

Replacing music in MvCI

Last edited 10/8/17 by [MasterWholigans](#)

Hi there! This is a short guide to walk through music-modding Marvel vs Capcom: Infinite on PC. I'm a beginning modder myself, so I'll try and make this as clean as possible to walk through and follow the steps that I took to create your very own music mods! Please feel free to reach out to me if you have any questions or would like to add any content to this guide [at my Twitter page](#). Check out my own [Full Soundpack here!!!](#)

Tools Needed

[DFPSSlayer](#)

- This is a decryption program that removes the security put in place on all of the game's files. We'll use this before unpacking the files. (Thanks, [Altimor!](#))

[QuickBMS](#) with the associating [MvCI.bms](#) file (right click link and hit Save Link As)

- This will unpack MvCI's files to let us explore and adjust as needed, once the game is decrypted.

[Eternity Audio Tool 1.0](#) (You must have 1.0 or higher)

- This tool is made to replace the sound in an .acb file format. While MvCI seems to have .acb files, the game doesn't seem to read those files when running the game. *Instead*, we'll be using Eternity 1.0's .hca file converter that's built into the program.

Any Hexadecimal Editor, like [HxD](#)

- This is gonna be the heart of sound mods in MvCI. You will have to use Hex editing to replace old sound data with our own sound data. Don't worry! It's not as bad as it seems!

[Wavosaur](#)

- Free audio tool that can create custom looping points in your .wav files. Pressing L in the program will bring up the Loop function. *Extremely* useful for those modders that love attention to detail!!

[Python 2.7](#)

- We'll be needing this installed to run the u4pak.py properly.

[U4pak](#)

- Once we have the mod set up, this program will compile everything into a nifty, clean .pak file that MvCI will read flawlessly!

[Audacity](#), or any preferred audio editing program.

- Not necessary really, but useful! Perfect little tool to help convert files, cut down music, edit, etc.

An External HDD, or tons of spare space.

- Also not necessary, but between the game install, the decrypted files, and the unpacked files, I recommend having at least ~150GB free *somewhere* to keep all of these unpacked materials and backup files. I was able to use a spare external 1TB HDD. Again, not necessary, but important to know this before you get started!

The Process

The process should fall in these steps, in which I will lay out each step in full detail further below.

- *Decrypt the Game's Files (One Time Process)*
- *Unpack the Game's Files (One Time Process)*
- *Convert your Music to .hca*
- *Copy the Hex data for the Converted File.hca*
- *Paste the Hex data into the .uexp file that you are wanting to mod*
- *Pack it up using U4Pak*
- *Test and play!*

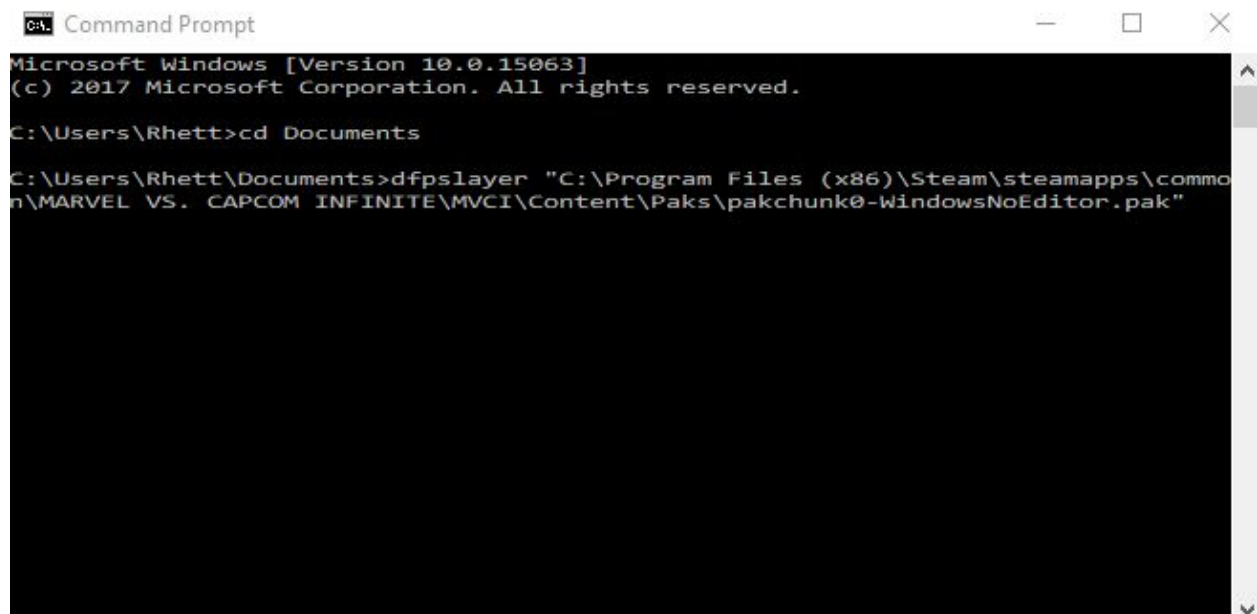
Let's start!

