Final Fantasy Tactics Advance Advanced Hacking Guide

by JoKyR

Updated to v2.0 on Jan 5, 2006

```
-----------------------------
Final Fantasy Tactics Advance
         US Version
     ADVANCED HACK GUIDE
          by JoKyR
-----------------------------
```

Version 2.0
12-21-2005

```
-----------------
TABLE OF CONTENTS
-----------------
```

```
---------------
I. INTRODUCTION
---------------
```

Hello. This is my very first video game faq ever, so please be gentle. I'll try to be as concise as possible. This guide was originally intended to focus solely on hacking the stats of the four growth items (Sequence, Sapere Aude, Peytral, and Acacia Hat). Other hackers have already done an excellent job of cracking this game to pieces, and this was one of the only corners of the game that had been left untouched. However, as you can see in my version history, it was rejected. Twice.

Some folks on the message board were very kind, and politely suggested that I find ways to add more codes. I thought long and hard about this, and decided that I didn't want to repeat the work of others. However, there were a few other tiny little areas of the game that I might be able to hack "a few more codes" out of. These new codes really have nothing to do with each other, and are united in this guide only by the common trait of being left out of previous faqs. Indeed, none of them are really as useful as the growth item codes, but you may be able to find an interesting purpose for them.

So, before you go any further, you need to know that you won't find any master codes here, nor should you expect a comprehensive code database. This site already has several faqs with all the standard item, character, and monster bank codes. The only codes I'm including in this guide are codes that I couldn't find anywhere else, in any format.

You also need to know that this guide is NOT intended to teach you how to hack, write, or encrypt your own codes. Labmaster has an excellent set of primers on www.cheatandwin.com to do that, already. This guide is written primarily for seasoned hackers who want some new information about unexplored areas of the game. I will do my best to explain how the game deals with these specific topics, give you some addresses and values, and offer some sample codes in as many formats as I know how.

I hope you find this helpful. But, if you want to alter the sample codes to different values, I'm sorry to say that you'll have to learn how to do it on your own. Don't worry, though. It's not that hard. In fact, this is the game I learned how to hack with. It has a wonderfully diverse set of challenges that have definitely prepared me for future endeavors. And, while I'm eager to move on to other projects, I suppose I don't mind being forced to juice a little bit more out of one of my favorite games of all time.

-Happy Hacking,
JoKyR
maestro at KYSOff dot com

II. GROWTH ITEMS

II.i. The Basics

The four growth items are earned in the game by completing special Tournament, Clan League, and Link Missions. An item is given as a reward the first time one of its associated missions is successfully completed. These items cannot be sold or traded. Without hacking the item's quantity, you can only ever earn one of each. After you earn (or hack) the item, one of its stat values will increase each time you successfully complete one of its associated missions.

The missions associated with each item are as follows:

Sequence: 40, Link 1, 2, 3
Sapere Aude: 41, Link 4, 5, 6
Peytral: 43
Acacia Hat: 42, Link 7, 8, 9

The stats that increase (and their original values) are as follows:

Sequence: Atk (32)
Sapere Aude: Pow (5)
Peytral: Def (28)
Acacia Hat: Res (2)

The game stores an 8-bit hex value at the following addresses to determine the growth of the item's stat:

```
02002031 = Sequence
02002032 = Sapere Aude
02002033 = Peytral
02002034 = Acacia Hat
```

When you first obtain the item, the hex value is 0x00. Every time you successfully complete an associated mission, this value increases by 0x01. The hex value maxes out at 0xFF (255). Don't worry, the game won't loop it back to 0x00 if you keep completing missions. The stat value is equal to the original value plus the hex value. However, the in-game value maxes out at 255.

II.ii. Sample Codes

"Max Growth Item Stats"

These codes will set all four hex values to 0xFF, giving you the max stats for all four items.

CBA:

```
32002031 00FF
32002032 00FF
32002033 00FF
32002034 00FF
```

MadCatz GS:

```
FEEC9DFC 8CB1
F67D9F9C C9B1
F6EC9DBC C9B1
BE75BFDA 8CB1
```

GSV1/2:

```
04C6E332 2699A82D
D6602D2C 60B0B4B0
DE31F6F8 B12F2F70
B43D05D7 0F313818
```

PARV3:

(Yes, this is a slide code, so it only needs two lines.)

```
704C4B3B AFDE9843
54FA8281 4E7BADFC
```

III. CHARACTER LEVELS

III.i. The Basics

Standard experience levels for all characters (including monsters in the Monster Bank) range from 0-50. Of course, characters generally start with a level greater than zero. Levels greater than 50 can be hacked, but a character that is level 50 or higher can no longer level up. He/she will continue to earn experience points until he/she has 99 experience points, and then will stop earning experience altogether.

