPRGAME.BIN 29590
data/LANDDC1.AFS 29580
data/LANDDC2.AFS 29570
data/BINC3.AFS 29560
data/VOICE01.AFS 29550
data/BINC2.AFS 29540
data/BINC1.AFS 29530
data/SONG01.AFS 29520
data/0GDTEX.PVR 29510
data/1ST_READ.BIN 29500
```

It's in the standard form now. If your game has subdirectories, there will be more "\" bars that must be converted to "/", use the replace tool again. In case of Crazy Taxi, we don't need to it.

It could be used now, but we're forgetting the extra files that we'll have in CD. One is IP.BIN. Again, IP.BIN is not necessary to be with files in CD, but I like to have a copy there. And, more important, we need to force dummy file to be in inner part of the disc. Remember, higher numbers = inner part of the disc. So we manually edit the file list assigning higher numbers for extra files. So we would add these 2 lines:

```
data/DUMMY.DAT 30002
data/IP.BIN 30001
```

So the final result will be:

```
data/DUMMY.DAT 30002
data/IP.BIN 30001
data/AICADRV.BIN 30000
data/SNDDC0.BIN 29990
data/SNDDC1.BIN 29980
data/SNDDC2.BIN 29970
data/SNDDC3.BIN 29960
data/SNDDC5.BIN 29950
data/SNDDC8.BIN 29940
data/MOTDC.BIN 29930
data/POLDC0.BIN 29920
data/TEXDC0.BIN 29910
data/SPRCENG.BIN 29900
data/SPRCJAP.BIN 29890
data/SPRCOMM.BIN 29880
data/SPRADV.BIN 29870
data/SPRBOX.BIN 29860
data/SNDDC6.BIN 29850
data/SNDDC7.BIN 29840
data/POLDC3.BIN 29830
data/POLDC1.BIN 29820
data/POLDC2.BIN 29810
data/OBJDC3.BIN 29800
```

```
data/OBJDC2.BIN 29790
data/OBJDC1.BIN 29780
data/SPLDC3.BIN 29770
data/RECENDAC.BIN 29760
data/RECENDDC.BIN 29750
data/RECADVAC.BIN 29740
data/RECADVDC.BIN 29730
data/SPLDC1.BIN 29720
data/SPLDC2.BIN 29710
data/COLDC3.BIN 29700
data/COLDC1.BIN 29690
data/COLDC2.BIN 29680
data/TEXDC3.BIN 29670
data/TEXDC2.BIN 29660
data/TEXDC1.BIN 29650
data/SPRMENG.BIN 29640
data/SPRMJAP.BIN 29630
data/SPRMENU.BIN 29620
data/SPRGENG.BIN 29610
data/SPRGJAP.BIN 29600
data/SPRGAME.BIN 29590
data/LANDDC1.AFS 29580
data/LANDDC2.AFS 29570
data/BINC3.AFS 29560
data/VOICE01.AFS 29550
data/BINC2.AFS 29540
data/BINC1.AFS 29530
data/SONG01.AFS 29520
data/0GDTEX.PVR 29510
data/1ST_READ.BIN 29500
```

Again, be careful about the case and don't let spaces after the number. After that, our file list is done.

6. Preparing recording process

If you don't have an ASPI driver installed, you need to install one and reboot windows. Please google for one.

Now copy selfboot directory on drive c, so you its contents must be on c:\selfboot\. of course you can put it wherever you want, but it probably won't work on "c:\im dumb so i like long directories name with a lot of space\and subdirectories too \selfboot".

Move all the game files (in our case, all files of Crazy Taxi) to c:\selfboot\data, the folder is already created there.

Now get a CD-RW. It's a good idea to use a CD-RW because you can make as many tests as you want and you can try your game on emulator before wasting CDs for the real Dreamcast.

Next step is to get the device number for cdrecord. Use the command:

```
cdrecord.exe -scanbus
```

and it will show your HD and CD/DVD drives. Find your drive there and take note of the numbers. In my case, it's 4,1,0, but it can be anything for you. Now these

numbers will be refered as a,b,c.

Note: If it shows the numbers, it means that cdrecord is probably working nice. If not, you'll have to figure out by yourself why things are not working. Did you really reboot Windows after ASPI installation? Is the version of your cdrecord adequate for you? You can try newer versions too. You can try this site too (http://www.dchelp.net/index.php?id=cdr), it has some tips to make cdrecord work if you're having trouble.

Now as first recording test, erase your CD-RW using this command:

