

[iNES Header/Format Information File]
[Version 2.2 Compiled on 1/4/99]
[VmprHntrD - New Nick & Email address withheld]

NOTE: This document is on the iNES/FWNES98 format and is almost totally done. I still need to add some little tidbits as the iNES format is updated. Please do not modify this document in any way without MY prior consent. If you wish to add to this please talk to me through e-mail, or in the evenings on the IRC - EFnet, in the channel #emu. [I'm an OP in there.] I use a new nick now, and don't wish to give it out, ask for me under my old nick I may respond, or someone there will point me out if I'm away. Sorry but I just don't like to do some of what I used to and what got tied to that old name. PLEASE IF YOU DO FIGURE IT OUT DON'T POST IT ON WEBPAGES, ETC. USE THE OLD NAME AT THE TOP OF THIS DOCUMENT, THANK YOU. There's a hint though in this text somewhere to who I now am, so if you like play my guessing game and figure it out. =)

<Personal thanks goes to:>

- * Conjurer which for some reason with a related request inspired me to not to discontinue this list with v2.1.
- * Zophar for help with some math and errors.
- * FanWen for the greater extension of the iNES Extended Header format and all the associated work along with it, and especially for NES Emulator called FWNES98. It's a next to perfect NES Emulator, keep up the great work!
- * Groups like Damaged Cybernetics, GitM, NPS, Vertigo 2099 and more that have in one way or the other brought us the games this guide pertains to.
- * Anyone who has ever e-mailed and/or talked to me in IRC that assisted in the fixes and upkeep of this guide.
- * And thanks to all the loyal readers that actually care about this document and what I have to say here.

[Table of Contents]

Section 1 .Known HEXidecimal Values of the First 8 Bytes of the NES/FDS Header
Section 2 The Seventh and Eighth Bytes' HEXidecimal Values
Section 3Those Many Mysterious Mappers Now Supported
Section 4 HEXidecimal Values of the PRG/CHR
Section 5 Miscellaneous Information on the NES Format
Section 6 Let's Get Cracking on Prepping Your RAW/FFE/PasoFami Image
Section 7 Information (Recommended Tools for Converting iNES Images)
Section 8 UPDATES (The History of the iNES Header/Format Documentation)

[Sect.1 - Known HEXidecimal Values of the First 8 Bytes of the NES/FDS Header]

...Those FIRST (8) bytes of your NES, or to be a NES file...

(Bytes 0->3 MUST BE What is stated below.)

0 = 4E (N)
1 = 45 (E)
2 = 53 (S)
3 = 1A (Character Break, necessary!)

(Bytes 4->8-15 Can BE MODIFIED!)

4 = PRG (Hex number depends on size of PRG file)-/-- (See Sect. 4)
5 = CHR (Hex number depends on size of CHR file)/
6 = Mapper 0-15/ V or H Mirroring, Battery, 4 Screen VRAM, Trainer Switches
Also is used as the "Ones" place holder digit for the extended iNES Header format. (See Sect. 2)

7 = Mappers 16-?, the extended iNES Header format "Tens" place holder digit.
Again (See Sect. 2 for details on how this is set up.)

8->15 = Not used at this time but MUST BE ALL ZEROS or games will not work.

NOTE: Now that there is a fully established Extended iNES Header Format set up gracefully by FanWen and Marat Fayzullin ALWAYS leave in games that do not use over mapper 15 blank in bytes 8->15 or the games WILL NOT WORK!

* FWNES FDS Support NOTE *

As of FWNES98 v.300+ there has been Famicom Disk System support. To make sure that it works right for bytes 6->15 of the game header leave it all 00's or it will NOT WORK. All that is needed is a Famicom Disk System BIOS file, which is NOT included, and I can't give you it either.

(The .FDS Header Contents)

0 = 46 (F)
1 = 44 (D)
2 = 53 (S)
3 = 1A (Character Break, necessary!)

(Byte 4) - Number of Sides to FDS Disk(s)

4 = 04 (4 Sided disk) The 04 changes per amount of disk(s) side(s).

