

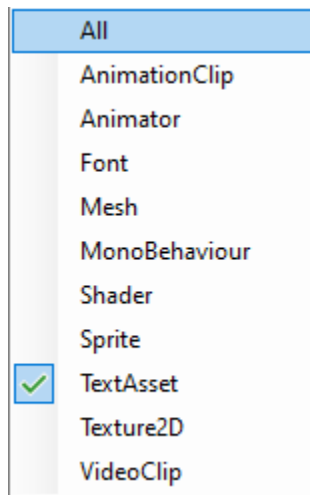
A guide on how to modify story dialogue for Limbus, by May

For any inquiries/questions dm goldenpineapple on discord I'm always glad to answer any questions about modding

Part 1: The Basics (Locating the files to edit)

For Information on how to use the tools listed here, there is guides in the Limbus Modding Discord (<https://discord.gg/pwcMw24U>)

Open AssetStudio, once it's loaded, sort files by specifically TextAsset files.



For the story, there are two different files which each contain separate parts of the dialogue, there are the dialogue files (Each has a language, either EN_, KR_, or JP_ as a prefix followed by the stage ID*), and the setup files, which have the same stage ID as the dialogue files, but with no language prefix.

Example dialogue file

EN_S523A

Example matching setup file

S523A

* You can break down the stage IDs into 4 parts



The first letter (Underlined in red): This can either be an E (Intervallo), an S (story), an X (side chapter), a P (uptie story) or a D (being the second character, indicating being a dungeon, will come after the canto number instead of before it)

The second character (underlined in yellow): Represents the canto that the stage is in (in terms of dungeons as stated before, comes before the D, instead of after like all the other examples)

The next two characters (underlined in blue): These show the dialogue number (Does not match up with the node number or the dialogue number in the theater, it starts at 1 and increments up from there with each individual story scene (Before and After battle scenes are counted under the same value))

The last letter (underlined in green): This can either be a B or an A, representing either a Before battle scene (B) or an After battle scene (A)

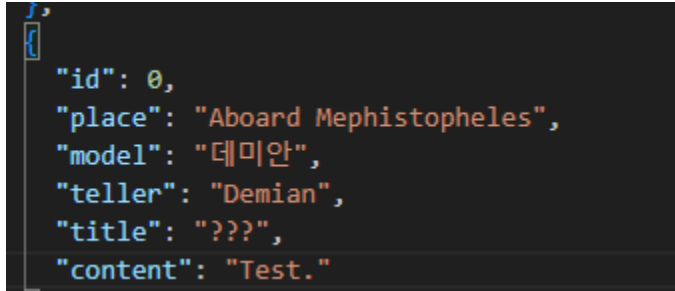
Extracting Text Files

To extract the text files, refer to the UABEA tutorial in the modding discord (<https://discord.gg/pwcMw24U>), you can import/export it through the same method as for image asset files.

Part 2: Basics of Changing Dialogue

Once you've extracted the dialogue file, open it up with any kind of text editor. I prefer using either Notepad++ or Visual Studio Code.

Once you have it open in a text editor, we'll go through how the dialogue files are organized



```
{
  "id": 0,
  "place": "Aboard Mephistopheles",
  "model": "데미안",
  "teller": "Demian",
  "title": "???",
  "content": "Test."
}
```

These are the 6 different categories of things that you can put into dialogue files

- "Id" being the id of the line, matching up with the corresponding id in the setup file
- "Place" being the location in the top left of the screen (after being set, stays until changed)
- "model" being the Korean name of the speaker ([all listed in the index](#), located in the farthest left column), this decides the color of the name in the dialogue box
- "Teller" being the name of the character speaking (Not needed for any of the sinners, Charon, or Vergilius)
- "Title" being the title above the character's name in the dialogue box (also not needed for any of the sinners, Charon, or Vergilius)
- "Content" being the actual dialogue that is said

You can remove model from this entirely if you'd like to have it be an internally spoken line, and not said by any character

No lines besides the id and content are required

Part 3: Changing Setup Files

3.1 Characters showing up in positions

To start with, you can add new characters to a scene in the setup file

You can do this through “characterlist”: parameter,
An example of this parameter filled in would be something like this

```
"id":0,  
"characterlist": "카론:1"
```

Using the Korean name for the character (in this case, Charon), with an optional colon before a number to determine the position on the screen (defaults to the middle if not specified)

Positions on the screen go from 1-10, from left to right, with -1 being used to put characters out of the scene.

