

Zipmod Basics with SB3UGS

Wogrim's Epic Guide to Making a Zipmod Using SB3UGS

Is There Another Way to Make a Zipmod?

Yes, the other way involves making your asset bundles with Unity (other steps are the same).

Why Make an Empty Zipmod?

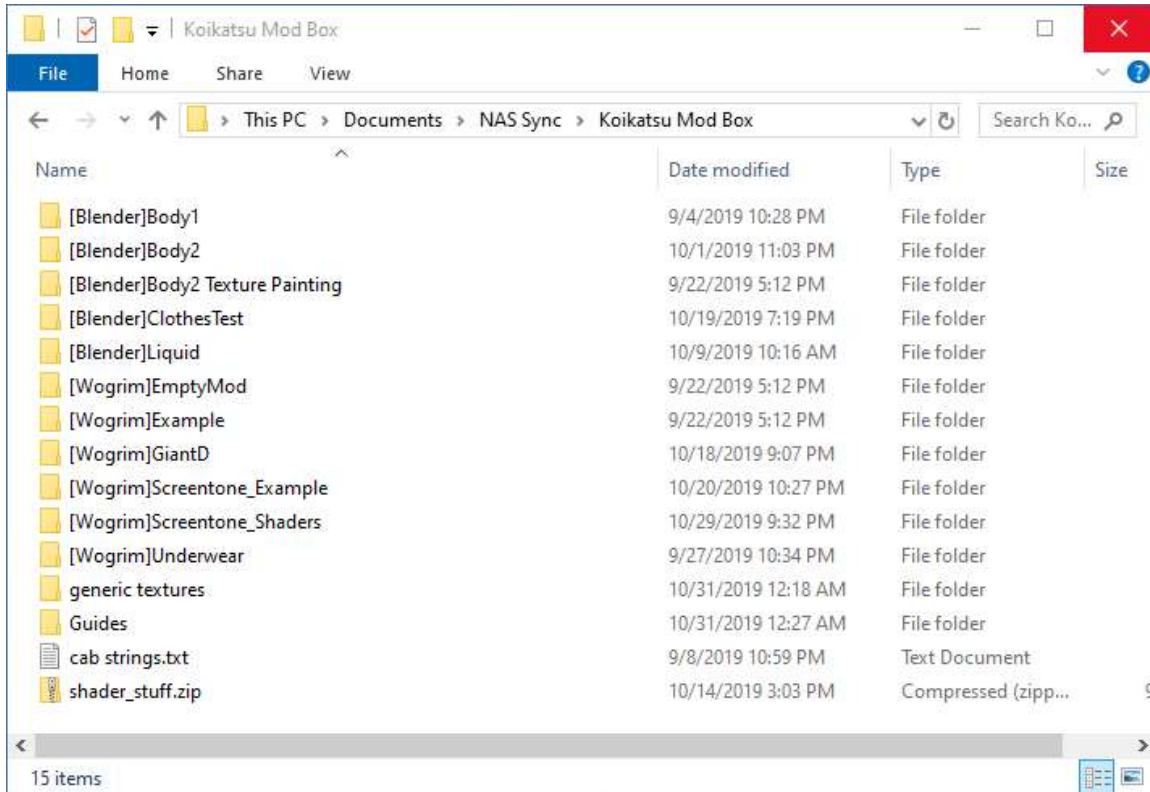
Before we make a working Zipmod, we will make an empty one. This will give you a folder that you just duplicate and modify whenever you want to make a new mod. It is faster than remaking everything from the beginning.

What Software is Needed To Follow This Guide?

- **Sideloader** - This is the plugin that loads zipmods. It is included in the HF Patch. I haven't looked into how to install it otherwise.
- **SB3UGS** - Get this from enimaroad's ~~github~~ gitea page. You need this for editing your copies of game files, and for looking at game files to see how your mod should work.
- **7-Zip** - This is used at the end to "zip" your mod's files with no compression. I don't know if there is other software that does this.
- **Text Editing Software** - Notepad is fine.
- **Table Editing Software** - Optional. I use Excel. It is to edit the list file, which is .csv format. Notepad can open this, but then it isn't organized into proper columns.
- **Image Editing Software** - Something to make thumbnails and textures. GIMP is free and works fine.

So I've Got the Software, What Now?

Make a folder somewhere on your computer for working on mods. I call mine "Koikatsu Mod Box". Inside of it, make a folder called "[YourName]EmptyMod". The naming doesn't really matter, but it is strongly recommended to be organized. Here is what my mod box currently looks like.



We will first create the manifest file, called **manifest.xml**. Create it and open with text editor. It can't be hiding in any subfolders or Sideloader can't find it. The only required field is "guid", which must be different from all other mods. For this reason, it is recommended that you put "YourName.ModName" to avoid conflicts. You should be able to copy-paste this and then change it.

```
<manifest schema-ver="1">
  <guid>Wogrim.EmptyMod</guid>
  <name>Empty Mod</name>
  <version>1.0</version>
  <author>Wogrim</author>
  <description>empty</description>
  <website></website>
  <game>Koikatsu</game>
</manifest>
```

Next we will need to make some folders, which are to emulate the folder structure of the game files. For this reason, I will explain important folders in the game files first.

How Are the Game Files Organized?

Go to your game install folder. There are 3 folders in here that are good to know about:

- **abdata** - Contains all the asset bundles (ABs), which are files that contain assets. Assets are basically any kind of textures, models, sounds, animations, or anything else the game needs to load. These are what we will make copies of and modify.
- **mods** - This is where the zipmod goes when we are done making it.
- **UserData** - This isn't important for this guide, but your screenshots, MaterialEditor exports, character cards, and game saves end up here.

Go into the **abdata** folder. There are 3 important folders in here:

- **chara** - Contains all of the game's models and textures for your characters. All skin textures, clothes, and accessories.
- **list** - Contains the lists that tell the game what types of things are in the other folders. We have to make a list file for any items we make to show up in the game.
- **studio** - I don't like studio because you don't get to interact with the girls, but this is where studio items are.

Go into the **chara** folder. All of the .unity3d files in here are asset bundles. The only folder in here is **thumb**, which contains assetbundles for thumbnails (which are used for selecting items in Maker). There are a couple extra things I will not explain, but for the most part these are organized as follows:

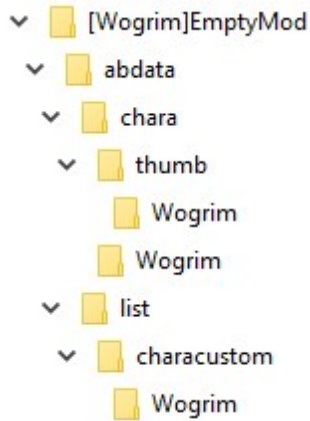
- starts with **ao_** - These are accessories.
- starts with **bo_** - These are hair types.
- starts with **co_** - These are clothes.
- starts with **mt_** - Some of these are texture-only items like eye textures, and some are textures shared by multiple items.
- ends with **_00** - These are the base game files; if it ends in a different number, it is part of a DLC. This number matches a list file, so it can help you find the right file.