Above the example being used is Super Mario Bros. 2 (Known as Super Mario Bros. the Lost Levels to Americans in the SNES game Super Mario Allstars)

(For Easy Conversion purposes, one side of a FDS disk is 65500 bytes of data!)
The correct file sizes for FWNES's .FDS support then is as follows:

1 sided game = 16+65500 = 65516
2 sided game = 16+65500 X 2 = 131016
4 sided game = 16+65500 X 4 = 262016

* If you wish you can convert .DKA(Pasofami), .FAM(Famicom), and .FF3(Far Front East) files to the .FDS(FWNES) format with a tool included in it named 2FDS.EXE.

Use it, but use it wisely as the real file will give life, the other will take it away. (Indiana Jones and the Last Crusade plug/pun intended.) =)

(Bytes 5->8-15 CAN NOT BE MODIFIED!) <-- Reserved leave as 00's

[Sect. 2: - The Seventh and Eighth Bytes' HEXidecimal Values]_____

Byte 6 in a hexeditor will look something like this:

59 <---This is for CastleVania III: Dracula's Curse

(The "5" is the mapper number, the "9" stands for Vertical Mirroring AND 4 Screen VRAM ON!) -- See below for all the possible values!

Byte 7 in a hexeditor will look a little different, something like this:

* 1->F/0<---Only the first part is used in correlation with the first part of byte 6 above. *READ BELOW FOR EXAMPLE!*

Famicom Disk System Format Design

This has got to be the easiest setup yet for a Nintendo/Famicom file yet. In

this setup contributed by Fanwen for FWNES98 v.300+ there are simple single files with the extension of .FDS, and unlike the Pasofami with its .DKA, etc system the files are all in one. The header for these is quite simple as well which just consists bytes 0-3 reading (FDS Character Break) and the rest as 00's the emulator itself will handle the rest with no more information needed. Also unlike Marat's format as you are probably thinking about since it's on Mapper 20, Fanwen's has NO mapper, less fuss, no muss.

Extended iNES Header Format Design

Basically in this extended design the 7th byte is used to implement mappers of 16 and higher designed by FanWen, thought Marat likes to think he made it up. It does this through a hexadecimal conversion of the actual mapper number. In my example I'll use Mapper 69 with a vertical switch(1).

Example: Mapper 69 = 4E in hexadecimal...

Therefore in the new iNES Extended Header format this is how bytes 6 and 7 would look in hex and work together.

Example:

Bytes=>	0	1	2	3	4	5	6	7
	N	E	S	^Z	PRG	CHR	MAP/SW	MAP/0(always)
	4E	45	53	1A	10	20	E1	40

* See above that the ONES DIGIT (E) is in the 6th byte, and the TENS DIGIT (4) is the first half of the 7th byte. But be advised that the second half of the 7th byte in hex HAS TO ALWAYS BE 0 (zero)!

AND NOW FOR THE BAD NEWS FOR "YOU"

Considering iNES has set the standard for "Mapper Compatibility" we are faced with a new problem ...CONVERSION. We must convert the current games to the new format which is very TIME CONSUMING because of the iNES Header Editors out there that insert extra garbage into the header. Therefore let's get to FIXING YOUR GAMES for the new "Extended iNES Header Format":

- * First get out your hexeditor and move over to the eighth byte. For example lets use the NESImage(an old NES header editing tool) tag of: "DiskDude!"
- * On the 7th byte it will be covered by the "D", to make your game a "clean" image you must on the hexadecimal side put a double zero "00" through bytes 7-15, then save it.
- * Be very careful doing so, if you hit the first byte of the game, its DEAD if you aren't paying attention and save it...you've been warned.
- * If you can find a program on the internet called "NESFIX.EXE" this will do all the tasks above in a fraction of the time of doing it manually. Ask around a few people will be able to get it to you.
- * NOW your games will work in FWNES98 and iNES v0.7+

[Below are the current possibilities for the second figure of the 6th byte.]

This part of the byte is the place you setup the game to use, or not use Horizontal or Vertical mirroring, Battery Backup, 4 Screen VRAM, and a built in Trainer. Below are the different combinations of all the listed options in HEXidecimal format.