Note: Characters will stay on the screen permanently after you use them, you must set them to -1 if you want them off the screen, (you can do ALL: -1 under characterlist if you'd like to remove all characters from the screen)

To add multiple characters on the screen at once, put both Korean names within the same quotation marks, separated with a comma, make sure both characters are in different positions.

```
"id":0,  
"characterlist": "카론:1, 그레고르:2"
```

3.2 Changing character emotions/using emoticon bubbles

(A list of all character emotions/emoticons can be found in the story index)

Character emotions can be changed through the “feeling”: parameter

While the emoticons can be used through the “emotion”: parameter

Not all characters have emotions, you can find which characters have them through the index

```
{  
  "id":0,  
  "characterlist": "홍루:1",  
  "feeling": "surprise",  
  "emotion": "Flower"
```

This is an example of Hong Lu being in position 1, with the “surprise” feeling, and the Flower emoticon

Important note

Emotions and feelings WILL NOT work if the character is not actively speaking a line, so the id in the setup file must be matched up to the same id of the character speaking in the dialogue file, so for the previous image to actually display the emotion on Hong Lu’s face, he needs to be set as the model for the dialogue in id: 0 in the dialogue file

```
{  
  "id": 0,  
  "place": "Aboard Mephistopheles",  
  "model": "홍루",  
  "content": "Test."
```



This is how it looks when packaged and imported into the game while setup correctly

3.3 Changing Backgrounds, CGs, SFX, and BGM

(All BGM, CGs, BGs, and SFX are listed and organized in the index)

Backgrounds can be changed through the “bg”: parameter

CG's can be changed through the “cg”: parameter

SFX can be played through the “soundeffect”: parameter

BGM can be played through the “bgm”: parameter

Backgrounds are the most simple, they determine what the background is, and only one can be on screen at a time, to set a bg, you would do something like this

```
{  
  "id":0,  
  "characterlist": "홍루:1",  
  "feeling": "surprise",  
  "emotion": "Flower",  
  "bg": "story_D_backsteat"  
}
```

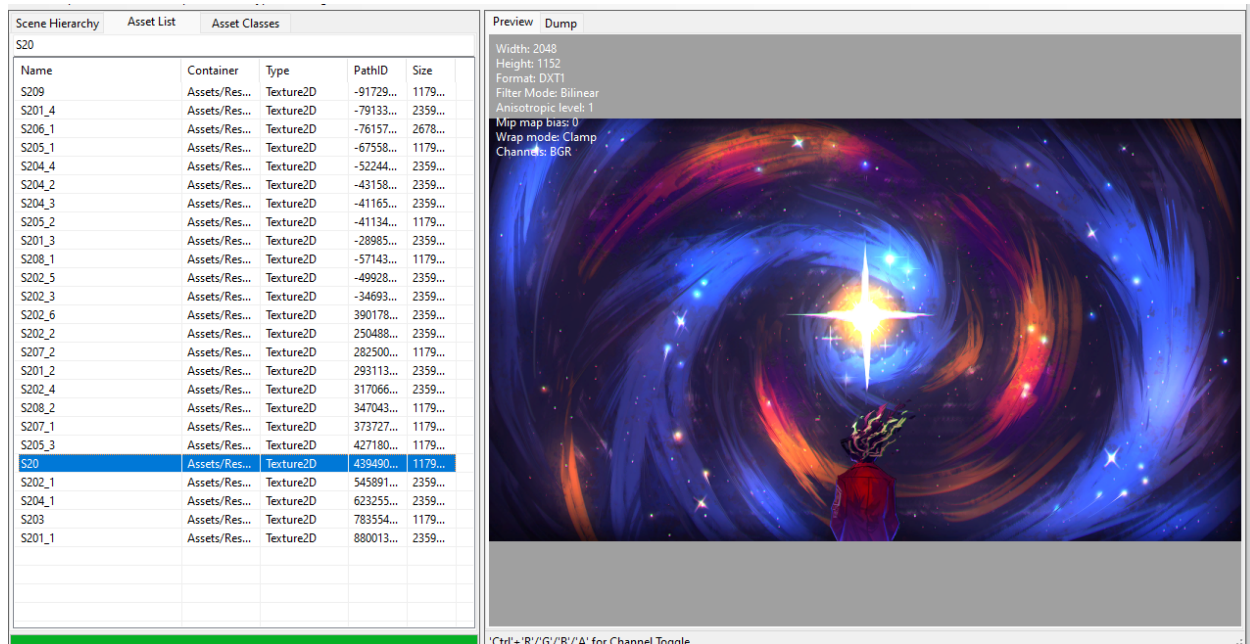


Using the name copied from the index, in quotation marks after the “bg”: parameter

NOTE: If there is no background set, some characters sprites will be placed too high up, and their heads will be too tall to fit on the screen

```
"id":0,  
"characterlist": "홍루:1",  
"feeling": "surprise",  
"emotion": "Flower",  
"cg": "S20"
```

CGs can be changed in practically the same way to bgs, except characters won't show up on top of them, here, only the CG S20 will be shown, and not Hong Lu
(to see what a specific CG is, you can look up the name listed in the index in AssetStudio)



BGM is pretty simple, but has some additional parameters, namely the volume

```
{
  "id":0,
  "characterlist": "홍루:1",
  "feeling": "surprise",
  "emotion": "Flower",
  "cg": "S20",
  "bgm": "StoryNervous:80"
},
```

Volume is set from 0 to 100, by using a colon after the name of the BGM, inside of the quotation marks.