Go into the **thumb** folder. If you are trying to find an in-game item, this is the easiest place to find its name, because the in-game name is different from the item's name in the asset bundle.

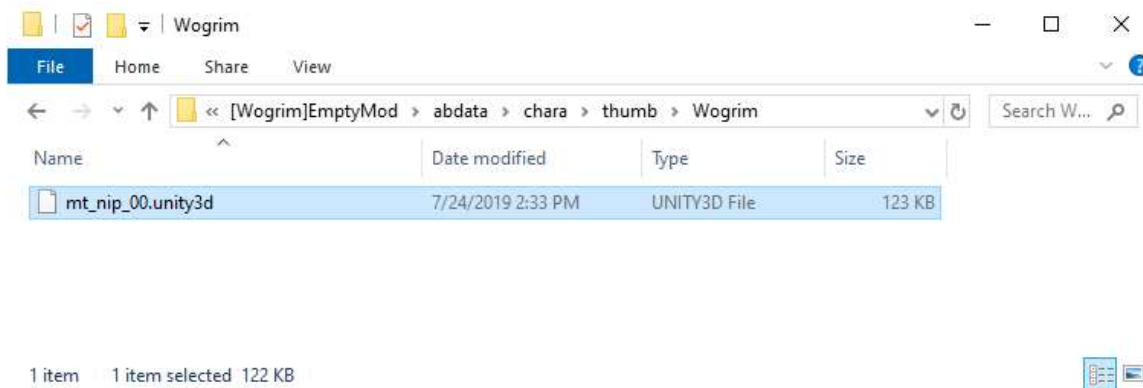
Go back up to the **abdata** folder and go into the **list** folder. From there, go into the **characustom** folder. These asset bundles contain all the lists, numbered like the asset bundles in the **chara** and **thumb** folders; when you need to see how an item type's list file should look, check out 00.unity3d.

Okay, Back to Making EmptyMod

For now, we will recreate the folder structure in our EmptyMod folder, starting with **abdata**. I add my name as a subfolder in all the folders because I don't want to worry about naming conflicts. You don't have to do that though.



We will start with a thumbnail file. Go back to the game files in **abdata/chara/thumb** and copy **mt_nip_00.unity3d** to the same folder in your EmptyMod folder, or the YourName subfolder if you created one.



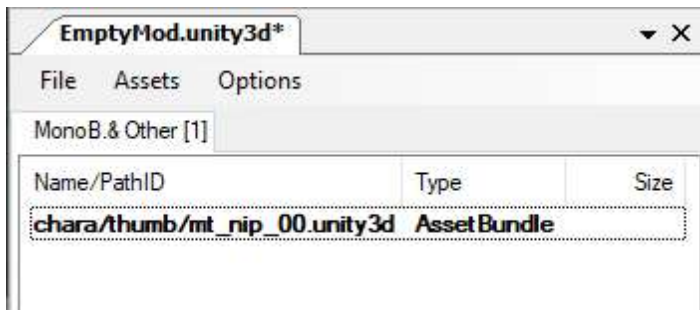
Rename the file to **EmptyMod.unity3d**. We will open this in SB3UGS.

How to Use SB3UGS?

Open SB3UGS. There are some important areas:

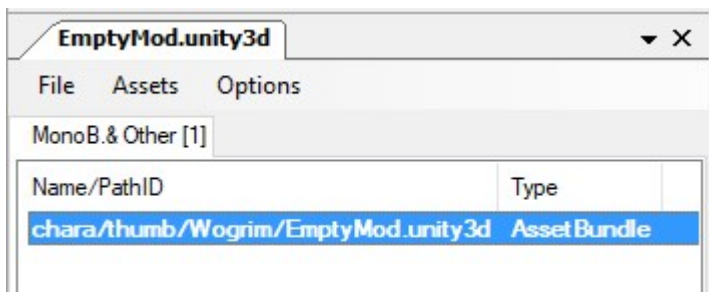
- **Files** - Any asset bundles that are open show up as tabs here. Drag your **EmptyMod.unity3d** into this area.
- **Editors** - If you double click on an asset it will generally show up here. Sometimes you will edit things, sometimes you will replace things, sometimes you will export things.
- **Renderer** - If you select one or more meshes you will get a preview here. It will change to an **Image** window if you have a Texture2D asset selected.
- **Log and Script** - Useful messages sometimes show up here.

We will now empty the asset bundle. Select all the Texture2D assets in the **Files** area and go "Assets=>Attention! =>Remove". Then do the same with all the Sprite assets. You can't delete the "AssetBundle" asset. It should now look like this picture.

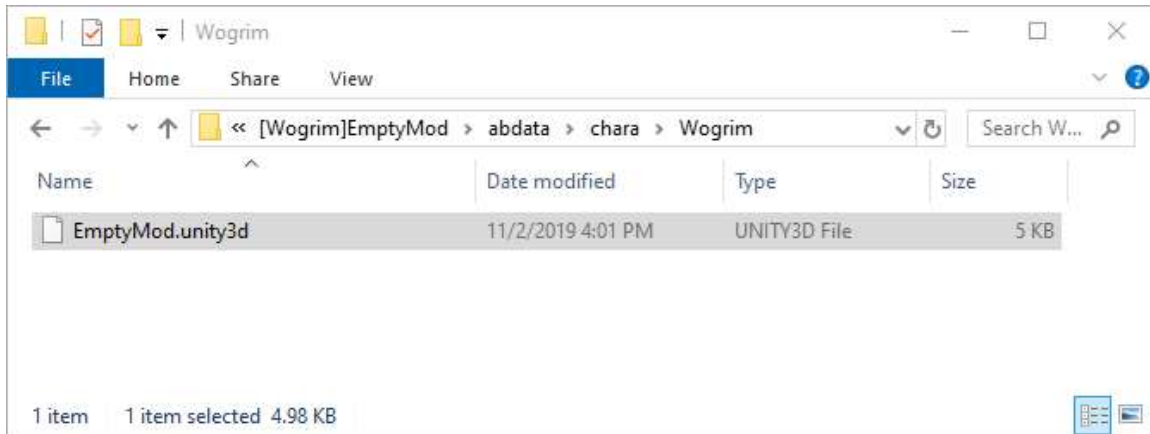


It is important to note that SB3UGS is not able to delete all types of assets. If you have an asset bundle with extra stuff you can't delete, you have to select the assets you do want, go to "Assets=>Mark for Copying", and in an empty asset bundle go to "Assets=>Paste All Marked".

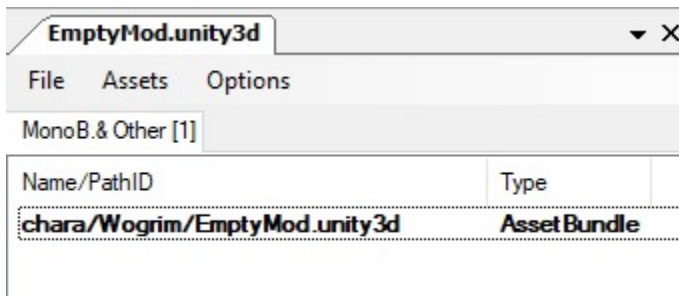
The asterisk means that the file has unsaved changes, so go to "File=>Save .unity3d". Now rename the AssetBundle asset by selecting it and going to "Assets=>Rename". Make it match the folder structure and file name. Save it again.