- 0 = H (Horizontal Mirroring ONLY)
- 1 = V (Vertical Mirroring ONLY)
- 2 = H + Bat. (Horizontal Mirroring + Battery ON)
- 3 = V + Bat. (Vertical Mirroring + Battery ON)
- 4 = H + Train. (Horizontal Mirroring + Trainer ON)
- 5 = V + Train. (Vertical Mirroring + Trainer ON)

6 = H + Bat. + Train. (Horizontal Mirroring + Battery and Trainer ON)
 7 = V + Bat. + Train. (Vertical Mirroring + Battery and Trainer ON)
 8 = H + 4scr. (Horizontal Mirroring + 4 Screen VRAM ON)
 9 = V + 4scr. (Vertical Mirroring + 4 Screen VRAM ON)
 A = H + Bat. + 4scr. (Horizontal Mirroring + Battery and 4 Screen VRAM ON)
 B = V + Bat. + 4scr. (Vertical Mirroring + Battery and 4 Screen VRAM ON)
 C = H + 4scr. + Train. (Horizontal Mirroring + 4 Screen VRAM and Trainer ON)
 D = V + 4scr. + Train. (Vertical Mirroring + 4 Screen VRAM and Trainer ON)
 E = H + Bat. + 4scr. + Train. (Horizontal Mirroring + Battery, 4 Screen VRAM, and Trainer ON)
 F = V + Bat. + 4scr. + Train. (Vertical Mirroring + Battery, 4 Screen VRAM, and Trainer ON)

[Sect. 3: - Those Many Mysterious Mappers Now Supported]_____

Below is the current possibilities for the all the supported mappers.

NOTE: The format for this graph is taken from the FWNES98 doc to give you a better example of what each mapper is and runs... I would personally like to thank Marat Fayzullin, Yang FanWen, YØSHi, and anybody else I left out that assisted in the adding of these new MAPPERS!

NOTE: As of FWNES98 v.300+ there was full Famicom Disk System support added, #2 which is for the games, extra sound, microphone, etc. Anyways FWNES98 uses the extension .FDS for these games and does NOT use Mapper 20. It uses a standard header with 00's through the mapper, the program itself will handle the FDS emulation.

* 74 * Mappers now supported thanks to FanWen and FWNES98 as of v.302!

Mapper Name	Examples (Game Names, etc.)
0 No Mapper	All 32kb ROM + 8kb VROM games like Super Mario Bros, Balloon Fight, Slalom, etc.
1 Nintendo MMC1 Chipset	1942, Bomberman II, Mega Man II, etc.
2 ROM (PRG) Switch	Konami games like CastleVania and Stinger.
3 VROM (CHR) Switch	Twin Bee, Q*Bert, and many japanese games.
4 Nintendo MMC3 Chipset	Super Mario Bros 2 & 3, Silver Surfer, etc.
5 Nintendo MMC5 Chipset	CastleVania III: Dracula's Curse
6 FFE F4XXX Games	F4XXX Games off the FFE CD-ROM
7 32kb ROM (PRG) Switch	Wizards & Warriors, Marble Madness, etc.
8 FFE F3XXX Games	F3XXX Games off the FFE CD-ROM
9 Nintendo MMC2 Chipset	Mike Tyson's Punch-Out!!, and Punch-Out!!
10 Nintendo MMC4 Chipset	Japanese Punch-Out!!, and japanese games.
11 Color Dreams Chipset	Crystal Mines, Tagin' Dragon, Babyboom, etc.
12 FFE F6XXX Games	F6XXX Games off the FFE CD-ROM
15 100-in-1 Cart Switch	100-in-1 Multicart Switch

Below are all the Extended iNES Header Format Mappers which is included in the FWNES98 NES Emulator which supports way beyond iNES v0.7+.

* Read above in Section 2 on how to implement Mappers 16 and up in FWNES/iNES.

