

Beginner's Guide: How to Create Mods

Information for people getting started with making their first mods. Level mods, sound mods, string mods, and more.

- **General Information**
 - **Extracting Resources**
 - **File Extensions**
 - **Creating Mods**
 - **Command Prompt Tips**
 - **EternalMod.json**
- **Level Modding**
 - **Getting Started**
 - **Text Editor**
 - **EntityHero**
- **Sound Modding**
 - **Extracting Game Audio**
 - **Creating Sound Mods**
 - **Sound File Extensions**
- **String Modding**

- **Extracting Strings**
- **Creating String Mods**
- **String Customization**
- **String File Extensions**
- **Texture Modding**
 - **Texture Modding Redirect Links**

General Information

Start here for learning how to create mods.

Extracting Resources

In order to create mods, you will need to have the game's resources extracted into the various files that can be edited. Resources are divided by levels or general categories. They will be sorted into folders when extracted.

Extraction Tools:

EternalResourceExtractor by PowerBall253 - [Download](#)

Command-line tool. Exports everything, but textures/models are exported as "headers" only - not for texture mods.

CMD Syntax: `EternalResourceExtractor.exe [path to .resources file] [out path] [options]`

[\[Documentation & Source Code\]](#)

SAMUEL Asset Extraction Tool by SamPT - [Download](#)

GUI tool. Exports textures, .decls, and .entities files.

[\[Documentation & Source Code\]](#)

VEGA by DTZxPorter - [Download](#)

GUI tool. Exports textures and models. Preferred tool for texture modding.

Make sure you have enough disk space when extracting resources. As of Game Update 6.3, EternalResourceExtractor requires 17.4 GB of free space to extract everything. SAMUEL or VEGA require even more space, since they export full-size images. If using those tools, you should only export the textures you need.

List of Resources:

gameresources contains general information that will be used for all levels. You will be using this resource frequently.

Resource	Description
gameresources	General

e1m1_intro	Hell on Earth
e1m3_cult	Cultist Base
e1m4_boss	Doom Hunter Base
e2m1_nest	Super Gore Nest
e2m2_base	ARC Complex
e2m3_core	Mars Core
e2m4_boss	Sentinel Prime
e3m1_slayer	Taras Nabad
e3m2_hell	Nekraval
e3m2_hell_b	Nekraval - Part II
e3m3_maykr	Urdak
e3m4_boss	Final Sin
e4m1_rig	UAC Atlantica Facility
e4m2_swamp	The Blood Swamps
e4m3_mcity	The Holt
e5m1_spear	The World Spear
e5m2_earth	Reclaimed Earth
e5m3_hell	Immora
e5m4_boss	The Dark Lord
e6m1_cult_horde	Horde Mode - Cultist Base
e6m2_earth_horde	Horde Mode - Reclaimed Earth
e6m3_mcity_horde	Horde Mode - The Hold

hub	Campaign Hub
dlc_hub	ARC Carrier
shell	Main Menu
pvp_bronco	Battlemode - Celestial
pvp_darkmetal	Battlemode - Corrosion
pvp_deathvalley	Battlemode - Penance
pvp_inferno	Battlemode - Tundra
pvp_laser	Battlemode - Forsaken
pvp_shrapnel	Battlemode - Torment
pvp_sideswipe	Battlemode - Stronghold
pvp_thunder	Battlemode - Armageddon
pvp_zap	Battlemode - Extraction
tutorial_sp	Slayer Tutorial
tutorial_demons	Demon Player Tutorial
tutorial_pvp_laser	Battlemode - Tutorial
warehouse	Skins
meta	Contains "container.mask"

There are folders with "_patch_#" appended at the end and they contain overriding files for their parent folder.

The patch number does not always mean it has a high priority because DOOM Eternal often changes the priorities with each update.

File Extensions

This is a list of the most common files you'll edit when creating a mod. It is recommend that you download the tools to [extract the game resources](#) before continuing. That way, you can reference these files as you go.