Now we can close it. Copy it to your **abdata/chara** folder or subfolder.



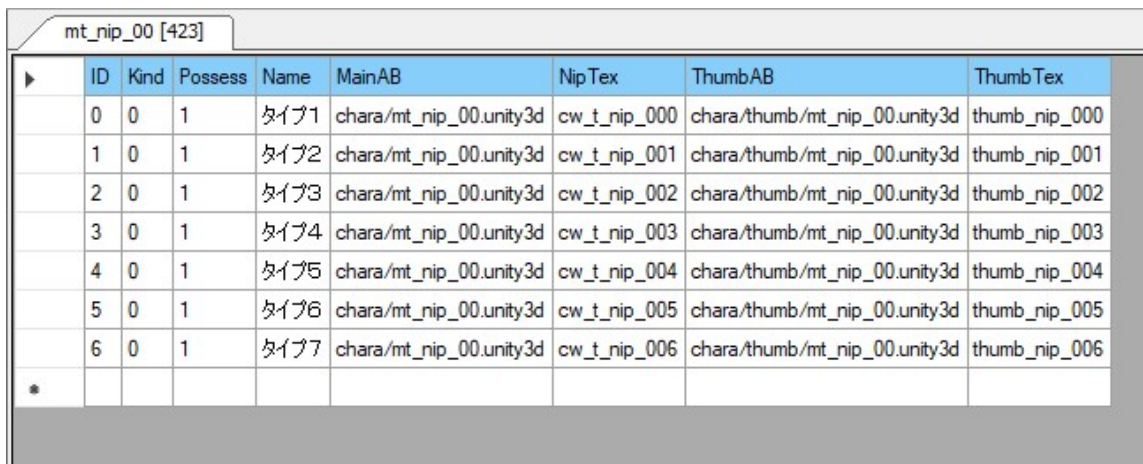
Open this in SB3U and rename the AssetBundle asset (the path is different). Save and then close it.



The List File

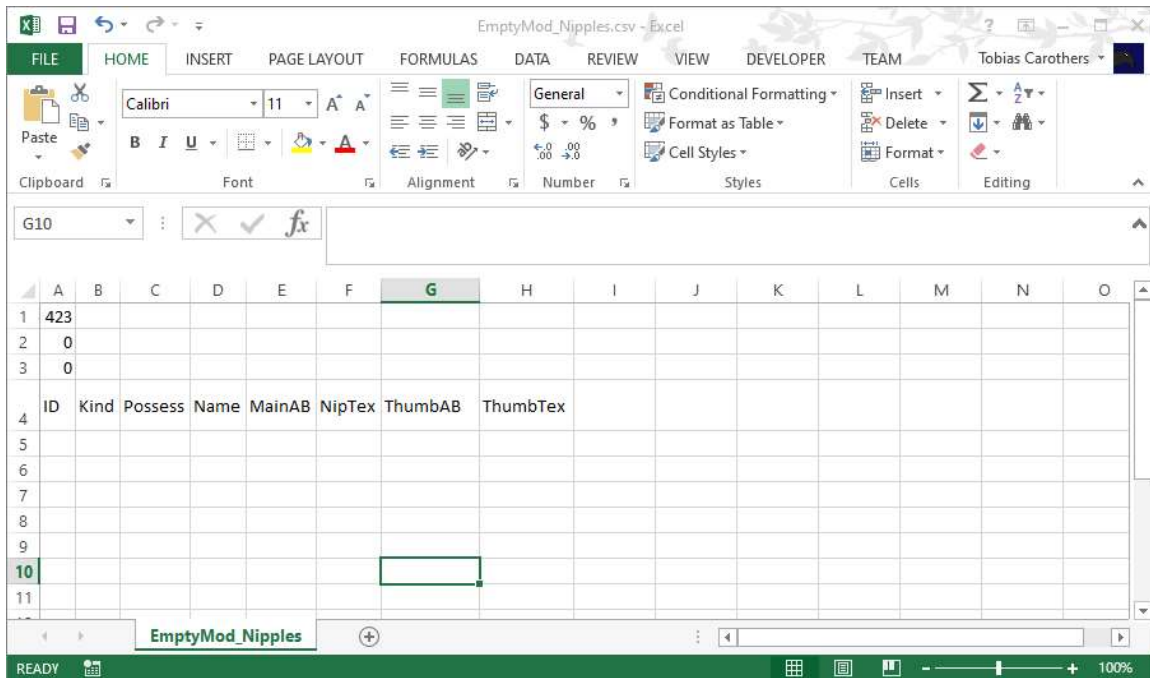
The last thing a zipmod normally has is a list file, which tells the game what types of items are in the asset bundles. We have no items in our EmptyMod, but we will make a list file as an example. The format for a list file depends on the type of item it is. If you are making a mod with multiple item types, you have to make a list file for each type. I will make a nipple texture list file for example, but it doesn't matter what type of list file you put in your EmptyMod. What matters is that you understand the process of how to create it.

Make a file in your EmptyMod's **abdata/list/characustom** folder (or subfolder) called **EmptyMod_Nipples.csv**. You can open with a text editor or a table editor. To find what a nipple texture list file should look like, we will look at the game's list files. Go the game's **abdata/list/characustom** folder and open **00.unity3d** in SB3UGS. Double click on the **mt_nip_00** asset and the Editor section will look like this.



ID	Kind	Possess	Name	MainAB	NipTex	ThumbAB	ThumbTex
0	0	1	タイプ1	chara/mt_nip_00.unity3d	cw_t_nip_000	chara/thumb/mt_nip_00.unity3d	thumb_nip_000
1	0	1	タイプ2	chara/mt_nip_00.unity3d	cw_t_nip_001	chara/thumb/mt_nip_00.unity3d	thumb_nip_001
2	0	1	タイプ3	chara/mt_nip_00.unity3d	cw_t_nip_002	chara/thumb/mt_nip_00.unity3d	thumb_nip_002
3	0	1	タイプ4	chara/mt_nip_00.unity3d	cw_t_nip_003	chara/thumb/mt_nip_00.unity3d	thumb_nip_003
4	0	1	タイプ5	chara/mt_nip_00.unity3d	cw_t_nip_004	chara/thumb/mt_nip_00.unity3d	thumb_nip_004
5	0	1	タイプ6	chara/mt_nip_00.unity3d	cw_t_nip_005	chara/thumb/mt_nip_00.unity3d	thumb_nip_005
6	0	1	タイプ7	chara/mt_nip_00.unity3d	cw_t_nip_006	chara/thumb/mt_nip_00.unity3d	thumb_nip_006
*							

I will describe how to fill in the list file as Excel, but I will show a picture after how it looks in a text editor. On the tab at the top, the number in brackets is the type of item. That goes in A1 in our list file. Just put 0 in A2. Just put 0 in A3. The row of A4 is where all column titles of the list file goes, so copy them from the game's list file. If our EmptyMod actually had nipple textures, we would fill in one row for each of them, but we will leave it empty because our asset bundles are empty. Here is the result.