16 Ban Dai Chipset	Japanese: Dragon Ball Z, Gaiden etc.
17 FFE F8XXX Games	F8XXX Games off the FFE CD-ROM
18 Jaleco SS8806 Chipset	Japanese: Jaleco's Sport Series
19 Namcot 106 Chipset	Japanese: Famista 90-92, Megami Teisen 1-2
20-> Famicom Disk System(iNES)	Reserved for the Disk System, DON'T USE IT!
21 Konami VRC4 2a Chipset	Japanese: Wai Wai World2, Goemon Gaiden 2
22 Konami VRC4 1b Chipset	Japanese: Twinbee 3, etc.
23 Konami VRC2b Chipset	Japanese: Wai Wai World, Kid Dracula, etc.

24	Konami VRC6 Chipset	Japanese: CastleVania 3j, MADAKAJ, etc.
25	Konami VRC4 Chipset	Japanese: Konami Baseball, Gradus 2, etc.
26	Konami VRC6V Chipset	Japanese: Madara and Esper Dream.
32	Irem G-101 Chipset	Japanese: Image Fighter 2, etc.
33	Taito TC0190/TC0350	Japanese: PonDoKoPon, Bubble Bobble 2j, etc.
34	Nina 1 (PRG) Switch	Japanese: Impossible Mission, Deadly Towers
48	Taito TC1090/RC0350	Japanese: Flintstones
64	Tengen Rambo-1	Klax, Shinobi, Skull and Crossbones
65	Irem H3001 Chipset	Japanese: Mr. Love, Mr. Big Hammer, P-Man2
66	74161/32 Chipset	Japanese: Dragonball 1, BioWarrior DAN
67	Sunsoft Mapper 3	Japanese: Fantasy Zone 2J
68	Sunsoft Mapper 4	Japanese: Afterburner 2
69	Sunsoft Mapper 5	Japanese: Batman Return of the Joker J
70	74161/32 Chipset	Japanese: Kamen Rider Club, Saint Seiya, etc
71	Camerica Mapper	Firehawk
72	Jaleco Early Mapper #0	Japanese: Jaleco TennisJ, Pinball Quest J
73	Konami VRC3	Japanese: Salamander
74	Taiwan MMC3 -Variant	Japanese: Kid Niki 3J(Hacked)
75	Jaleco SS805/Konami VRC1	Japanese: Jaleco BaseballJ, Ganbare Goemon1J
76	Namco 109 Chipset	Japanese: Megami TenshiJ
77	Irem Early Mapper #0	Japanese: Napoleon Senki, Holy Diver
78	74161/32 Chipset	Cosmos Carrier
79	Amer. Video Entertainment	F-15 City War, Krazy Kreatures, Trolls
80	Taito X-005 Chipset	Japanese: Demon Sword
81	Taito C075 Chipset	Japanese: Unknown
82	Taito X1-17 Chipset	Japanese: Unknown
83	Cony Mapper	Japanese: Garbo Dentsu
84	PasoFami Mapper!	Japanese: (.500 .MOD) Supported
85	Konami VRC7	Japanese: Tiny Toon Adv2J, Lagrange Point
86	Jaleco Early Mapper #2	Japanese: More Pro Baseball
87	74161/32 Chipset	Japanese: Hyper Sport MT
88	Namco 118 Chipset	Japanese: Devil Man J
89	SunSoft Early Mapper	Japanese: MitoKuomon
90	HK-TK2(Pirate Cart Switch)	Japanese: Samurai Spirits 2, MK3, Tekken 2
91	KH-SF3(Pirate Cart Switch)	Japanese: Street Fighter II Pro
92	Jaleco Early Mapper #1	Japanese: Moe Pro Soccer, Moe Pro Baseball88
93	74161/32	Japanese: Fantasy Zone
94	74161/32	Japanese: Senjono Okami
95	Namco 1??	Japanese: Dragon Buster
96	74161/32	Japanese: Anpanman no Hiragana Daisuki
97	Irem 74161/32	Japanese: Kaiketsu Yanchamaru, Crazy Climber
100	MMC3/Trainer/Buggy Mode	USED FOR HACKED ROMS!!
112	PC-Asder	Haung-Di
113	PC-Sachen/Hacker	Majong Champion, Metal Fighter, etc
114	PC-Supergames	Lion King
115	PC-CartSaint	Yuuyuu
116	PC-Reserved	AV Beauty Fighting
117	PC-Future	Sangokushi 4 (clone of "Warrior of Fate")
118	?????	Alumagiko J
119	TQROM	High Speed, Pinbot
122	74161/32	Madoola No Tsubasa
225	58/64-in-1 Cart Switch	58/64-in-1 Multicart Switch
226	72-in-1 Cart Switch	72-in-1 Multicart Switch
227	1200-in-1 Cart Switch	1200-in-1 Multicart Switch
228	Action 54 Cart Switch	Action 54 Multicart Switch

