Hacking FAQ

Summary

This is a FAQ and a list of resources regrouped by themes. It contains links to wiki pages, documents, forum threads and external links. Nothing that is posted here is new, this page has as purpose to help newcomers to find information depending of what they are hacking. This list does also not contain everything (obviously). However, it should be enough to get someone started and progress in the right direction.

You are invited to improve the Q&A, topics, documents, links and forum threads suggestions. If you ever see a broken link here, fix it or please notify someone.

FAQ

Here are the most asked questions and common interrogations of anyone starting hacking FF6.

How can I apply a patch ?

To apply a patch, you need a patcher with the good format. There are many patch format but most commons one are IPS and XDelta. Most patches are IPS format but that does not mean it is the best format, only one that has been used for a long time. One good format is BPS, a XDelta variant. It offer advantages over IPS and XDelta such as smaller patche size, checksum storing and infinitely sized files are among them. To create or apply a patch, you can use beat (BPS), Floating IPS (BPS / IPS) or Lunar IPS. You can also use the online patcher hosted here at ff6hacking.com/patcher/, courtesy of Marc Roblero. Finaly there are other patching utilities on this wiki page.

Another thing to know about patches is that in the SNES ROM hacking world, there are patches for ROMs with a header and patches for ROMs with no header, see below questions.

What is a header and how can I add or remove it ?

A header is 0x200 bytes of padding at the beginning of the ROM. There's not really a functional difference between a headered and an unheadered ROM, but you have to take into account the extra bytes when you are hacking. We suggest you use none and apply one only when you are applying a patch requiring one or use an editor requiring one. It will simplify your life. You can add / remove headers with an utility like SNEStuff.

After applying a patch, my game crash! What have I done wrong ?

A: If the game is violently crashing when you load the game or at another point, reach the first battle, etc., you are almost certainly applying the patch to the wrong ROM. There are two different versions of the game (1.0 and 1.1) that have minor differences. In addition, for each version of the game there are both headered and unheadered ROMs. You have to apply the patch to the same type of ROM that the creator used to make the patch. Usually the ROM you should use is specified in the readme or wherever you downloaded the patch from. Don't ignore this info!

What is an offset and a ROM address ?

Both terms are equivalent. One can refer to a ROM address by the term offset, it is the address you see in the hex editor. However a SNES address is different. The two more common ways the ROM cartridge are mapped to memory are called LoROM and HiROM. In the case of a FF6 cartridge we have a **HiROM + FastROM** ROM type. In a HiROM mapping occupies banks $40-$7D and $C0-$FF and HiROM games commonly access the ROM at banks $C0-$FF exclusively. In the case of FF6, since the ROM size is 24Mbit, the possible addresses are in the range of $C00000-$EFFFFF. This mean address or offset $000000 in a hex editor is SNES address $C00000, $100000 is $D00000, etc. If you expand the ROM too 32Mbit, you add addresses $F00000-$FFFFFF. It is possible to expand beyond this point but then the address following $FFFFFF will be $400000, matching what is in the hex editor (see below question). If you want to know more details about SNES memory mapping, there are documents available on this wiki page

---

How can I expand my ROM and what does expansion affect ?

SNEStuff or Lunar Expand can do ROM expansion. Expansion basically only add more space for code, data or graphics. The most common expansion is from 24Mbit to 32Mbit. This add 16 new banks which is enough for most projects. FF3usME can do 28Mbit and 32Mbit expansions. The ROM mapping for an expansion bigger than 32Mbit is called ExHiROM which has some specificity. Some utilities (like assemblers) might not work as expected with some assembly instructions whn using ExHiROM offsets. This is almost a case by case thing but there are always a way to work in ExHiROM and utilities that exist to do so.

What is Hex? I don't understand it, please help!

Hex stands for hexadecimal, which is a base 16 numeral system while our decimal system is on base 10. If we break down this more, Hexadecimal is based on Binary System which is used partially when looking at a byte bits. In one simple sentence, in the Hexadecimal system, letters A to F cover the missing digit needed to go to 15. You can search online for Binary, Decimal and Hexadecimal systems and find documents like this one. There is also a forum thread with more resources.

Where is located the code performing task X ?

All the game code has been disassembled and partially documented into text files called bank disassemblies. The disassemblies links are on the document page. Bank $C0 regroup mainly the regular map code and bank $C1 regroup the code of the visual and graphics of the battle screen. Bank $C2 is divided into two parts, first is the battle engine while second half is a continuation of bank $C1. Bank $C3 regroup the menu code while bank $EE has to do with overworld maps and movements. Finally, bank $C5 is the bridge between the game and the SPC-700 (sound). There is also a compressed code block related to the intro that is stored in RAM (bank $7E) and code related to HDMA stored in bank $D4.