Step 1. Decrypt the Game's Files

This step will remove the security placed by Capcom in order to read the data without any interruptions. We will use DFPSlayer in a command line prompt to run this program.

If you'll take a look at your MvCI's install folder, inside MvCI/Content/Paks you'll see two .pak files, one named "pakchunk0-WindowsNoEditor.pak" and the other "pakchunk0-WindowsNoEditor_P.pak" We will be needing to unpack "pakchunk0-WindowsNoEditor.pak" specifically since it contains all of the music in the game.

Keep in mind where your "pakchunk0-WindowsNoEditor.pak" file is located, because now we're going to decrypt it! On your command line, you're going to go the folder that you keep the dfpslayer.exe and run the program in such a fashion that directs it to where the pakchunk is located.



```
Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Rhett>cd Documents

C:\Users\Rhett\Documents>dfpslayer "C:\Program Files (x86)\Steam\steamapps\common\MARVEL VS. CAPCOM INFINITE\MvCI\Content\Paks\pakchunk0-WindowsNoEditor.pak"
```

Run the program (which will take a while) and you will end up with a “pakchunk0-WindowsNoEditor.pak.decrypted” file! We’re done with this step!

Step 2. Unpack the Game’s Files

This next step will take our new, decrypted file, and unpack all of its contents for us to poke around in! We will be using QuickBMS to execute this.

Open up quickbms_4gb_files.exe and it will run you through three steps:

1. Select MvCI.bms
2. Select “pakchunk0-WindowsNoEditor.pak.decrypted”
3. Select somewhere you would like to dump all the files out. Any spare space you like it fine, preferably not mixed in with your current install of MvCI

Allow this program to dump your files (which will take time) and eventually you’ll get a nice folder with all of the game’s contents! Well done!