.decl File Extension

.decl files (short for "declaration") are the most commonly used files in Doom Eternal modding. They are a form of script or "pseudo-code" that is interpreted by the game engine. Many of the variables and numeric values used by the game are defined in these files. They can be edited with a text editor.

Filepath:

```
<resource>/generated/decls
```

Example:

```
gameresources/generated/decls/weapon/weapon/player/rocket_launcher.decl
```

> File to edit the Rocket Launcher.

.entities File Extension

The .entities file contains all of the data in a map/level except geometry — if you want to make a level mod, you will need to learn how to edit the .entities file. These files are stored compressed, and require [a tool to decompress them](#). After that, the file can be edited with a text editor.

Filepath:

```
<resource>/maps/game/<campaign>/<level_name>/<level_name>.entities
```

Example:

```
elm1_intro_patch1/maps/game/sp/elm1_intro/elm1_intro.entities
```

> Level file path for Hell on Earth.

.tga File Extension

These are texture files. If you want to create custom skins or other artwork, you will most likely be editing these .tga files. Not everything "visual" is a .tga file. Many of the game's particle effects, lighting effects, and other visual effects are controlled by .decl files instead.

Texture editing software is required to edit .tga files.

Filepath:

```
<resource>/models and <resource>/art
```

Example:

```
gameresources/models/monsters/arachnotron/*
```

> Texture files for the Arachnotron's model.

Despite the name, the ".tga" files used in Doom Eternal are not actually .tga (targa) images. You will need to use a tool such as SAMUEL or VEGA to extract them in a usable format (.dds, .png, or .tiff).

Creating Mods

Getting Your Mod to Work:

- For your mod to function in the injector, you must have your folder structure EXACTLY the same as it is structured in the extracted files.
- When making mods, you should copy only the files you need to edit. Do not change anything in the extracted files so you can reference them again in the future.
- Be sure to check if your mod is in the highest priority it can be. Editing something in "gameresources" may not work if it is found in "gameresources_patch1".
[Here is a list showing the resource load priorities, from top to bottom](#)
- You can find the top-to-bottom resource load order in #resources, called "PackageMapSpec.json sheet".
- When finished with your mod, put all the contents into a .zip file. When testing your mod, you do not necessarily have to make it a .zip file. You can just move the folders into the "Mods" folder for your convenience.

If you are having trouble, you can always look at another person's .zip mod for reference.

Updating Your Mod:

- When DOOM Eternal updates, the resource load order will often change and will sometimes break your mod.
- Updating your mod can often be as simple as moving your files to it's higher priority, which you can tell if you see those files moved into another patch.
- Still, you should test your mod to see if anything has changed. Sometimes ID Software makes extra changes that can be hard to identify or revert.

Command Prompt Tips

There are thousands of extracted files available and sometimes it can be difficult to locate what you need.

On the Windows Operating System, you will be using the Console Command Prompt (CMD). To access it, simply type "command prompt" or "cmd" in the search bar. Windows PowerShell also works.

Change Directory

It will be much easier issuing commands when changing the current directory. You should be changing the directory to the folder where you extracted all your files.

Windows and Linux | cd

The Change Directory command for Linux and Windows is the same.

Syntax:

```
cd [directory]
```

Example:

```
cd C: \Users\Username\Desktop\Extracted Files
```

On Windows:

An easy way to change directories in is to navigate to the directory in File Explorer, select the upper list that names all the folders in your directory (usually starting with "This PC"), copy what is listed, then paste it into CMD.

Find String

To find specific text strings within a file or group of files. Useful for finding .decl files.

Windows | [findstr](#)

Syntax:

```
findstr /arguments "text" [filename]
```

Example:

```
findstr /si "ai/fodder/imp_stone" e5m1_spear\generated\decls\*
```

> This example command will list out the .decl files that define the Stone Imp as an entity to be spawned in The World Spear.

> The `/si` argument makes the command also search sub-directories of the file and ignore case-sensitivity.

> The `*` at the end is a wildcard, meaning that it will search for everything within the decls folder.

Although the Windows file structure divide their directories with the back slash ("\), the extracted files will reference directories using the forward slash ("/). You can see this in the example for findstr above.