[Sect. 4: - HEXidecimal Values of the PRG/CHR]_____

[PRG - (Range 1 x 16kb pages -> 64 x 16kb pages)]

-There is a BARE MINIMUM required for PRG which is 1 x 16kb pages!

* NOTE: To figure out the exact size in bytes each of these pages are worth just start at 1 x 16kb pages (aka: 16384 bytes) and just keep adding 16384 more for each "page" higher.

(1 x 16kb pages) = 01	(2 x 16kb pages) = 02	(3 x 16kb pages) = 03
(4 x 16kb pages) = 04	(5 x 16kb pages) = 05	(6 x 16kb pages) = 06
(7 x 16kb pages) = 07	(8 x 16kb pages) = 08	(9 x 16kb pages) = 09
(10 x 16kb pages) = 0A	(11 x 16kb pages) = 0B	(12 x 16kb pages) = 0C
(13 x 16kb pages) = 0D	(14 x 16kb pages) = 0E	(15 x 16kb pages) = 0F
(16 x 16kb pages) = 10	(17 x 16kb pages) = 11	(18 x 16kb pages) = 12
(19 x 16kb pages) = 13	(20 x 16kb pages) = 14	(21 x 16kb pages) = 15
(22 x 16kb pages) = 16	(23 x 16kb pages) = 17	(24 x 16kb pages) = 18
(25 x 16kb pages) = 19	(26 x 16kb pages) = 1A	(27 x 16kb pages) = 1B
(28 x 16kb pages) = 1C	(29 x 16kb pages) = 1D	(30 x 16kb pages) = 1E
(31 x 16kb pages) = 1F	(32 x 16kb pages) = 20	(33 x 16kb pages) = 21
(34 x 16kb pages) = 22	(35 x 16kb pages) = 23	(36 x 16kb pages) = 24
(37 x 16kb pages) = 25	(38 x 16kb pages) = 26	(39 x 16kb pages) = 27
(40 x 16kb pages) = 28	(41 x 16kb pages) = 29	(42 x 16kb pages) = 2A
(43 x 16kb pages) = 2B	(44 x 16kb pages) = 2C	(45 x 16kb pages) = 2D
(46 x 16kb pages) = 2E	(47 x 16kb pages) = 2F	(48 x 16kb pages) = 30
(49 x 16kb pages) = 31	(50 x 16kb pages) = 32	(51 x 16kb pages) = 33
(52 x 16kb pages) = 34	(53 x 16kb pages) = 35	(54 x 16kb pages) = 36
(55 x 16kb pages) = 37	(56 x 16kb pages) = 38	(57 x 16kb pages) = 39
(58 x 16kb pages) = 3A	(59 x 16kb pages) = 3B	(60 x 16kb pages) = 3C
(61 x 16kb pages) = 3D	(62 x 16kb pages) = 3E	(63 x 16kb pages) = 3F
(64 x 16kb pages) = 40		

** SPECIAL NOTICE ** There is now DUAL MMC1 Chipset handling available which pushes the total PRG banking up to 32kb pages via 2 16kb pages PRG chips.

This was quite a feat performed with the work of imid, EFX, star69, and Marat Fayzullin to get it to run, and because of this DRAGON WARRIOR III, and other "like" games now work! This game for instance uses a 32kb banking mode between two MMC1 chips, and to make them work together they communicate with each other on the "4th" byte of the game image(not including the iNES header.)