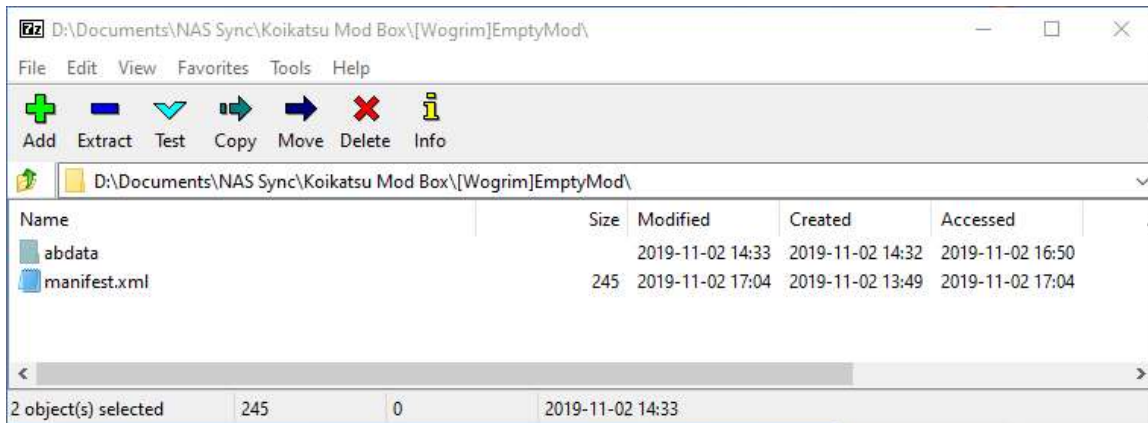


Unfortunately Excel sometimes acts weird, so you should open it in a text editor when you're done to make sure there is no strange formatting. A newline is a new row, and commas separate columns. So we want it to look like this picture.

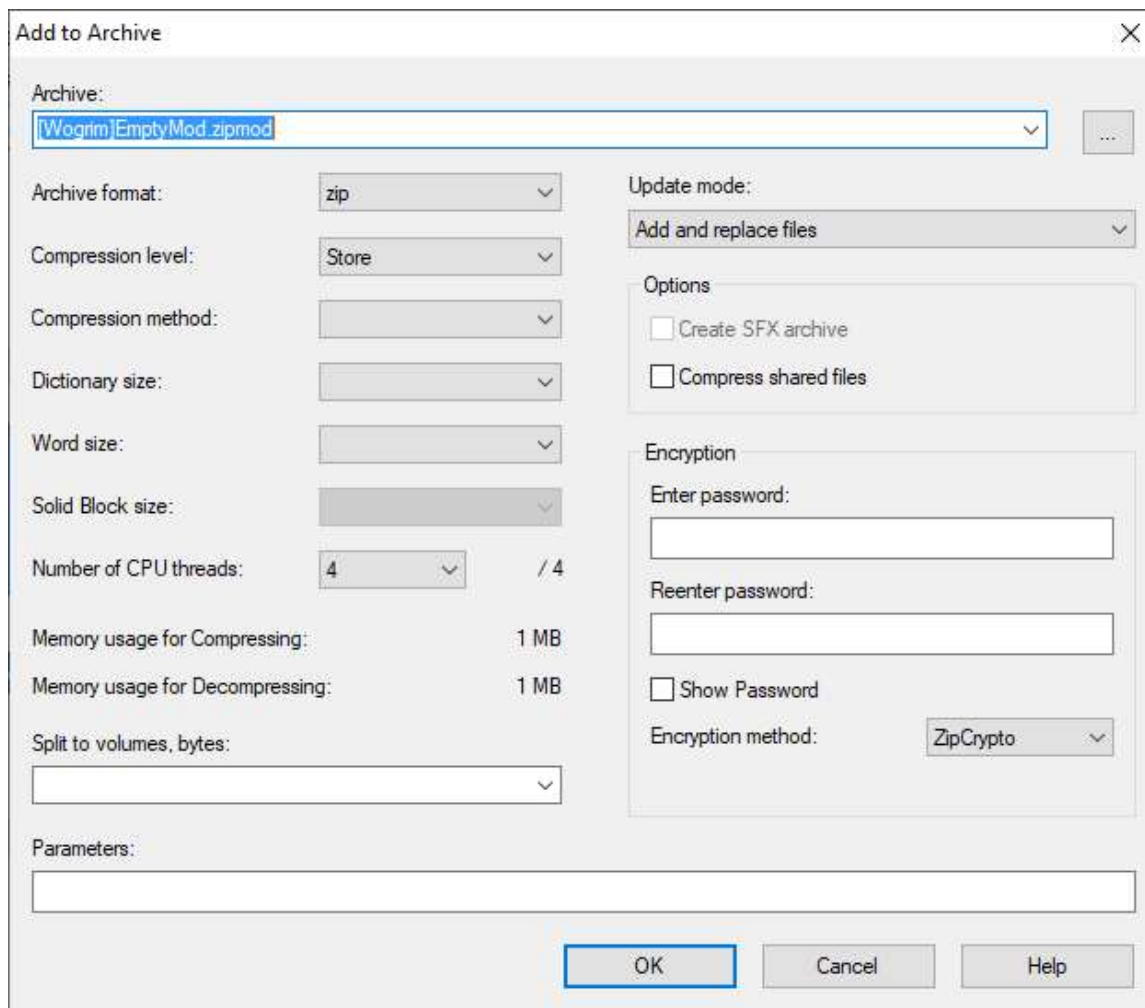


How to Make It a Zipmod?

Now we can zip our EmptyMod. Not that it does us any good, but I'm just showing the technique here so I don't have to explain it again later. Open 7-Zip and navigate to your EmptyMod's root folder. Select the **abdata** folder and the **manifest.xml** file and click "Add".



Make sure Archive format is "zip", Compression level is "Store", then change the extension to **.zipmod** and press OK.



Congratulations! You now have a Zipmod that doesn't do anything! I know it seems like a lot of work to do nothing, but making mods from the EmptyMod is easy.

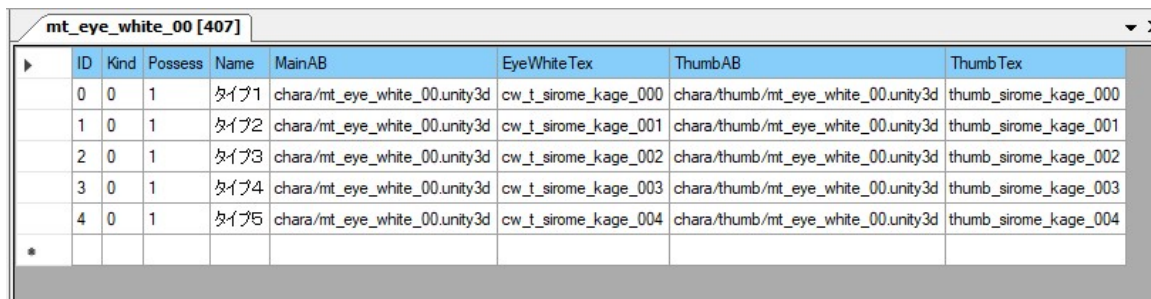
You can delete the .zipmod, it is just a waste of space.