```
cdrecord.exe -dev=a,b,c -blank=minimal -speed=x
```

-speed parameter is optional, default value is -speed=4 (it will erase disc in 4x if you don't put it). Don't forget that a,b,c must be replaced by the name you found with -(hyphen) on scanbus, that's the last time I say that. I hope everything is working fine for you now.

7. Recording game

Now we're at the last steps of making a selfboot copy of your game. Not only a working copy, but a great copy because we're dumming and using correct file order.

Except for a change on "mkisofs" step, everything else here is of the Echelon selfboot method. So if you have mastered their tutorial, everything is going to be very easy for you. If you haven't, don't worry, things will be explained step-by-step.

7.1. First session: audio track

Use this command:

```
cdrecord.exe -dev=a,b,c -multi -audio audio.raw -speed=x
```

Again, speed parameter is optional (default speed is 4). After it's done, you'll have the first session burned.

Now use this command:

```
cdrecord.exe -dev=a,b,c -msinfo
```

It will return some numbers. They'll probably be 0,11702. I think there is a chance that you get 0,11700 too, or maybe other number. So, if you get something different of 0,11702, just replace them with your number of next steps.

7.2. Last hacking process

Now you don't have to hex edit yourself, Echelon made some nice tools that will make the work for you. All game files are in c:\selfboot\data right? Move 1ST_READ.BIN (that's the case of Crazy Taxi and most games, but remember that

the file you must use is the one that was indicated when we hacked IP.BIN in section 3) and IP.BIN to c:\selfboot (or copy IP.BIN if you want to have a copy in game files, that's what I'm going to do). Now open "binhack.exe" and you must use these informations:

```
Enter name of binary: 1ST_READ.BIN
Enter name of bootsector: IP.BIN
Enter msinfo value: 11702
```

Again, maybe your game first binary is called different than 1ST_READ.BIN, use the right name. And if you got another value than 11702 replace it.

After this, move 1ST_READ.BIN (or whatever) to c:\selfboot\data again.

7.3. Creating Image

Put your file list (already modified to standard format like told in section 5) in c:\selfboot and name it with a short name, for example, list.txt. Now use the following command:

```
mkisofs -C 0,11702 -V TITLE -sort list.txt -l -o data.iso data
```

Maybe you need to change 0,11702 for the value you got. Change TITLE for the title of CD you want (use short names, no space).

For Crazy Taxi, my file list had the name of ctaxi.txt, so I used this line:

```
mkisofs -C 0,11702 -V MY_CRAZYTAXI -sort ctaxi.txt -l -o data.iso data
```

It may take a while to finish the process. After this, you'll have a file called data.iso in c:\selfboot.

7.4. Checking File Order

Now we must check if file order is the same of file list. Use this command:

```
isofix.exe data.iso 11702
```

After the process, you'll have a file called "fixed.iso" on c:\selfboot. Don't delete "data.iso" yet, we'll use it soon.

Now open "fixed.iso" in IsoBuster, and file will appear on right window. There are some details about each file, and one of them is LBA number. The lowest the LBA, closer to the center are the file. So order files by LBA (click LBA column title), and check if file order is okay, just looking and comparing with file list. If your game has subdirectories, ignore LBA of directory and look at LBA number of files inside each directory.

For Crazy Taxi, remember that the file list used is this one:

```
data/DUMMY.DAT 30002
data/IP.BIN 30001
data/AICADRV.BIN 30000
data/SNDDC0.BIN 29990
data/SNDDC1.BIN 29980
data/SNDDC2.BIN 29970
data/SNDDC3.BIN 29960
data/SNDDC5.BIN 29950
data/SNDDC8.BIN 29940
data/MOTDC.BIN 29930
data/POLDC0.BIN 29920
data/TEXDC0.BIN 29910
data/SPRCENG.BIN 29900
data/SPRCJAP.BIN 29890
data/SPRCOMM.BIN 29880
data/SPRADV.BIN 29870
data/SPRBOX.BIN 29860
data/SNDDC6.BIN 29850
data/SNDDC7.BIN 29840
data/POLDC3.BIN 29830
data/POLDC1.BIN 29820
data/POLDC2.BIN 29810
data/OBJDC3.BIN 29800
data/OBJDC2.BIN 29790
data/OBJDC1.BIN 29780
data/SPLDC3.BIN 29770
data/RECENDAC.BIN 29760
data/RECENDDC.BIN 29750
data/RECADVAC.BIN 29740
data/RECADVDC.BIN 29730
data/SPLDC1.BIN 29720
data/SPLDC2.BIN 29710
data/COLDC3.BIN 29700
data/COLDC1.BIN 29690
data/COLDC2.BIN 29680
data/TEXDC3.BIN 29670
data/TEXDC2.BIN 29660
data/TEXDC1.BIN 29650
data/SPRMENG.BIN 29640
data/SPRMJAP.BIN 29630
data/SPRMENU.BIN 29620
data/SPRGENG.BIN 29610
data/SPRGJAP.BIN 29600
data/SPRGAME.BIN 29590
data/LANDDC1.AFS 29580
data/LANDDC2.AFS 29570
data/BINC3.AFS 29560
data/VOICE01.AFS 29550
data/BINC2.AFS 29540
data/BINC1.AFS 29530
data/SONG01.AFS 29520
data/0GDTEX.PVR 29510
data/1ST_READ.BIN 29500
```

Remember that it means, DUMMY.DAT is the closest file to the center, then IP.BIN, etc. In IsoBuster, ordering by LBA I got this result:

DUMMY.DAT	11727	579,86 MB	608.022.335	25/2/2009 18:55:03
IP.BIN	308613	32,00 KB	32.768	25/2/2009 08:14:15
AICADRV.BIN	308629	55,88 KB	57.216	13/11/1999 17:49:29
SNDDC0.BIN	308657	777,21 KB	795.860	13/11/1999 17:49:29
SNDDC1.BIN	309046	466,47 KB	477.668	13/11/1999 17:49:29
SNDDC2.BIN	309280	180,64 KB	184.972	13/11/1999 17:49:29
SNDDC3.BIN	309371	36,73 KB	37.608	13/11/1999 17:49:29
SNDDC5.BIN	309390	68,50 KB	70.148	13/11/1999 17:49:29
SNDDC8.BIN	309425	177,46 KB	181.724	13/11/1999 17:49:29
MOTDC.BIN	309514	2,72 MB	2.854.020	13/11/1999 17:49:29
POLDC0.BIN	310908	2,25 MB	2.358.960	13/11/1999 17:49:29
TEXDC0.BIN	312060	1,46 MB	1.534.368	13/11/1999 17:49:29
SPRCENG.BIN	312810	96,00 KB	98.304	13/11/1999 17:49:29
SPRCJAP.BIN	312858	88,00 KB	90.112	13/11/1999 17:49:29
SPRCOMM.BIN	312902	572,94 KB	586.688	13/11/1999 17:49:29
SPRADV.BIN	313189	215,50 KB	220.672	13/11/1999 17:49:29
SPRBOX.BIN	313297	75,38 KB	77.184	13/11/1999 17:49:29
SNDDC6.BIN	313335	203,82 KB	208.716	13/11/1999 17:49:29
SNDDC7.BIN	313437	142,16 KB	145.576	13/11/1999 17:49:29
POLDC3.BIN	313509	21,63 KB	22.152	13/11/1999 17:49:29
POLDC1.BIN	313520	361,15 KB	369.816	13/11/1999 17:49:29
POLDC2.BIN	313701	148,09 KB	151.640	13/11/1999 17:49:29
OBJDC3.BIN	313776	64,00 KB	65.536	13/11/1999 17:49:29
OBJDC2.BIN	313808	64,00 KB	65.536	13/11/1999 17:49:29
OBJDC1.BIN	313840	64,00 KB	65.536	13/11/1999 17:49:29
SPLDC3.BIN	313872	193,60 KB	198.244	13/11/1999 17:49:29
RECENDAC.BIN	313969	105,47 KB	108.000	13/11/1999 17:49:29
RECENDDC.BIN	314022	105,47 KB	108.000	13/11/1999 17:49:29
RECADVAC.BIN	314075	105,47 KB	108.000	13/11/1999 17:49:29
RECADVDC.BIN	314128	105,47 KB	108.000	13/11/1999 17:49:29
SPLDC1.BIN	314181	316,17 KB	323.756	13/11/1999 17:49:29
SPLDC2.BIN	314340	370,16 KB	379.044	13/11/1999 17:49:29
COLDC3.BIN	314526	1,74 MB	1.821.220	13/11/1999 17:49:29
COLDC1.BIN	315416	3,06 MB	3.205.672	13/11/1999 17:49:29
COLDC2.BIN	316982	3,12 MB	3.268.952	13/11/1999 17:49:29
TEXDC3.BIN	318579	781,66 KB	800.416	13/11/1999 17:49:29
TEXDC2.BIN	318970	1,05 MB	1.096.512	13/11/1999 17:49:29
TEXDC1.BIN	319506	1,10 MB	1.158.016	13/11/1999 17:49:29
SPRMENG.BIN	320072	416,50 KB	426.496	13/11/1999 17:49:29
SPRMJAP.BIN	320281	533,00 KB	545.792	13/11/1999 17:49:29
SPRMENU.BIN	320548	142,25 KB	145.664	13/11/1999 17:49:29
SPRGENG.BIN	320620	120,00 KB	122.880	13/11/1999 17:49:29
SPRGJAP.BIN	320680	93,00 KB	95.232	13/11/1999 17:49:29
SPRGAME.BIN	320727	1,22 MB	1.277.056	13/11/1999 17:49:29
LANDDC1.AFS	321351	1,25 MB	1.312.768	25/2/2009 01:36:51
LANDDC2.AFS	321992	1,25 MB	1.312.768	25/2/2009 01:36:51
BINC3.AFS	322633	2,93 MB	3.076.096	25/2/2009 01:36:51
VOICE01.AFS	324135	12,99 MB	13.619.200	25/2/2009 01:36:53
BINC2.AFS	330785	10,99 MB	11.526.144	25/2/2009 01:36:51
BINC1.AFS	336413	11,69 MB	12.257.280	25/2/2009 01:36:51
SONG01.AFS	342398	29,43 MB	30.861.312	25/2/2009 01:36:52
1ST_READ.BIN	357467	1,40 MB	1.468.208	26/2/2009 19:51:09

DUMMY.DAT has the lowest LBA, so it's closest to the center, then IP.BIN, (...), then SONG01.AFS, and in the end, 1ST_READ.BIN. So our file list were made correctly, and ISO has the correct file order.

7.5. Burning last session: game files

Now execute "ipins.exe" and use the following data:

```
Enter name of bootsector: IP.BIN
Enter name of ISO: data.iso
```

And boot sector will be included in data.iso. Now it's time for burning. Use the following command:

```
cdrecord.exe -dev=a,b,c -xa1 data.iso -speed=x
```

Again, -speed=x is optional. Well, now it will burn the game files, and as we dummied the game for full CD, it will take around 20 minutes in 4x.

CDRecord is very easy to crash, any program that makes access to CD drive will make cdrecord exists with error. For example: IsoBuster, another CD recording program, and even some programs with no standard file managers. So I recommend that you go AFK and let your computer record the disc.

If at the start of recording you get this warning:

```
WARNING: Files may not fit on the disc.
```

or something like that, cancel recording start using CTRL+C. It means that you haven't dimensioned dummy file right, and program will make overburn, what is never good. Try making right size dummy, remaking image, etc, like told before in this guide.

8. Testing the disc

Game should now be bootable on Dreamcast. But you recorded on a CD-RW remember? So only your computer can read it. Test it on emulators. For this, make a image with Disc Juggler and test it on NullDC or test directly on CD with Chankast. You don't need to play the game, just see if it boots fine. If it does, you have done everything right.

If you have done Crazy Taxi like the example, it will work in both emulators. Remember to check before if the emulator you're using to test is compatible with the game you're trying to make selfbootable. An easy test for that is try the GDI you started with on NullDC, if it works, then selfbootable version must work too.

9. Distribution

If you want to distribute your rip, it's a great thing. But remember that these things

spread fast. If you distribute a good thing, you'll make a lot of people happy. If you distribute a bad thing, you'll make a lot of people confused. So be careful, and think if your rip is really worthy of distribution.

Don't forget to write a readme file, including important informations like what was ripped, downsampled, etc. In case of Crazy Taxi, nothing, so write that. Don't forget to tell your sources too, like GD dump, a group's release, etc. This will help to make scene clean of bad stuff.

Most common format for Dreamcast games is .cdi (Disc Juggler), so use it to make your image for distribution. Alcohol 120% format is acceptable, but not encouraged. And for God's sake, don't make images with Nero, newer versions can't burn Dreamcast selfboot games right.

Last step for distribution is to check files. For that, make a image of your CD-R/W (again, I recommend DiscJuggler). Now use Daemon Tools to mount the image on a virtual drive. Opening the drive, you'll see the game files you have recorded. Now you must have RapidCRC installed with standard configurations (so it should be included on right-click menu). Now, on virtual drive, select all the files (except dummy file, you don't need to check it), right-click and select "Open with RapidCRC". It will take a while to analyse every file CRC. Now use Create SFV file -> Create one .SFV file for all files and save it on c:\selfboot\data (you should still have the files you used there). Now open the .sfv you just created double-clicking it (it must be associated with RapidCRC or other SFV checker) and it will check the files on c:\selfboot\data. If check is alright, it means that files on image and files that you used on recording process are identical, so nothing is corrupted on your image.

And last comment, don't forget to try your game on real Dreamcast. I had a game once that booted fine on NullDC but froze on main screen on real Dreamcast.

10. Thanks

- (http://www.megagames.com/console/cgi-bin/dl.cgi?id=dc&file=dcis-tut.001!rar) Ripping guide of dcisos.de (site dead), it helped hacking my first dreamcast game.
- (http://www.dreamwell.pwp.blueyonder.co.uk/dcselfboot/Tutorial_Index/tutorial_index.htm) This site that has tons of tutorials that helped me doing stuff on DC.
- The nice guys who created the programs used here.
- All people that rip and distribute GDI.