I should warn you, though, that unlike some RPGs, or games like Pokemon, in FFTA a character's level has NO direct effect on their stats. Changing a character's experience level will NOT change their statistics, nor will it affect their performance in combat (excepting, of course, a few spells that only affect characters of certain levels). A character's statistics only increase when they level up, and the amount they increase is independent of their level.

So, why would you want to hack a character's level? I can think of three reasons.

First, a character's level affects their competence when completing dispatch missions. The higher the level, the better. Check out Terence Fergusson's Mechanics Guide for more info on this.

Second, the average level of the members of your party determines the approximate levels of the enemies you encounter in random engagements. If you have a large party, but have only been concentrating on leveling up a few of its members, then you will soon find that leveling up becomes difficult, in general. The enemies you face will all have levels in the middle of your own level range. So, the low level members will be too weak to survive long against the relatively powerful enemies you face, and the high level members will be getting very little experience for fighting relatively low level enemies. Adjusting the levels of your characters can fix this problem, and can be used to make enemies stronger or weaker, as needed.

Third, you may want to lower your characters' levels to give them more opportunities to level up. Their stats stop growing after they reach level 50, so why not knock them back down to zero and start all over? Or, if you're really maniacal, you can use a pair of 8-bit constant write slide codes (included in the sample codes section) to set their levels to 0 and their experience points to 99. Then, every time they earn any amount of experience, they'll level up... to infinity. Just four lines of code and your characters get obscenely powerful, fast. Sure, you could just hack their stats manually, but that takes quite a few more codes, and may not be as fun.

Each experience level is an 8-bit value ranging from 0x00 (level 0) to 0x32 (level 50). These values are located at the following addresses:

```
02000089 = Experience Level for Character 1
02000191 = 2
02000299 = 3
020003A1 = 4
020004A9 = 5
020005B1 = 6
020006B9 = 7
020007C1 = 8
020008C9 = 9
020009D1 = 10
02000AD9 = 11
02000BE1 = 12
02000CE9 = 13
02000DF1 = 14
02000EF9 = 15
02001001 = 16
02001109 = 17
02001211 = 18
```

```
02001319 = 19
02001421 = 20
02001529 = 21
02001631 = 22
02001739 = 23
02001841 = 24
```

III.ii. Sample Codes

"ALL Lvl 0"

These codes will reset all the characters' levels to zero.

CBA:

```
32000089 0000
32000191 0000
32000299 0000
320003A1 0000
320004A9 0000
320005B1 0000
320006B9 0000
320007C1 0000
320008C9 0000
320009D1 0000
32000AD9 0000
32000BE1 0000
32000CE9 0000
32000DF1 0000
32000EF9 0000
32001001 0000
32001109 0000
32001211 0000
32001319 0000
32001421 0000
32001529 0000
32001631 0000
32001739 0000
32001841 0000
```

MadCatz GS:

```
3477720A 53B3
0443622B 73AF
1473720B 32A3
2407E2AA 12DF
3437D28E 53D3
0403C2AF 73CF
1433D28F 32C3
A6450A2E 12BF
B6753A0A 53BF
86412A2B 73A3
96713A0B 32AF
A605AAAA 12D3
B6359A8E 53DF
86018AAF 73C3
96319A8F 32CF
21576202 732E
31677222 7322
```

```
01536203 123E
11637223 1232
2117C286 734E
3127D2A6 7342
0113C287 125E
1123D2A7 1252
A3552A02 7322
```

GSV1/2:

```
2C2F9FD0 CE30C39C
6F5CBD1B A45B9C08
6F695F0A F499AAA4
12DE170D BFC89407
642401B5 D7041F58
4CD2329E 8EF9F1F2
5F3B546E 2502C602
0E90C451 4C7463A8
A093A5DA C8657070
8DB0FCC4 CDA57EE7
B7BB95EF 6FFDF9FD
017E263A 33FB5224
10D27DE1 819518F4
4A5ECBF7 6ABCF25B
DD6458A2 414C2328
2E5A448E EE041101
DFEDBD4C 7A0CECB2
88FC1DB8 E2795C9E
D7E808F8 EAAC2335
5448CC03 36DDE4B0
25A24576 B4026B8B
9EF07E18 9CFE8F21
DEEE30D4 D0D117C7
9ED13A34 216D37DE
```

PARV3:

(Yep, another two line slide code.)

```
DEA4AC35 AC0EC0FE
69572D1C 2F49DEEF
```

"ALL Lvl 50"

The opposite of the previous batch, these will max out every character's level at 50.