[CHR - (Range 1 x 8kb pages -> 64 x 8kb pages)]

-There is NO MINIMUM required for CHR data because some games have the CHR data imbedded into the PRG info.

* NOTE: To figure out the exact size in bytes each of these pages are worth just start at 1 x 8kb pages (aka: 8192 bytes) and just keep adding 8192 bytes more for each "page" higher.

(1 x 8kb pages) = 01	(2 x 8kb pages) = 02	(3 x 8kb pages) = 03
(4 x 8kb pages) = 04	(5 x 8kb pages) = 05	(6 x 8kb pages) = 06
(7 x 8kb pages) = 07	(8 x 8kb pages) = 08	(9 x 8kb pages) = 09
(10 x 8kb pages) = 0A	(11 x 8kb pages) = 0B	(12 x 8kb pages) = 0C
(13 x 8kb pages) = 0D	(14 x 8kb pages) = 0E	(15 x 8kb pages) = 0F
(16 x 8kb pages) = 10	(17 x 8kb pages) = 11	(18 x 8kb pages) = 12
(19 x 8kb pages) = 13	(20 x 8kb pages) = 14	(21 x 8kb pages) = 15
(22 x 8kb pages) = 16	(23 x 8kb pages) = 17	(24 x 8kb pages) = 18
(25 x 8kb pages) = 19	(26 x 8kb pages) = 1A	(27 x 8kb pages) = 1B
(28 x 8kb pages) = 1C	(29 x 8kb pages) = 1D	(30 x 8kb pages) = 1E
(31 x 8kb pages) = 1F	(32 x 8kb pages) = 20	(33 x 8kb pages) = 21
(34 x 8kb pages) = 22	(35 x 8kb pages) = 23	(36 x 8kb pages) = 24
(37 x 8kb pages) = 25	(38 x 8kb pages) = 26	(39 x 8kb pages) = 27
(40 x 8kb pages) = 28	(41 x 8kb pages) = 29	(42 x 8kb pages) = 2A
(43 x 8kb pages) = 2B	(44 x 8kb pages) = 2C	(45 x 8kb pages) = 2D
(46 x 8kb pages) = 2E	(47 x 8kb pages) = 2F	(48 x 8kb pages) = 30
(49 x 8kb pages) = 31	(50 x 8kb pages) = 32	(51 x 8kb pages) = 33

(52 x 8kb pages) = 34	(53 x 8kb pages) = 35	(54 x 8kb pages) = 36
(55 x 8kb pages) = 37	(56 x 8kb pages) = 38	(57 x 8kb pages) = 39
(58 x 8kb pages) = 3A	(59 x 8kb pages) = 3B	(60 x 8kb pages) = 3C
(61 x 8kb pages) = 3D	(62 x 8kb pages) = 3E	(63 x 8kb pages) = 3F
(64 x 8kb pages) = 40		

[Sect. 5: - Miscellaneous Information on the NES Format]_____

* The End of the CART.NES file ...the Title]

If you wish you can add a title to your NES ROM and below shows you how to do that manually, and the size limits for that title.

- 1) First at the very end of the ROM you will eventually ADD 128 bytes to the ROM. But first, which is part of this 128 bytes, is the addition of the NES Title. Starting at the first added byte type in the name normally in the character part of the hexeditor.
- 2) Next, to finish this is successfully, go over to the HEXidecimal part of the HEXeditor and add in double zeros (00) until you have added in a TOTAL OF 128 BYTES (which includes the the Title you added yourself.) (So subtract the total bytes of the title from 128 so you do it right.)
- 3) If you did it right you now have successfully added a title to your NES ROM and it will show in the title bar in the ROM checkers like NESImage.