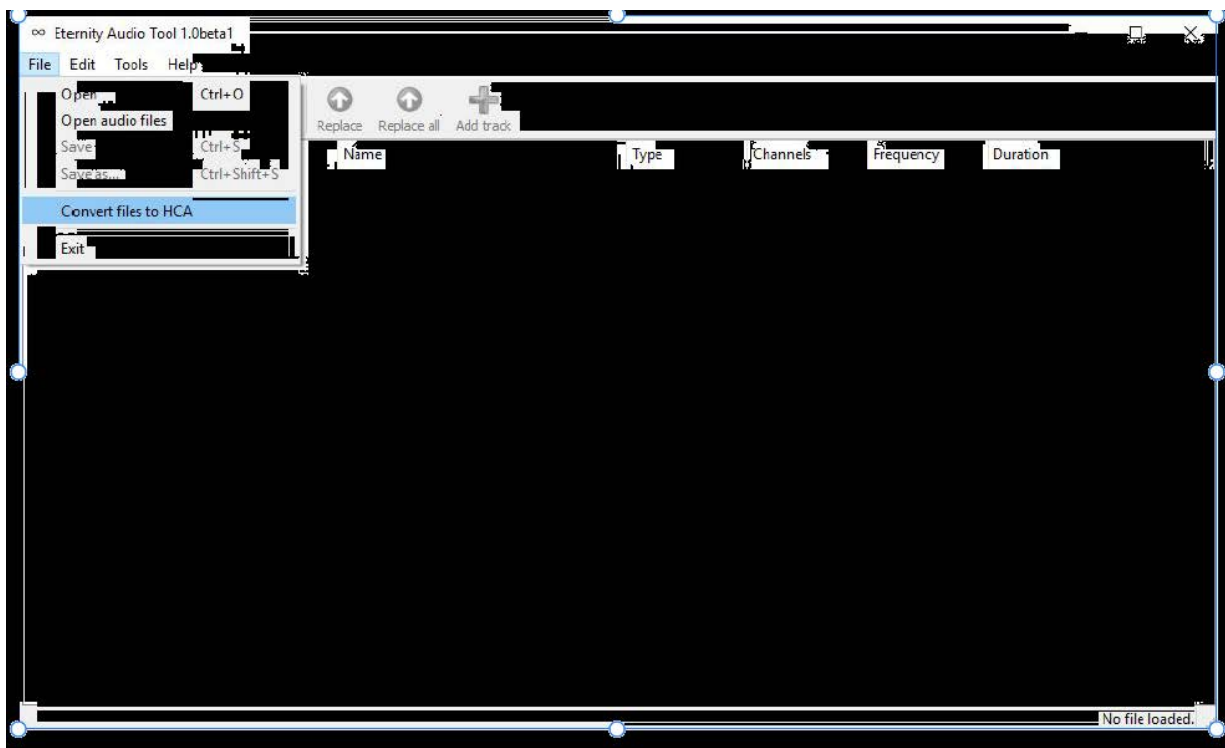
Step 3. Convert Your Music to .hca Format

Alright! We’re gonna take a break from the game’s contents for a bit and zone in on whatever music file you’d like to use first. There are a couple stipulations as to how this file should be in order for the game to load it properly:

- a. It's best practice to have your audio ready to be converted into .hca format by first saving it as a .wav file in 44100 Hertz (Although, MP3s should work as well)
- b. The file *cannot be larger in size than the original song*. (I'll show how to determine this in a moment)
- c. We will need the file to eventually be .hca format to work with MvCI

So let's get started! We'll assume you'll have a .wav file ready to use! (If not, Audacity can do that for ya!) We're going to take your .wav file and first convert it into .hca format and work from there.

Let's open up Eternity Audio Tool, we're going to not really "use" this program as much as *abuse* its HCA converting power. Go to File->Convert Files to HCA and simply find your .wav file you'll be using, and convert it. The program will save a copy of your song in the same folder the song is located in .HCA format.



Next we're going to compare the size of our converted music and compare it to the file you would like to swap out. (Use the unpacked files we made for this!)

All of the game's music is kept in MVCI/Content/Sound/BGM/, and from there it breaks down into Characters and Story folders as well. Use the chart below to find which character you would like to swap out music for

00A	Morrigan	005	Thanos
00B	Cpt. America	006	Ryu
00C	Strider	007	Captain Marvel
00D	Spider Man	008	Dante
00E	Chun-Li	009	Iron Man
00F	Hulk	010	Firebrand
001	Ultron	011	Thor
01A	Nemesis	012	Chris
01B	Hawkeye	013	Rocket Raccoon
01C	Spencer	014	Frank West
01D	Dormammu	015	Dr. Strange
01E	Arthur	016	Zero
002	Megaman X	017	Nova
003	Gamora	018	Haggar
004	Jedah	019	Ghost Rider

020 is Ultron Sigma, 021 is Ultron Omega

Once you know what file or character you'd like to swap out, we're going to go and look at that file's .uexp file. That file right there is where the original music is kept. Compare your converted .hca file to the game's .uexp file and make sure that your converted file is *smaller than the game's music file*.