CBA:

```
32000089 0032
32000191 0032
32000299 0032
320003A1 0032
320004A9 0032
320005B1 0032
320006B9 0032
320007C1 0032
320008C9 0032
320009D1 0032
32000AD9 0032
```

```
32000BE1 0032
32000CE9 0032
32000DF1 0032
32000EF9 0032
32001001 0032
32001109 0032
32001211 0032
32001319 0032
32001421 0032
32001529 0032
32001631 0032
32001739 0032
32001841 0032

MadCatz GS:

1C33F3CB B6C1
2C07E3EA 96DD
3C37F3CA D7D1
0C43636B F7AD
1C73534F B6A1
2C47436E 96BD
3C77534E D7B1
8E018BEF F7CD
9E31BBCB B6CD
AE05ABEA 96D1
BE35BBCA D7DD
8E412B6B F7A1
9E711B4F B6AD
AE450B6E 96B1
BE751B4E D7BD
0913E3C3 965C
1923F3E3 9650
2917E3C2 F74C
3927F3E2 F740
09534347 963C
19635367 9630
29574346 F72C
39675366 F720
8B11ABC3 9650

GSV1/2:

47386AC1 BB66F252
8D0B558D 7F17160C
BEE4A230 FBAA3756
9344CA71 6B0F6766
22705BC0 61FC4DDE
11B6F443 76348FA1
A36C8D67 AEACF8E6
4F32BD13 0F3EDF89
46050E46 D09A542A
21560A46 AF9FEEFA
B2510484 D60A8555
DC6D30DE 19B13C24
12A73DE4 4693BC74
BAAF5E60 7BA1CDE8
C3ED4359 F54A9656
A75CA94A C55ABE81
0BE30480 A582130E
```

```
16A2D9BF 59153750
475BB858 15F7D8A7
D0E89375 E2470A17
1BF51D7C F76F089A
4FB7A57D 0E2ABD58
4A005DFF 67FA1812
0912C1B7 8D605113
```

PARV3:

(Once again, slide code heaven.)

```
DEA4AC35 AC0EC0FE
369104C2 0DF80DA5
```

"ALL Lvl 0 Exp 99"

These codes are described above in The Basics. Keep your codes turned on while you play, and your characters should boost their stats extremely quickly. I'm only writing the PARV3 codes, though, because it's the only hacking device with 8-bit slide codes. Sorry. It shouldn't be hard to just mix the above level zero codes with Labmaster's "Level Up" codes, though.

PARV3:

```
DEA4AC35 AC0EC0FE
69572D1C 2F49DEEF
59FCEE15 64D3D6EF
388B9EAD A88A0879
```

```
---------
IV. NAMES
---------
```

Unfortunately, this section is not yet complete. Some of it is pure speculation. I still have a lot of testing and hacking left to do, but I've uncovered enough information that other hackers may still find this section useful. TetrisTheMovie seems to be the first to have broken ground on this subject, and I highly recommend you read his "Clan and Hero Name changer codes" before you read this section. I will be building upon and further expanding the information in his guide. That said, here we go.

Any given name in FFTA is represented by an address that contains another 32-bit address, that points to a string of bytes which represent the individual characters of the name. In other words, when the game checks for a name, it goes to one address. Then, it is directed to a new address which contains either the first character of the name, or a code for the starting point of a name. Finally, it will read all the characters of the name until it reaches the byte(s) designated as the end of the character string.

For example, when looking for the name of the main character, the game will first go to address 0x02000080. By default, that address contains the 32-bit value 0x02001F1C. At this point, if you've read TetrisTheMovie's faq, you will hopefully recognize 0x02001F1C as the address for the first 16-bit character of the name. If you've left the character's name as Marche, then it should be a capital "M", or 0xBC80. The next two bytes would be lower case "a", or 0xCA80, and so on until you reach the end of the name and the value 0x0000, which signals the end of the string.

So, why does the game do this? Why doesn't the game just store the character string at the original address in the first place? Because, as I'm sure you've noticed, the vast majority of the names in FFTA cannot be customized. Again, if you read Terence Fergusson's Mechanics Guide, you'll find several long lists of almost all the possible names used in FFTA. These names are stored as character strings in the actual ROM, and therefore can't be altered (easily). Rather than creating a new character string in the RAM, the game simply creates a 32-bit address that points to the location of the character string in the ROM.

The original addresses (the pointers) for each character's name are as follows:

```
02000080 = Main Character's Name (Set to 0x02001F1C by default.)
02000188 = Montblanc (Default 0x085512C7)
02000290 = Character 3
02000398 = 4
020004A0 = 5
020005A8 = 6
020006B0 = 7
020007B8 = 8
020008C0 = 9
020009C8 = 10
02000AD0 = 11
02000BD8 = 12
02000CE0 = 13
02000DE8 = 14
02000EF0 = 15
02000FF8 = 16
02001100 = 17
02001208 = 18
02001310 = 19
02001418 = 20
02001520 = 21
02001628 = 22
02001730 = 23
02001838 = 24
```

I suspect that there are similar addresses for the clan name, enemies, and monster bank monsters. I just haven't looked for them all, yet.