How to Make the Empty Zipmod Into an Actual Texture Mod?

Duplicate your whole EmptyMod folder. I'm going to change things up and make an Eye Whites mod. Start by changing all of these:

- mod folder name
- manifest guid and mod name
- assetbundle file names
- assetbundle "AssetBundle" asset names
- list file name (add more list files if the mod will have more than one item type)

If you don't know what assets your item type needs, check the list file. In this case, an Eye White has a texture and a thumbnail texture.



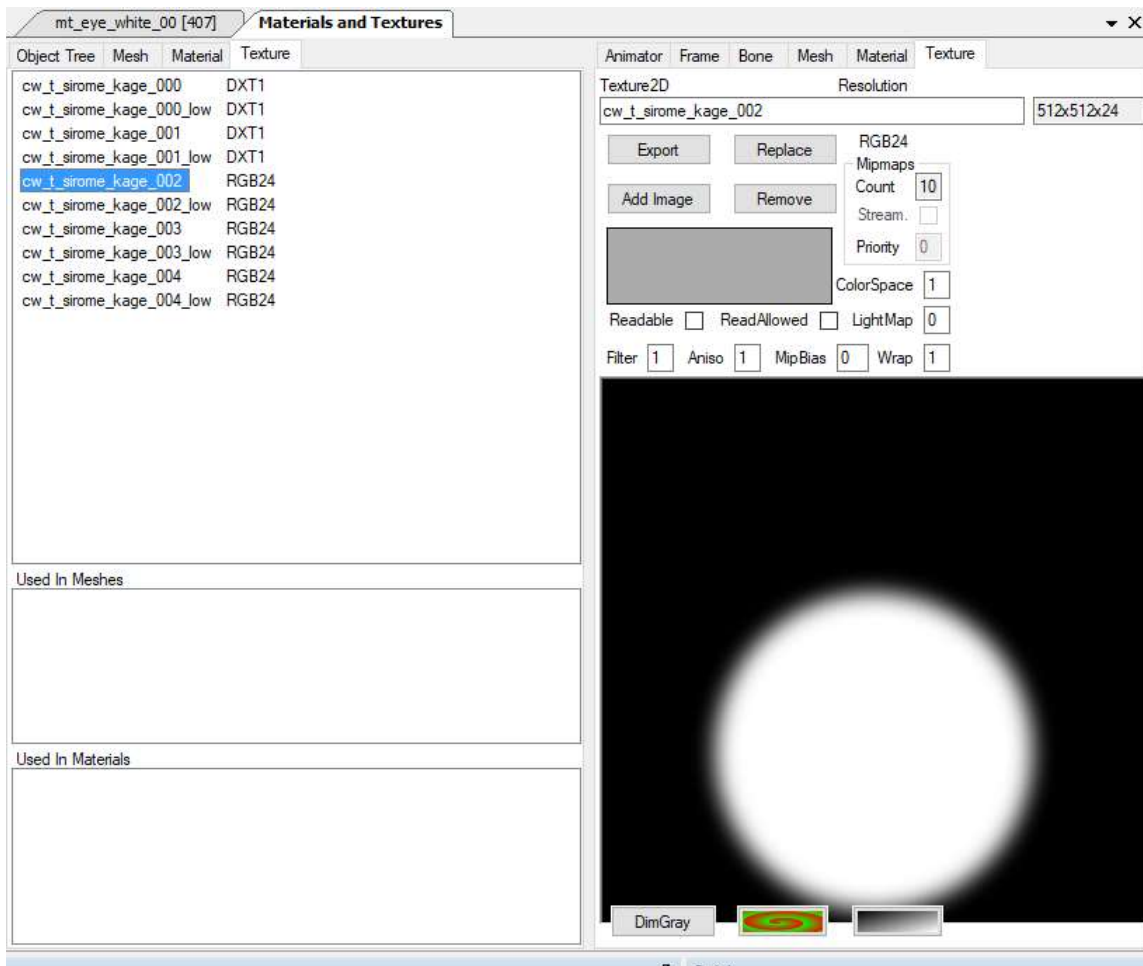
The screenshot shows a window titled "mt_eye_white_00 [407]" with a table of assets. The table has columns: ID, Kind, Possess, Name, MainAB, EyeWhiteTex, ThumbAB, and ThumbTex. There are five rows of data, each representing a different type of eye white (タイプ1 through タイプ5). Each row shows the asset bundle path for the main texture, the eye white texture, the thumbnail asset bundle, and the thumbnail texture.

ID	Kind	Possess	Name	MainAB	EyeWhiteTex	ThumbAB	ThumbTex
0	0	1	タイプ1	chara/mt_eye_white_00.unity3d	cw_t_sirome_kage_000	chara/thumb/mt_eye_white_00.unity3d	thumb_sirome_kage_000
1	0	1	タイプ2	chara/mt_eye_white_00.unity3d	cw_t_sirome_kage_001	chara/thumb/mt_eye_white_00.unity3d	thumb_sirome_kage_001
2	0	1	タイプ3	chara/mt_eye_white_00.unity3d	cw_t_sirome_kage_002	chara/thumb/mt_eye_white_00.unity3d	thumb_sirome_kage_002
3	0	1	タイプ4	chara/mt_eye_white_00.unity3d	cw_t_sirome_kage_003	chara/thumb/mt_eye_white_00.unity3d	thumb_sirome_kage_003
4	0	1	タイプ5	chara/mt_eye_white_00.unity3d	cw_t_sirome_kage_004	chara/thumb/mt_eye_white_00.unity3d	thumb_sirome_kage_004

While we are looking at the list file, we might as well copy the necessary info to our own list file.

For non-texture items, there will usually be more columns, and some of the items in those columns can reference extra things that are not part of the list file. So as good practice we'll go look at one of these eye whites.

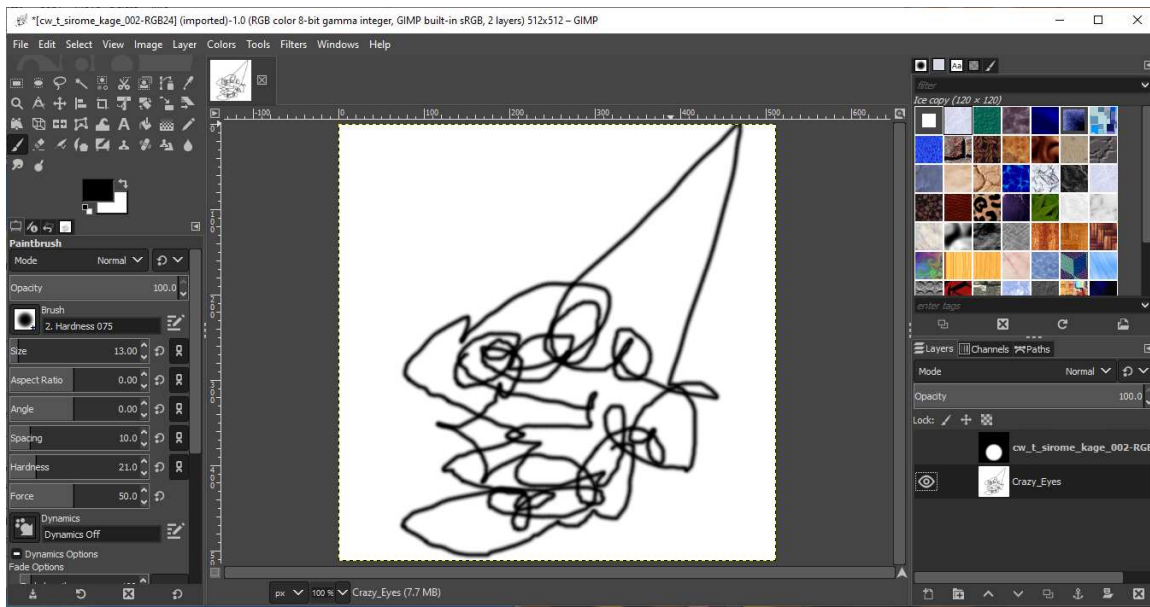
Open the game's asset bundle for the eye whites, **abdata/chara/mt_eye_white_00.unity3d** (as noted in the MainAB column). You should notice that there is a "_low" version of each of them, which is used when roaming in story mode. We must make a _low version of our asset so that our mod works for that mode. For textures, this just means cut the resolution in half (height and width). We will use an existing game asset as reference.