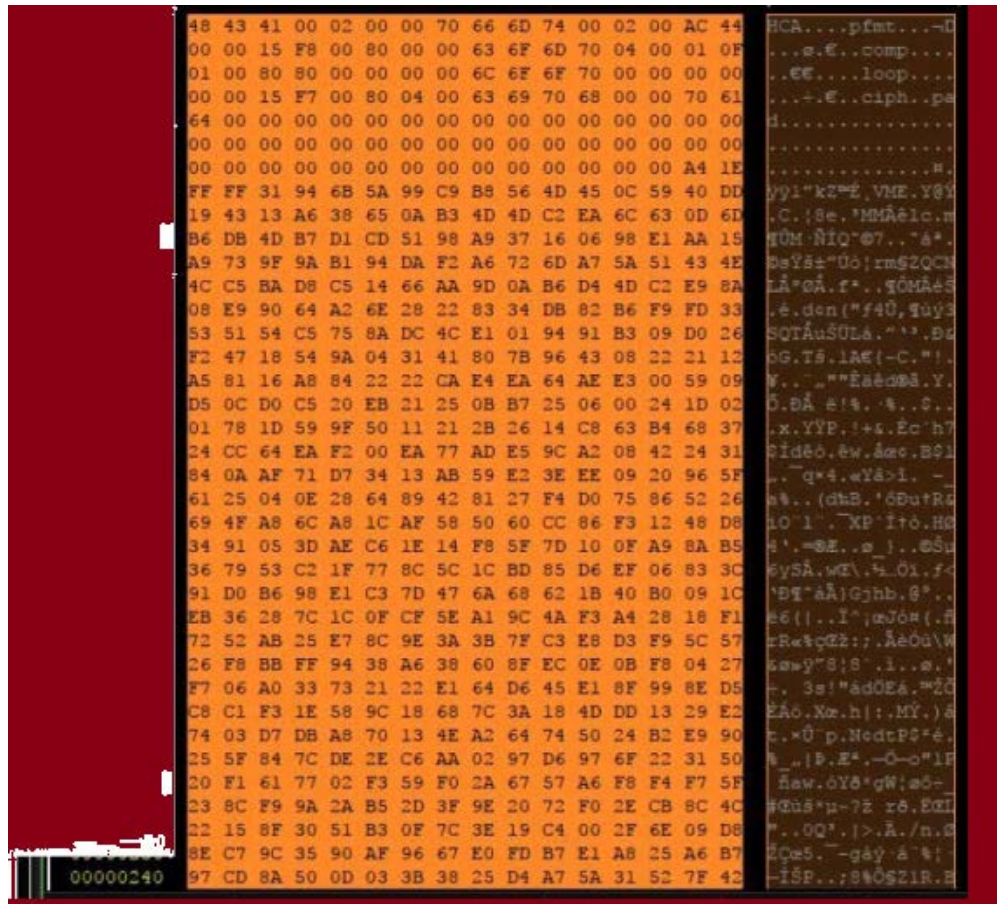
Vs.

In this case, our converted .hca file is smaller than the game's music, so this will work perfectly! This next step will be where we shove our music into the game's files, **so I recommend making a back-up of the .uexp file that we're modifying** just in case!!