[Sect. 6: - Let's Get Cracking on Prepping Your RAW/FFE/PasoFami Image]_____

* "RAW" Image Conversion to iNES

- 1) For starters check and see if that RAW NES rom image you got is or is not interleaved. (See #3 if it's interleaved.)
- 2) Great! It's not interleaved so what you need to do is to just add 16 bytes onto the beginning of this ROM image. Then just add the standard NES header to the ROM as stated in Section 1, and then toy around with the mappers until you can or can't get the game to work or do something.
- 3) Crap, it's interleaved, don't worry it's fixable. What you need to do is to write or find a program that will UN-interleave your ROM. This is the process of seperating all the even and odd bytes. The even bytes are the ROM (or PRG) data, and the odd bytes are the VROM (or CHR) data if that exists.
- 4) Next, in MS-DOS, re-join the PRG and CHR data in the proper order. (Which is PRG data before CHR data) First take the split ROM and VROM files and name them something like PARODIUS.PRG(ROM) and PARODIUS.CHR(VROM)... (Parodius is a Japanese side shooter game I'm using for this description)
- 5) Now type: copy /B PARODIUS.PRG+PARODIUS.CHR PARODIUS.NES
(This is a BINARY copy mode in DOS which will glue the two files together where the PRG ends and the CHR begins.)
- 6) Now do as it says in Step #2 and just add the 16byte header to the top of the NES rom, and then add the NES header as stated in Section 1.

* "FFE" Image Conversion to iNES

- 1) To start check and see if the rom is in a standard file format as shown in Section 3. If it isn't check to see if it is 512 bytes larger than

normal. If so, this is most likely a FFE header.

- 2) To remove this get into your HEXidecimal editor and just have it delete the first 512 bytes from your NES rom.
- 3) Then just add a 16byte header to the top of the NES rom, and then add the NES header as stated in Section 1.

NOTE: If this rom ends up being interleaved as well just follow the guidelines set above in the RAW rom format section for UN-interleaving a ROM.

BE ADVISED that Some NES roms maybe seem to be 512 or 1024 bytes larger than normal. This is because it either has a trainer, or a combination of a trainer and a FFE header. My best advice to you is that if its a trainer leave it, if its a FFE header remove it. If the rom is 1024 bytes its a combo of both, just remove the first 512 bytes and you'll be ok.

* "PasoFami (*.PRG/*.CHR -ONLY-)" Image Conversion

- 1) Start by deleting that annoying PRM file. This file is used by the PasoFami emulator just as the 16byte header that iNES uses.
- 2) Next is the easy part, a simple binary copy (again I'll use Parodius as the example.) Type: copy /B PARODIUS.PRG+PARODIUS.CHR PARODIUS.NES (This is a BINARY copy mode in DOS which will glue the two files together where the PRG ends and the CHR begins.)
- 3) Now do as it says in Step #2 in the RAW section and just add the 16byte header to the top of the NES rom, and then add the NES header.

NOTE: Some PasoFami images such as Super Mario Bros. 2 for example are in the Famicom Disk System format (EX: SMB2J.DKA) and can't be converted. When they can be, I'll add that info here.

[Sect. 7: - Information (Recommended Tools for Converting iNES Images)]_____

* RECOMMENDED TOOLS FOR USE IN THE ABOVE SECTIONS:

HEXeditor(BEDIT.EXE) - (ie: Norton Diskedit or BEDIT) -I use BEDIT, it's tiny with lots of commands, and very efficient. Using BEDIT you can just use the arrows to move around across the top line with the first sixteen bytes (0->15) where your header work is done.

CHRUTIL.EXE - This tool shows you the CHR data in a RAW, FFE, PasoFami, and NES file as PCX picture to view in MS Paint, etc. With this you can determine the correct sizes of the CHR pages. Also you discover if an image is in an interleaved format by seeing the PSX boxes scrambled with spaces between the CHR(sprite/character tiles).

UNLACE.EXE - This tool is something I had custom made for me. What this does is uninterleaves NES roms from the RAW, FFE (removes the trainer by use of command), and iNES format. What that means is that it splits up to rom and puts all the PRG data in one file called (ROM) and all the CHR data in a file called (VROM). And as my manual says above you just got to binary copy them back into one file ROM+VROM = NES rom. Then add the iNES header and you just tinker with it until it works, that's if its supported yet.