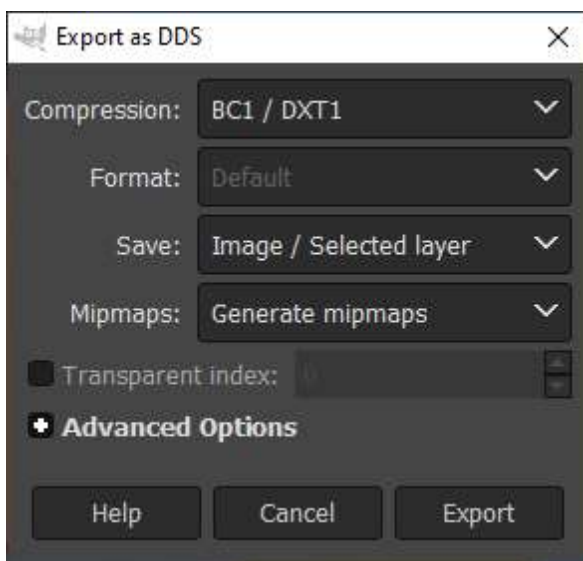
We generally want to match compression, texture resolution, color space, and mipmaps for what we make. I don't know what SB3UGS calls "RGB24" for the file format, but a couple of the eye whites are "DXT1" so we'll use that. We will go with 512x512 for texture resolution and 256x256 for the low version. I export the texture to use as reference for making my own texture. This puts it in the game folder, so move it somewhere else so you're not making a mess of your game folder.

As for how exactly the texture works, it can vary greatly from item to item. You can ask around or experiment to find out. In this case, eye whites let you select 2 colors, with the white part of the texture normally being the white part of the eye, and the black part typically being used as a shadowed area along the upper edge of the eye white. I'd rather make something more interesting though.

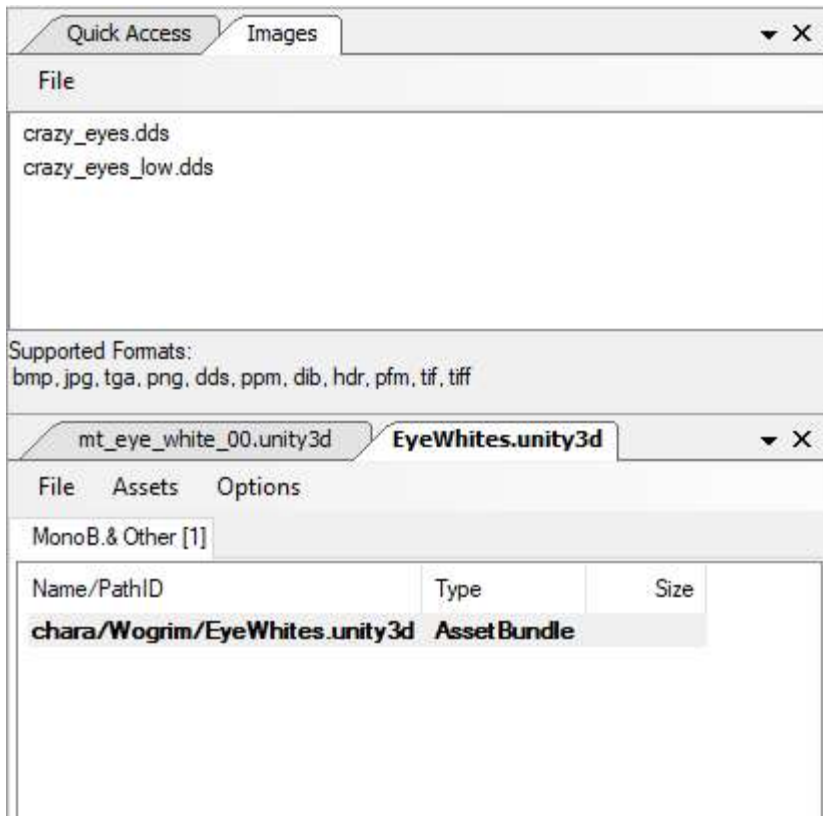
Open the exported texture in GIMP. Create a new layer to draw on, and use the original texture as reference for size/location. It is usually better to start out rough so that you make sure you put something in the right place before you spend a lot of time on it. I just do a quick scribble.



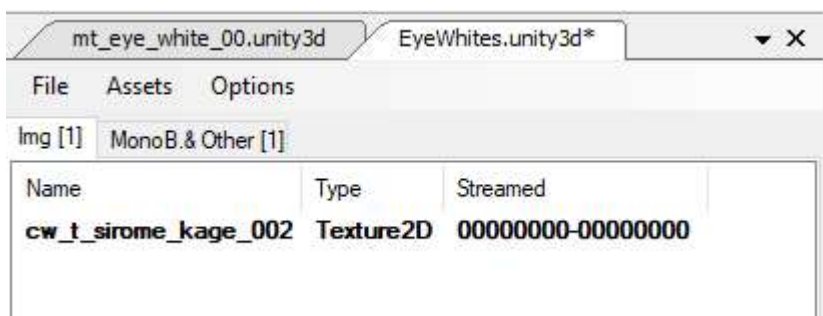
Save as a GIMP file, and then export as a .dds. In this case we're using DXT1 compression and mipmaps based on what we saw in the original game files.