How can I learn assembly and where do I start ?

There is no particular way to process. However, ALL the information you need is in the ASM / SNES resources links below. You'll need to get a bit familiar with the FF6 disassemblies, which are readable code text files. At the same time, the RAM map will be of a great help since coding involve most of the time reading from / writing to memory, which each game has mapped its own way. One approach to learn ASM is understanding a small block of FF6 code and try to modify it to get a different result, and code bigger blocks as you improve. An hex editor can do simple edits but for any edit of significant size, you might want to try an assembler like bass, xkas or asar. There are more assemblers but these two are popular ones.

Which utility can edit data X ?

A: Check out the utility page. For general SNES graphics, YY-CHR is a popular editor. For hex editing, HxD is a good all purpose editor and Windhex has table support (to edit text). As for recommended editor for a regular hack (with no data expansion), we recommend FF3usME for any general data, character sprite, animation data menu backgrounds and BRR samples. FF3SE is good for portraits, monster sprites and esper sprites.

Anything map related should be done with FF6LE Rogue (which require an expanded ROM). Avoid Zone Doctor as it has higher chances of ROM corruption. For chests, FF6LE Rogue or FF6MDE are good (which can also do animation data). For compressed graphics, Peer Sprite Viewer in junction with YY-CHR is the way to go (see compressed data list here). For music, there is a specific process that make things faster, refer to the music section down below. For specific palettes SNESpal is a good tool.

You should **avoid** the all purpose editor FF3Edit avaible elsewhere on the internet as it has high chances of ROM corruption. This is why we only have the source code available as a reference and no binary executable. If you are on Mac, FF6Tools is the only FF6 specific editor you can use. For assembly, we recommend for debugging bsnes+ or SNES9x Debugger. For assembling code, bass, xkas or asar are good.

Which data require Hex Editing ?

Anything code related can be done with a hex editor though working with an assembler is faster for medium / large edits. Events and Battle Events should be done with a hex editor. Zone Doctor has an event editor but it can corrupt your ROM. Animation data can be edited with FF3usME or FF6MDE but animation scripts must be hex edited. This is valid for battle event and attack animation scripts. Song instruments must be changed with a hex editor. Some text need to be edited with a hex editor and depending which text it is you'll need a specific table (available here). A good all purpose hex editor is HxD while a good one with table support is Windhex. Any OAM data must be hex edited as well as any data table scattered in the code banks. Finally, some specific data require hex editing, if you don't find something in the editors mentioned in previous question, check the ROM Map, the data you are looking for might not be supported with the existing editors. Anything in the ROM can be hex edited, using the proper editor will cut off on mistakes and fasten your hack progress!

Where is data X located in the ROM ?

Check out the ROM Map. It is not 100% completed but most data known up to now are listed there. There are good chance what you want to edit can be edited with one or many editors (see previous questions).

Is it possible / easy to expand data X ?

It depends. Some things like item or spell numbers would be hard to properly expand because they would require a rewrite of many ASM functions. Some other things like some graphics or text only require to move the data and change a few pointers in order to be expanded. Note that when you expand and move something, you usually break the editor that can edit the data, unless the editor is made to support the expansion. Some simple expansions are possible with FF3usME and there are other editors that can expand specific data. Existing expansion projects are listed here, but what is possible is not limited to this list. It's also always good to plan expansions at the beginning of development.

Where can I find patches and which ones do you recommend for my project ?

There are many place for patches but most of them can be found on our patch section, on RHDN, on Slick Productions and on some other webpages mentioned in our external links section. Now depending on the nature of your project, some patches will be more suitable than others. It's your task to read on them and ask questions if some point are unclear. Concerning FF3us bugs, C.V. Reynolds Compilation include most bugfixes in one patch and is updated regularly. We recommend it over Darkmage's project that had some problems in the most recent versions of his compilation. Finally for music we have a whole section dedicated to songs from other games. Those are not

patches but binary data you need to copy in the ROM manually to replace any song you want.

Should I make my hack on the SNES, GBA or Steam version ?

The most documented version and the one that has most utilities for is the SNES version so in a way it is the easiest version to hack. The GBA version come second: there are a few utilities and a somewhat decent ROM map. However, there are no public code disassemblies. There are a good numbers of GBA patches, notably by Novalia Spirit and LeetSketcher. The Steam version has a few cosmetic mods but the FF6 Steam modding group is not the most active I've seen and I'm not even sure if anybody currently does reverse engineering on the game executable file or unknown game file / data formats. One notable thing to mention is that the Steam version has a Japanese GBA ROM in the game binary blob used for some game data so it's possible to edit a lot with a hex editor. The Steam version also has a file structure unpacker, available here.

I can't find info X or my question was not fully answered! What can I do ?