Also to save time, I do recommend the use of DiskDude's NESImage from version 3.30+ because it can show you in its menu all the PRG and CHR sizes, as well

as all the other options available to you. This is just for people that don't want to hexedit their rom and just want to get it working right away... impatient aren't you? :)

NESFIX.EXE - If you do use a crummy editor like DiskDude's NES Image remember to clean out bytes 8->15 so games will work in Extended iNES Header format compatible emulators. To do this find and use NESFIX, and just run it like so... NESFIX CART.NES and it will clean out the extra data. Not only that this will save time in that it takes a fraction of the time to do it one by one manually.

2FDS.EXE - This excellent little program will convert a Pasofami (.DKA) Split Cart, Famicom (.FAM), and (.FF3) Far Front East Famicom Disk System image into a .FDS image for FWNES 98 v.300+ to use with its complete Famicom Disk System support. To get the instructions just type the filename in once to get the commandlines for it.

And finally I recommend you have a NES emulator that is as far along in development, or farther, than iNES, and a few smaller sized games to study with in conjunction with my text.

[Sect. 8: - UPDATES (The History of the iNES Header/Format Documentation)]_____

* UPDATE INFORMATION:

[2.2] - I Mask myself with a promise to myself to quit doing this list, but the information says otherwise below...so much for that.

- In Section 1 I added a section on the Famicom Disk System format established by Yang Fanwen for FWNES95 v.300+ (.FDS). This section explains how the file is setup, and so on.
- In Sections 2 and 3 added notes on the FWNES FDS format as it was needed.
- In Section 7 added a new recommended tool of 2FDS.EXE to convert those ugly .DKA and .FAM files to the .FDS format for complete support in FWNES v.300+.
- Added 2FDS.EXE to the ZIP to make things easier on all of us who care.
- Added and removed a few quotes and bits of humor.
- And who wouldn't have guessed, more fixes and changes to old information.
- Also to save my sanity and file size the History is cut to three updates.

[2.1] - Great merciful crap an overhaul and addition! ...Need rest...zombie.

- Removed the old Section 3 which were valid game ROM sizes.
- NEW Section 3 - Those Many Mysterious Mappers Now Supported section. Here I set aside the growing list of mapper support thanks to FanWen and FWNES98.
- FWNES98 acknowledged as the #1 NES Emulator, as it should be so check it out video and audio is superb. CastleVania III is an excellent test of that.
- Options for the use of the second half of byte 6 moved to Section 2.
- Section 7 overhaul; now BEDIT and CHRUTIL explained and how to use them.
- This list is NOW PKZIPED with BEDIT, CHRUTIL, UNLACE, and NESFIX so now you can fix all your own things with the tools I use.
- Cut the Update List because it got way too long pushing 100 lines. I bet you could care less and don't even read this stuff.
- And as always I've updated and fixed grammar and spelling bugs I usually do not catch due to pulling late nights (usually) writing this.

[2.0] - Wow v2.0! "SHOW ME THE MUSHROOM!" - Toad McGuire ...More mappers! =)

- In Sections 1-8 I reformatted some text and removed useless info.
- In Section 1 added a snippet on what byte 7 does in conjunction with byte 6 and explained that 7 = Tens digit and 6 = ones digit for the 16+ mappers. And fixed the chart so that 0 is the beginning digit, not 1. Stupid error.

- In Section 2 I added a notice about NESFIX so people can easily repair there polluted headers so they will work in Extened iNES Header format players.
- Added in Section 2 the all important further extension to the Extended iNES Header format created by FanWen. Mappers 66-70 80-84 are now added, and a rename to the Konami VRC 2 1a and 1b to VRC 1b and 1a.
- Reformatted the CHR and PRG cart size lists to fit into 1 screen in Section 2 as well.
- And in Section 7 I added the recommended tool of NESFIX.EXE to remove that nasty garbage out of iNES headers.
- More BS errors in grammar and spelling fixed again. Common occurance is it not? >:) Maybe I need to buy Hookt awn Fawniks and Lurn too Reed. =)
- In this Section 8 it got to damned big so now I'm limiting the list to the three most current revisions for now on. (Like anyone really looks back to read that stuff anyways, the important stuff is above.)

[End Of File] -Enjoy!