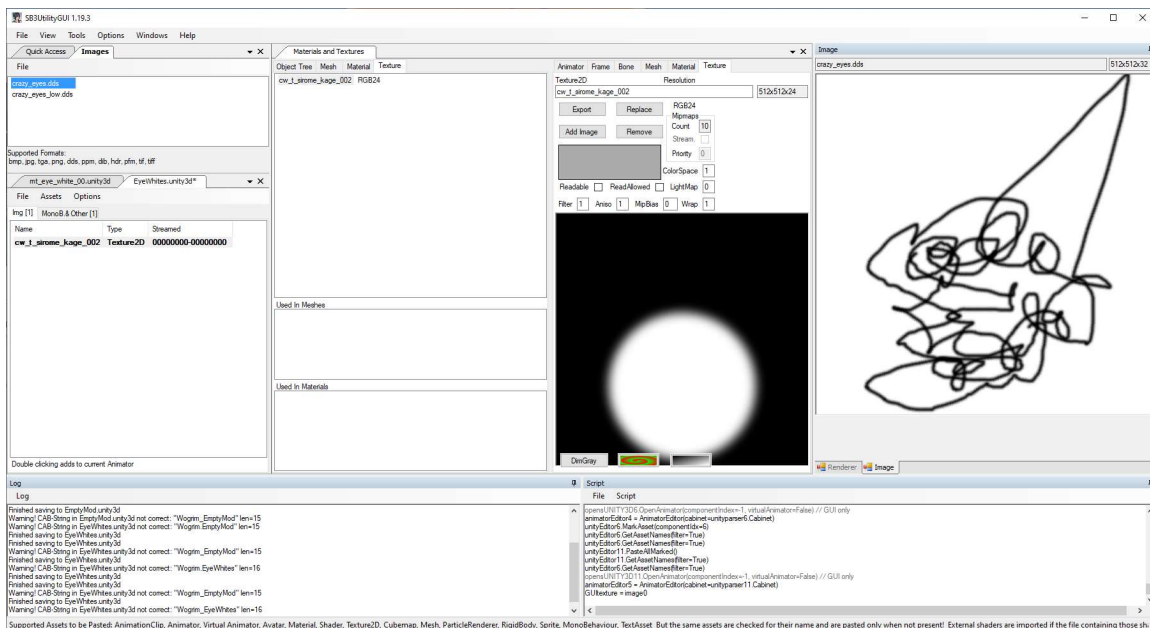
Then scale the image down as we determined earlier and export the _low version. Now to put these in our asset bundle. Open the asset bundle in SB3UGS and drag your new texture images into SB3UGS. An "Images" tab should show up. You can drag it somewhere else so it is not in the way.



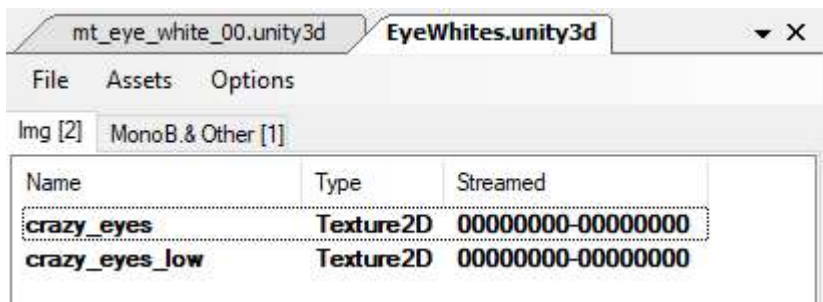
We want our images as a Texture2D in our asset bundle, but we unfortunately can't just drag them in. So, go back to the game's asset bundle and select one to copy. "Assets=>Mark for Copying" then go to your asset bundle and "Assets=>Paste All Marked". You'll get a window verifying what you want to paste. Your asset bundle will now have a texture in it.



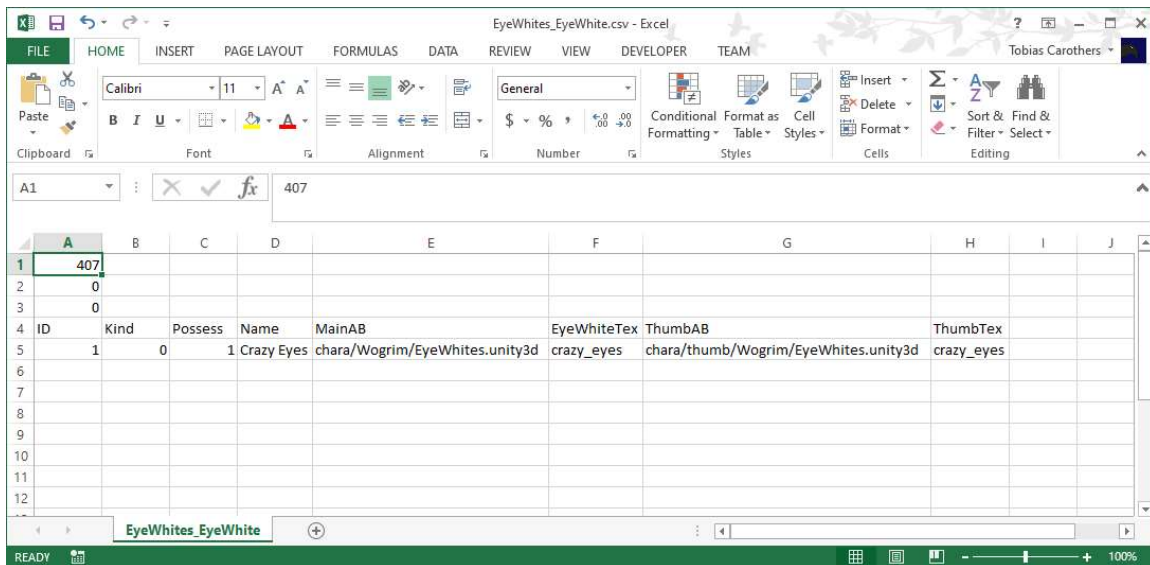
Double click the texture so the editor tab opens for it, and then select the texture you drew from the Images tab so that your texture appears on the far right.



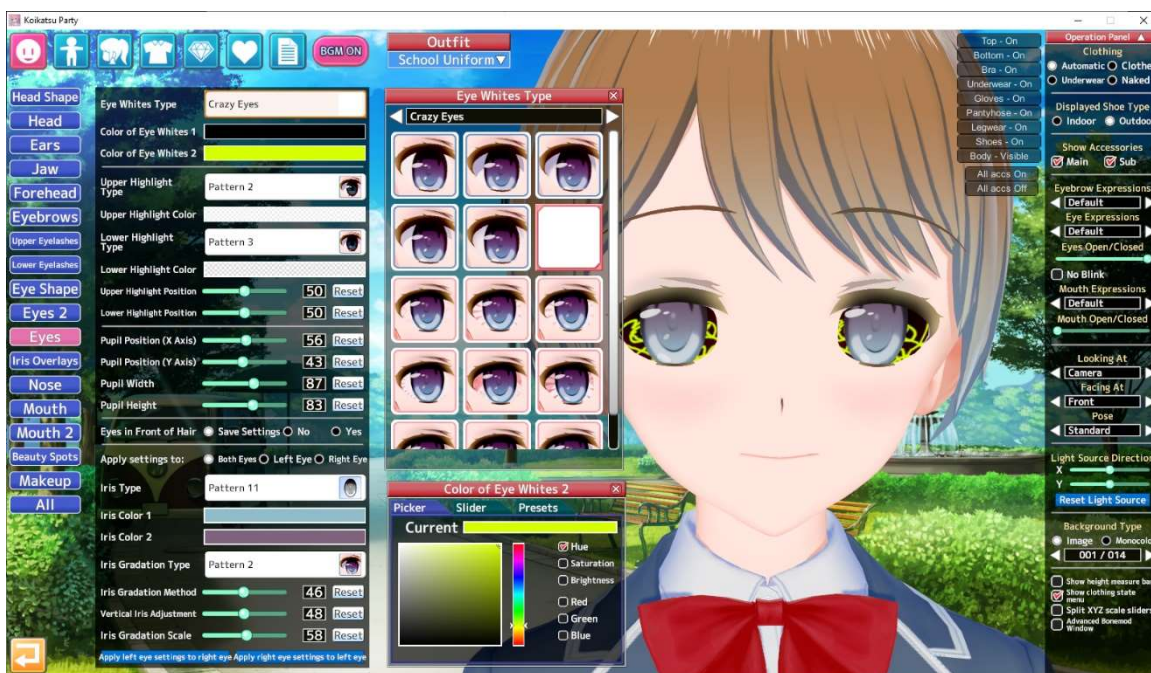
You can now use the "Replace" button to replace it with your texture in the asset bundle. Then select your "_low" texture on the Images tab and use the "Add Image" button. Save your asset bundle. Rename your assets and make sure they are named the same, but with the low-resolution one having "_low" at the end. Save your asset bundle again. You should now have both assets in your asset bundle. Double check that you changed the CAB string.