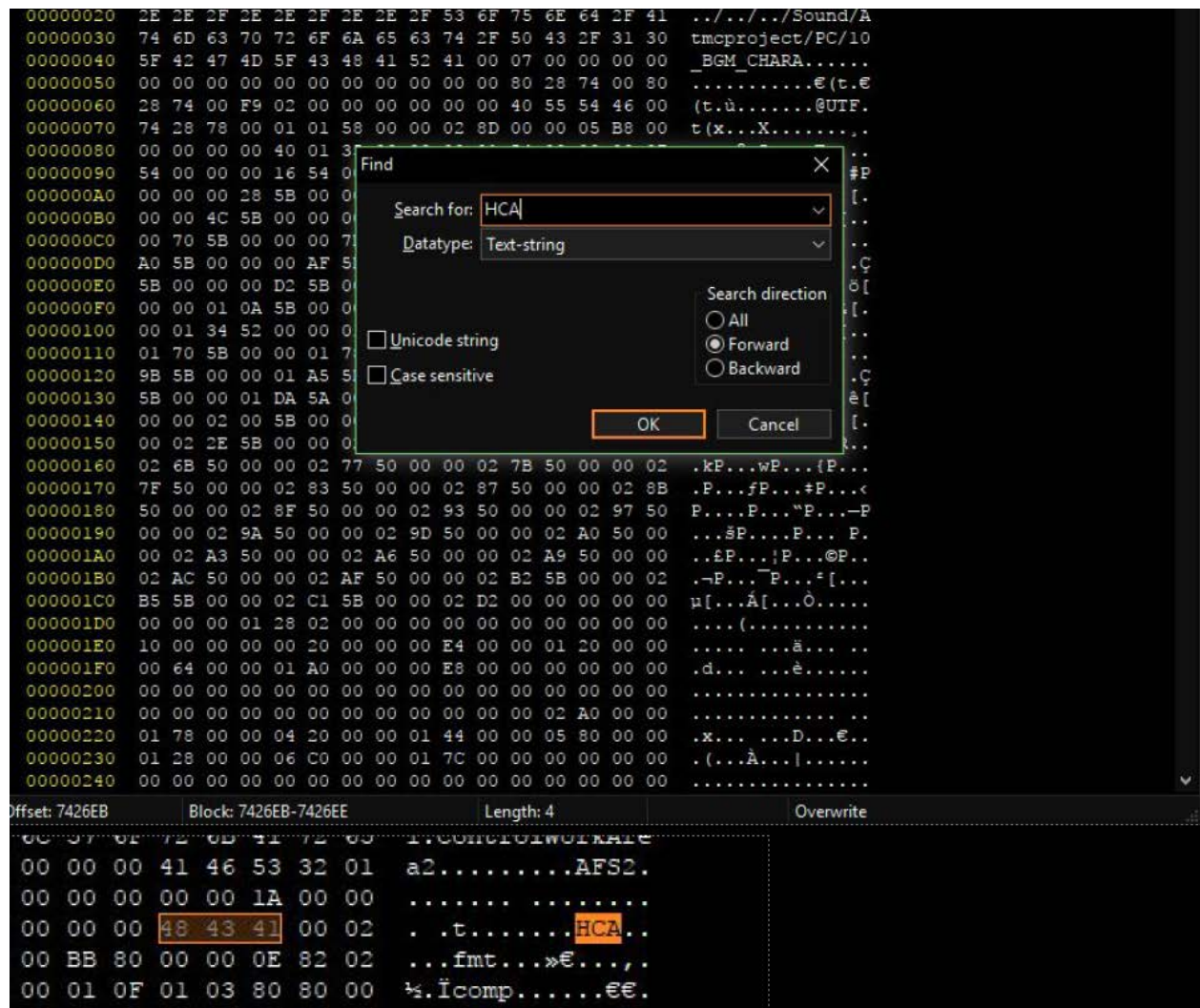
Step 4. Copy/Paste Hex Data into Game's Files

Alright, this is where things get a little interesting! I'll try and make this as easy as possible, so bear with me! This step, all we're doing is taking our *converted music data* and swapping it with the *original music's data*.