SFX is the most complicated of these 4, as it has a few more parameters
Such as looping, or delaying when the audio is played after the dialogue line starts

This can be done through something like this:

```
{
  "id":0,
  "characterlist": "홍루:1",
  "feeling": "surprise",
  "emotion": "Flower",
  "cg": "S20",
  "bgm": "StoryNervous:80",
  "soundeffect": "SFX_Zippo_Open:loop, SFX_Zippo_Open:delay=0.2"
```

With the first sound effect looping forever, until a future line where “soundeffect”: “SFX_Zippo_Open:end” is called to end the SFX, and the second sound effect playing 0.2 seconds after the dialogue line starts.

3.4 Fading In and Out

I haven’t had too much time to test fade ins/fade outs, and the game is currently in downtime, so I will be updating this later once I have more details on the specifics

As far as I know, fading in and out can be used for both CGs and characters

For characters, it can be done by following the position with “:mode=___”

With the blank being able to be FADE_OUT, FADE_IN, or FADE_IN_AFTER_FORMER_FADE_OUT, the first two being pretty self explanatory, with the third one being used to bring a character back into the scene after previously having them fade out.

An example of this in action would be

```
},
{
  "id":0,
  "characterlist": "홍루:-1:mode=FADE_OUT",
```

This is setting Hong Lu’s position to off the screen (-1), as well as causing him to fade out.

Whereas this would be setting him to position 2 and having him fade in

```
"id":0,
"characterlist": "홍루:2:mode=FADE_IN",
```

For CGS, there’s a bit more specifics to the fade in/out controls, you can have cgs fade in/out just by putting a : after the cgs name, followed by either FadeIn, FadeOut_Long, FadeOut,

FadeIn_Long, FadeIn_Continue, On_Continue and FadeOut_Continue, The first 4 are very self explanatory, and just determine how long the fade in/out lasts, if trying to transition between two cgs, you can do something like this:

```
"cg": "S20:FadeIn, black:off",
```

This would have the CG S20 fade in, while turning off the black cg which would've been set beforehand.

Part 4: Screen Effects

<https://youtu.be/8bQzB7W8E0w> Video showcase of the mod created with the two json files below

[Google Drive Link with 2 updated json files. meant to show off the effects described here](#)

There are various screen effects that can be setup and used through the "effect": parameter (Warning in advance, these can be very confusingly set up, and I'm not too sure about them right now either)

4.1 CameraMoveZoom

The first of these screen effects is CameraMoveZoom, which can be used like this:

```
"id":7,  
"characterlist": "카론:7",  
"effect": "CameraMoveZoom:mode=on:duration=0:vertical=-0.3:horizontal:0.5:zoom=1.5:delay=0"
```

This can be used to zoom into a position on the screen

The individual parameters in this are:

Duration: The time it takes to zoom to the point

Mode: can be either on or off, use on to enable to zoom, and use off to disable it

Vertical: The vertical movement of the zoom

Horizontal: The horizontal movement of the zoom

Zoom: The magnification of the zoom

Delay: The length of time to wait before starting the zoom after the line begins

4.2 StandingJump

The next screen effect is StandingJump, which is pretty simple, you call the effect, and specify a position, and the character located in that position will do a jump in place.

```
"id":2,  
"characterlist": "카론:3",  
"effect": "StandingJump:position=3"
```

4.3 Shake

The next screen effect is Shake, which is pretty simple, it will shake whatever character is currently speaking a line, in this case, charon, the only parameter here that can be changed is the strength, everything else is required to be kept.

```
{  
  "id":5,  
  "characterlist": "카론:5",  
  "effect": "Shake:standing:strength=20:vibrato"
```

4.4 Misc

There's two more miscellaneous screen effects that are pretty simple, so I'm putting both here.

The first of the two is ClosedEyes, and can be set to either be on or off,

```
{
  "id":4,
  "characterlist": "카론:4",
  "effect": "ClosedEyes:mode=on"
},
{
  "id":5,
  "characterlist": "카론:5",
  "effect": "ClosedEyes:mode=off"
```

This will simply play the eyelids closing/opening animation and make the screen black when on, and undo it when off.

The second of the two is ChangeUI, which may be used more in the future, but currently only works with the Walpurgis Lob Corp UI, which can be called like so:

```
},
{
  "id":8,
  "characterlist": "헤르만:8",
  "effect": "ChangeUI:type=lobotomy"
```



Part 5: Misc Extras

There are various extra things that don't really fit into any category here, so they'll all be included down here.

First we have two that go together in a pair:

"waittime" and "isautonextbywaittime"

The first is set to a number, and determines how long you will have to wait before going onto the next line (in seconds), and isautonextbywaittime can be set to true, when set to true it automatically will proceed to the next line of text once the waittime is up.

Example of this:

```
"waittime": 2,
"isautonextbywaittime": true
```

This is how you would call it:

This line sets both the bg and cg to black screen, and starts a monologue (you need to have some kind of cg set for a monologue to work, won't work if you just have a background)

When changing dialogue you can use text colors with hex codes on either side of the text in the dialogue file to change the color of the text, with the format:

You can italicize/bold text in a similar way to changing dialogue color, as shown in the image above

Part 6: Thanks

HUGE shoutout to Greg (batulgaming on discord) for all his help with the index and just getting me into modding in general