So, back to the list file. Use the game's list file to help you fill in the columns, just remember to put your file names in the right spots. For each item in the list file, ID must be different, so start with 1 and go up if you have more items. We have not yet created a thumbnail texture, but it won't break the game. It just shows a white square if it can't find your thumbnail.



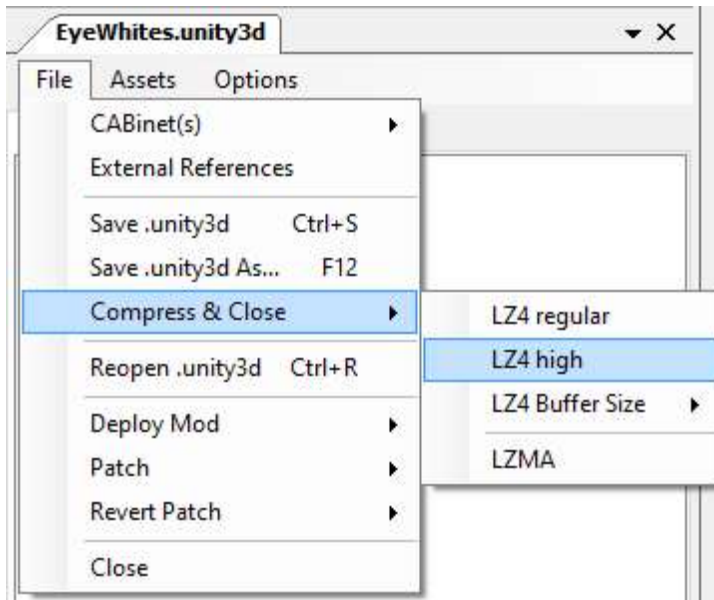
Now use 7-Zip to make the zipmod like we did with the EmptyMod, and put it in your mods folder. You should have results, but if you don't, check the troubleshooting steps at the end of this guide. I changed the eye whites off the default colors to make my scribbles look more interesting.



The only thing left to do is to make a thumbnail, which I won't cover in this guide because it's pretty much the same as you just did (you don't need a "_low" thumbnail though). Take a good screenshot of your item and put a border/background if you want. Be sure to check format and image resolution of existing thumbnails and make yours match, and make sure its name matches whatever you put in the list file.

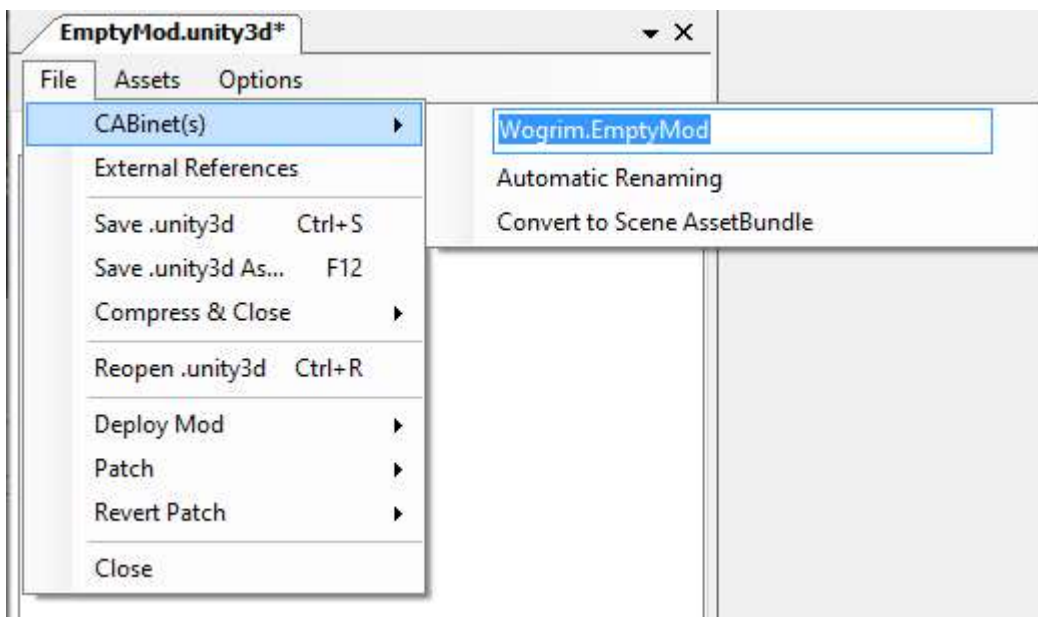
Compressing the Asset Bundles

If you make a mod you want to release, there is one more step: you should compress your ABs (before the zip step) to make the zipmod smaller. Open each AB in SB3UGS and go to "File => Compress & Close => LZ4 high". In my case, this AB goes from 219 KB to 45 KB.



CAB Strings

Each AB has something called a CAB string, which must be unique for each AB. Default is an automatic renaming which should prevent conflicts, but if you have problems, it is done under "File=>CABinet(s)=>". If you think this is the issue, don't forget to change the one on the thumb AB as well (to something slightly different).



How to Make Something More Than a Texture Item?

There are guides for some item types, and a lot of knowledge if you ask in **#mod-modelling**. Start with easier items and work your way up, you will learn important things so you can be successful with harder items.

How to Solve Common Zipmod Problems?

I can't find any of my items to select!

- There may be a conflict with the GUID in your manifest (or you messed up the formatting part)
- You may have zipped your mod wrong (if you extract it, manifest should be in root folder)
- You may have put your list file(s) in the wrong location (must be in "abdata/list/characustom" or subfolder)
- Your list file is super messed up (check it very carefully)

I can select my item but nothing shows up!

- You may have mistyped an asset bundle path or asset name in the list file
- There may be some kind of problem with your textures
- 3D items: it may be so small or so large you can't see it, or something with the shader
- There may be a CAB string conflict on your asset bundle (unlikely); go up one page