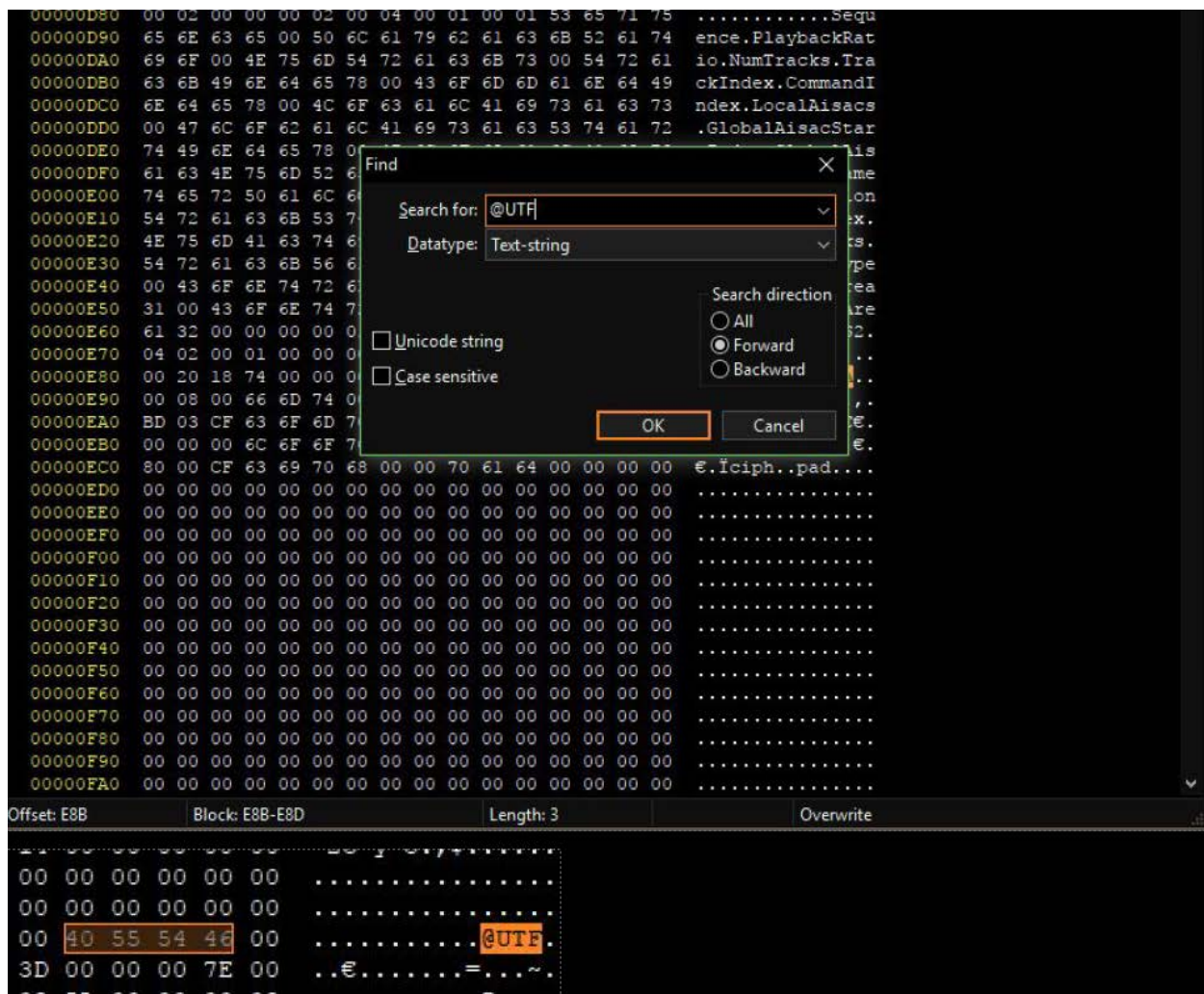
Find your converted .hca file we made earlier, right click, and Open With HxD (Or whichever program you use. For this tutorial, I'll be showing screenshots from HxD) All you need to do now is select all the data (CTRL+A) and copy everything.



Now, find your character's folder (or whatever music you're swapping) and open the .uexp file using HxD. This will contain a lot of data, but **all we need to find is the HCA audio** that is kept inside containing the original music file. To do this, Hit CTRL+F and Search for HCA



So this is where the music file starts! Keep this in mind by either jotting it down or remembering. Now we just need to find where it ends! To do that, hit CTRL+F again and search for @UTF



Excellent! Everything from “HCA” to just before “@UTF” is the music data! Select everything starting from HCA to before the @ symbol on @UTF. You can do that by clicking before the @ symbol, holding shift, and clicking before the H on HCA.