You can try rephrasing your question or try to explain in detail which difficulty you encounter. There is also our Discord chat where you can get some help, most of the time there will be someone there that can answer you. We are also not the sole resource online. While we try our best to answer every question, you might want to look elsewhere. Slick Forums regroup good FF6 hackers (and FF4 hackers too!), some that visit here. Mnrogar's Den has some nice info too but you can't register and it act more as an archive forum, most members having moved to Slick Forums. You can also try RHDN for general hacking or FF6 hacking questions.

Resources

Here is a list of resources categorized by type of hacking task.

Assembly

This is the base of changes involving the battle system, menu changes or special things like custom event commands. All expansions also require ASM changes.

- Assembly & SNES technical resources
- 65816 Opcodes list (Hatzen08)
- 65816 Opcodes detailed (TheGun)
- SNES Registers
- SNES / GBA Coding Sticky Thread

Events

Events govern how the game progress and all govern all the cut-scenes. There are roughly 256 event commands with parameters. Those commands are aggregated in the event data detailed in the event dump document. Another part of the event system are event bits. They play some sort of milestone, like “event X happened” so other portions of the game can be accessed or triggered. The masking of map NPCs use the same system.

- Event Dump (Imzogelmo / Novalia Spirit)
- Event Command List (Lockirby2)
- Event Command List (Imzogelmo)
- Event Bits Document (Novalia Spirit)
- Event Commands (Yousei)
- Movement Codes (Yousei)
- Overworld Character Commands (Yousei)
- Overworld Vehicles Commands (Yousei)
- Event Hacking Tutorials (Lockirby2)
- All Things Event Hacking Thread
- Basic Video Event Tutorial (madsiur)
- How to Make Space for Events (madsiur)

Battle Events

Battle Events are completely separated from the regular events. They have their own animation system (shared with Attacks Animations). The animations and character movements are animation scripts called in the battle event scripts.

- Battle Events Scripts (Everything)
- Battle Events Scripts (madsiur)
- Battle Event Commands (Everything)
- Animations Commands (Everything)
- All About Battle Events (outdated)

AI Scripting

AI scripting is done with FF3usME. It consist mainly of commands, sub-commands and conditional “if” statements. The vanilla monster scripts are the best source on “how to do thing X” in a script.

- AI Scripting Document (BTB)
- Monster Commands Script Guide (Terii Senshi)
- Enemy Commands Script Documents (Master Zed)

Animations

Animations can be edited with FF3usME or FF6MDE. They regroup spell animation but also battle animations. Battle events and attack animations share the same animation system but Attack animations have a 8 or 14 bytes data that tells the palettes, sound effect and animation scripts.

- Animations Scripts (Everything)
- Animations Commands (Everything)
- Animation Data Format

Spriting

Character sprites can be imported / exported with FF3SE or FF3usME or FF3SpriteEd (now incorporated in FF3usME). FF3usME has the spritesheet format that is the most used. There is a certain color order to respect in the sprite in order to make a good import.

- Overworld palette assignation guide (Runelancer)
- Editing Character Sprites and Portraits (madsiur)
- External Pixel Art Tutorial List

Music

Songs are composed in data proper to FF6 that is then read and transmitted to the SPC-700. You can change manually a song data, but it can be a tedious operation since you need to keep the 8 channels of the song coordinated at any time. There is also a technique involving a RS3 editor to convert a MML file to FF6 compatible song data.

- Music Commands (MetroidQuest)
- Song Scripts (Everything)
- Song List
- Music Commands
- MML Commands
- Insert a Song Hack in the Game(madsiur)
- Available Song Hacks
- FFVI Music Hacking (Gi Nattak)
- How to add more songs (Gi Nattak)
- Injecting Music with RS3exTool2 (Gi Nattak)
- Making FF7 Boss Theme for FF6! (part 1) (Jackimus)
- Making FF7 Boss Theme for FF6! (part 2) (Jackimus)

Sound

The SNES use BRR samples as music instruments, meaning a single sample will play an array of notes with different pitch and duration. The samples can be replaced with FF3usME or inserted manually.

Each sample have a specific ADSR, Pitch and Loop data aside of the BRR data. As for SFX they are generated from simpler BRR samples and each SFX has a two scripts, similar to the concept of a song music data.

- Sound Effect Scripts (Everything)
- Sound Effect List
- BRR Sample List (instruments)
- Instrument set Format
- Squaresoft BRR Database
- How to add more instruments (Gi Nattak)

Misc

- How to edit compressed graphics (madsiur)
- How to edit tileset graphics (madsiur)

From:
https://www.ff6hacking.com/wiki/ - **ff6hacking.com wiki**

Permanent link:
https://www.ff6hacking.com/wiki/doku.php?id=hacking_faq&rev=1521630195

Last update: **2019/02/12 11:37**