It is also important to note that, for some unknown reason, the game uses two different schemes for representing the characters used in names. I suspect this has something to do with using the main character's name and the clan name in the game script.

The first scheme is comprised of the 16-bit values found in TetrisTheMovie's guide. This scheme is only used for the main character and clan names. There is no code to signal the beginning of a character string, and therefore names can be truncated by slightly altering the pointer address. (e.g. If your main character's name is "Marche", then changing the pointer address to 0x02001F1E will result in the name "arche".) To signal the end of a name, use the value 0x0000.

The second is an 8-bit scheme very similar to the 16-bit scheme. I haven't yet tested all the values of this scheme, but it seems that most of them can be obtained by taking the left two digits of the 16-bit values and adding 0x01. For instance, "M" is represented by 0xBC80 in 16-bits, and 0xBD in 8-bits. When using the 8-bit scheme, it is necessary to signal the start of a character string with the value 0x01. In general, these names cannot be truncated. The pointer address must contain the location of the 0x01 byte. Pointing an address to the middle of a character string will generally result in garbled pixels, or possibly a system crash. Similar to the 16-bit scheme, the value 0x00 is used to end a character string.

As far as I can tell, the two schemes are interchangeable. The main character's name can be pointed to a ROM value, and secondary character names can be pointed to the main character's string without problems. You could probably, therefore, code the character string at 0x02001F1C using the 8-bit scheme, but I would guess that this might cause problems for the game script.

At this time, I don't have any kind of list of the locations of the different names stored on the ROM ready for you. However, I will say that they're all in the 0x085XXXXX area. If you have a ROM, though, using the information I gave you above, it shouldn't be too difficult to create a table file for the 8-bit scheme, and use a hex editor (along with TFergusson's guide) to find the ones you want.

Now, here's the part where we start taking advantage of all this information. There are several ways that we can change names in FFTA:

1.) If you just want to change the name of the main character or your clan, you can follow the method outlined in TetrisTheMovie's guide to change the name to whatever you want.

2.) If you want to change a character's name to something on the list of names programed in the ROM, you can change the value of the 32-bit address to point to the new name's character string. Keep in mind though, that doing this for the main character, clan, or any of the storyline characters (Ritz, Cid, etc.) probably won't change their names in the game script.

3.) If you want to make slight alterations to a name that is programed in the ROM, you may be able to use ROM patch codes to change the character string. Unfortunately, I don't have any experience with that type of hacking, so I can't help you much, there. And, I'm not 100% sure, but I believe such alterations would only be temporary, and the codes would need to be used every time you play the game.

4.) This last one excites me the most. If we can find an unused section of the RAM, then we should be able to create our own brand new character strings, and then point the original addresses to the new strings. By doing this, we could change any name to anything we want! I haven't even started testing this, though. I imagine it will take some work to find a good place to put the character string, where it won't get any interference but will still be copied to the game's battery file.

Well, that's pretty much where I'm at right now. I don't even have sample codes for this section, yet. But, if any prodigious hackers out there want to take this info and do something with it, please keep me updated on your progress. I'm very interested to see what can be done.

Fin.

V. VERSION HISTORY
----

v.2.0:
12-21-2005
Changed name to "Advanced Hack Guide"
Expanded faq structure.
Added sections on "Character Levels" and "Names"

v.1.01:
12-14-2005
Minor grammar and comprehension corrections.
Resubmitted to GameFAQs, begging CJayC to reconsider.
Rejected by GameFAQs because "The topic of your guide is too narrow, and I've seen these codes elsewhere, so it's definitely not that unique."
I'm not really bitter, but if anyone else has seen these codes elsewhere, please let me know. That site probably would've saved me a lot of time and effort.

v.1.0:
12-13-2005
Original document "Growth Item Hack Guide" focused only on hacking the stats of the four growth items (Sequence, Sapere Aude, Peytral, and Acacia Hat).
Rejected by GameFAQs due to "lack_of_content".

VI. CREDITS

Special thanks goes first and foremost to Labmaster for, essentially, teaching me how to hack. His FFTA guide, as well as his hacking primers have proved invaluable.

Thanks to Terence Fergusson and TetrisTheMovie for their excellent guides, which I have referenced several times in this document.

And finally, thanks to mikeTherob, bobeta, and DeMatt on the GameFAQs message boards for their support and criticism.

-JoKyR