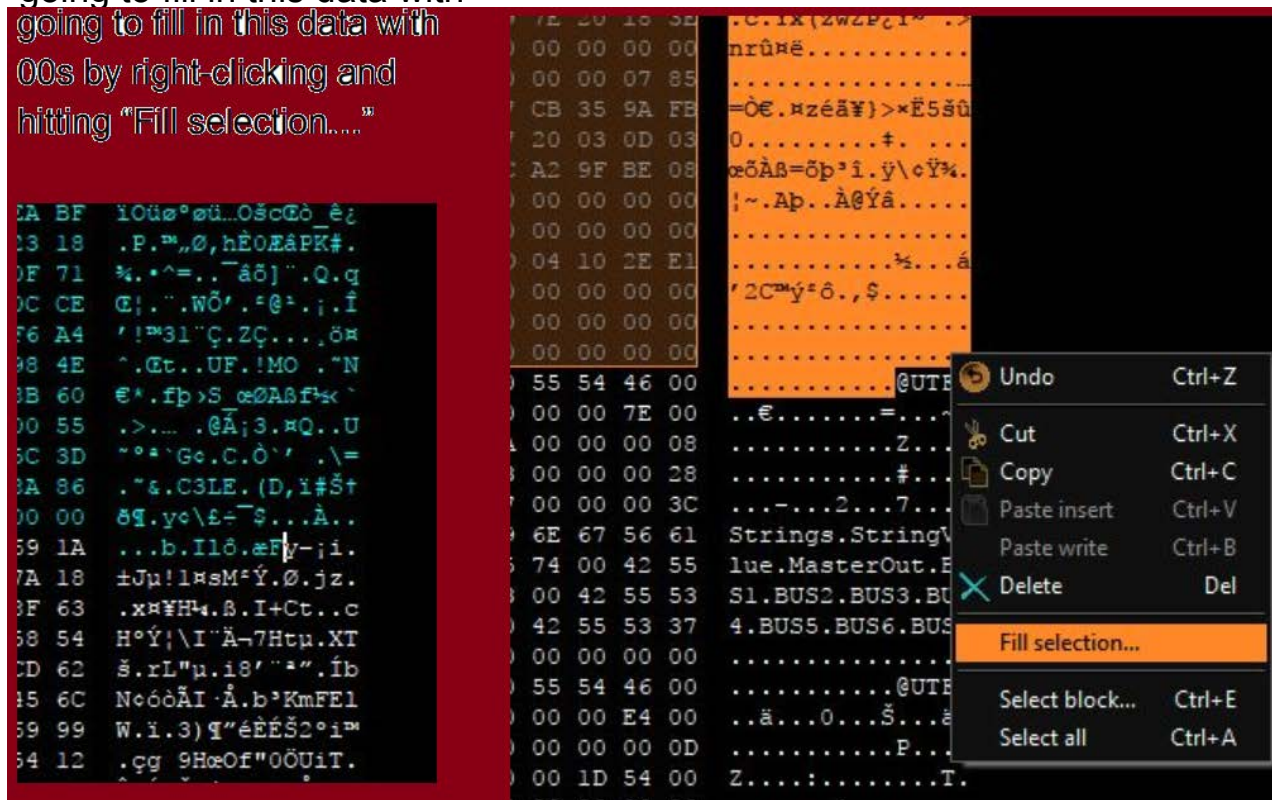


to

Now, using the copied .HCA data you copied earlier, hit Edit->Paste Write (or CTRL+B) to paste in the musical data! You'll see it overwrite the data that was there in bright red text! Excellent! But you'll probably notice some of the old data is still there however... All we need to do is find where OUR data ends and replace the rest of the old data up to the @UTF point. Do this by finding our end, finding @UTF again, and selecting all the data in between. (Again, you can select everything in-between points by hitting Shift+Left Click) Now, we're

going to fill in this data with

00s by right-clicking and hitting "Fill selection...."



Leave the “Fill selection” menu as filling everything in as 00 and click Okay. Now, all the data that was left over should be filled out as 00! And we’re pretty much done with this step now! Hit “Save” to save your hard work! Now, all we have to do is pack this guy up so that Marvel will read our mod!

Step 5. Packing and Testing!

Now, the moment of truth! Let’s pack up our mod and give it a shot!

We’ll be using u4pak.py to create an Unreal Engine-ready .pak file that MvCI can read, in which you will need Python 2.7 in order to run correctly.

First, in the Folder that u4pak.py is kept, you’re going to make a folder structure mimicking the files that the unpacked version of MvCI uses where we modified our file. So, in my case, if I’m modding Spiderman’s theme (00D), I would make a folder set that builds as follows:

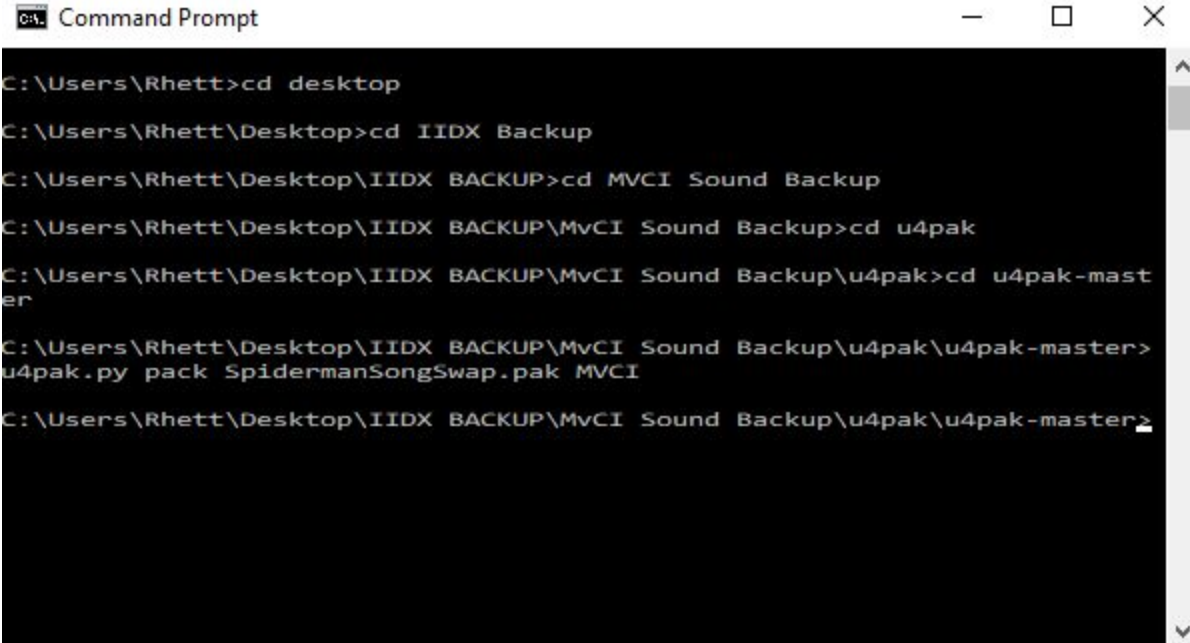
MVCI\Content\Sound\BGM\Chara\00D

And inside the 00D folder, I’m going to paste in the new BGM_CHARA_00D.uexp file that we just filled in with new Hex data.

Open your command prompt and tell it to go to the folder that u4pak.py is inside. Now, we’re going to tell u4pak to pack the MVCI folder structure by creating a line that follows this format:

```
U4pak.py pack nameofthemod.pak MVCI  
And then hit Enter to build the .pak
```


So, for naming conventions, if I wanted to say this was Spiderman's Theme, I would do something like the following:



```
Command Prompt
C:\Users\Rhett>cd desktop
C:\Users\Rhett\Desktop>cd IIDX Backup
C:\Users\Rhett\Desktop\IIDX BACKUP>cd MVCI Sound Backup
C:\Users\Rhett\Desktop\IIDX BACKUP\MvCI Sound Backup>cd u4pak
C:\Users\Rhett\Desktop\IIDX BACKUP\MvCI Sound Backup\u4pak>cd u4pak-master
C:\Users\Rhett\Desktop\IIDX BACKUP\MvCI Sound Backup\u4pak\u4pak-master>
u4pak.py pack SpidermanSongSwap.pak MVCI
C:\Users\Rhett\Desktop\IIDX BACKUP\MvCI Sound Backup\u4pak\u4pak-master>
```

Amazing! You should now see a new .pak file in the same folder that u4pak.py is kept in! Last step is to put it in the game and test it out!

Go to MvCI's install (the unpacked/Steam version) and go to the following folder: MVCI/Content/Paks and, if you don't have a folder named this already, create a new folder here called "~mods". Inside this folder, all you simply need to do is put your newly created .pak file in, and your game will make sure to load any .pak file inside it.

And you're done!

Give your new song mod a shot by going to the Collection inside the game and listening to your modified character's theme! It should be swapped out with whatever data you put inside of it! Enjoy!!!

Special thanks to SmashingVeteran on r/MahvelMods and the entire SFxT modding community. Without you all, I would have never figured this out. Sincerely, thank you all! - MasterWho