Linux | grep

Syntax:

```
grep - arguments "text" [filepath]
```

Example:

```
grep "ai/fodder/imp_stone" e5m1_spear/generated/decls/*
```

For Linux, the directories use the forward slash ("/) just like how the extracted files do.

Sometimes a surplus of results will print out depending on how vague the key term is.

File Compare

Compares the difference between two files line-by-line.
Useful for comparing a modded file with its original version.

Diffchecker | <https://www.diffchecker.com>

The Diffchecker website compares differences and similarities between 2 code syntax.
Select all syntax of the original .decl file and copy it over to one of the fields. Then do the same for the modified .decl file.

Windows | fc

Syntax:

```
fc /arguments [file1] [file2]
```

Example:

```
fc "e5m3_hell\generated\decls\aiupgrades\buffpod.decl" "test_buffpod.decl"
```

Linux | **diff**

Syntax:

```
diff -arguments [file1] [file2]
```

Example:

```
diff "e5m3_hell/generated/decls/aiupgrades/buffpod.decl"  
"test_buffpod.decl"
```

General Information

EternalMod.json

You can provide additional info for your mod by creating a file called "EternalMod.json" in the .zip file alongside the rest of your resource folders.

This file is not required when creating mods, but it can be helpful for users when organizing their mods.

Create the file **EternalMod.json** as shown:

Name	Date modified	Type
e4m1_rig	9/21/2021 10:46 PM	File folder
e4m1_rig_patch1	9/21/2021 10:46 PM	File folder
gameresources_patch2	9/21/2021 10:46 PM	File folder
EternalMod.json	9/14/2021 10:07 PM	JSON File

Usage:

```
{
  "name": "", \ Mod Name
  "author": "", \ Mod Author
  "description": "", \ Mod Description
  "version": "", \ Version Number
  "loadPriority": 0, \ Injector Load Priority
  "requiredVersion": 0 \ Required Mod Loader Version
}
```

Example:

```
{
  "name": "My First Mod",
  "author": "Newbie Modder",
  "description": "This is my first Doom Eternal Mod.",
  "version": "1.0",
  "loadPriority": 500,
  "requiredVersion": 14
}
```

```
}
```

These contents are only visible for users if they are using the GUI-based application, **EternalModManager.exe** which comes packaged with the Mod Injector.

Loading Priorities:

Manually setting loading priorities can be helpful when considering file conflicts with other mods.

`loadPriority` has greater importance than the other fields as it determines what order the mod will be injected.

By default, the value is 0. The higher the number, the sooner the individual mod will be injected. **Load priority numbers can get to the negatives.**

The Mod Injector loads its mods from **highest load priority to smallest**, meaning that if two mods change the same files, **the mod with the smaller load priority number will override the mod with the larger load priority number.**

Required Version:

`requiredVersion` refers to the version number of the Mod Loader, known as `DEternal_loadMods.exe` and not the Mod Injector itself.

To figure out what the current Mod Loader version, issue type in a command prompt:
"DEternal_loadmods.exe --version"

Be sure to be in the same directory as the .exe file before issuing the command.

See Also:

- [Command Prompt Tips](#)

Level Modding

Information for people who are looking to create Custom Master Levels, Slaughter Maps, or other mods that involve changing the levels.

Getting Started

Entities Files

Levels are edited through ".entities" files and are saved in:

```
<resource>/maps/game/<campaign>/<level_name>/<level_name>.entities
```

Example:

```
e1m1_intro_patch1/maps/game/sp/e1m1_intro/e1m1_intro.entities
```

> This is the level file path for Hell on Earth.

Remember to check for the highest resource load priority for the level

M347h00k:

M347h00k (aka Meathook or MH) originally by Chrispy - [Download](#)

> M347h00k unlocks developer commands and allows you to edit your level in-game.

> Although not mandatory, this would make level modding significantly easier.

To install: Download [XINPUT1_3.dll](#) and place it into the DOOM Eternal installation directory (like how you installed the Mod Injector).

Tools for Editing Levels:

Text Editor - [Notepad++](#) [Recommended](#)

EntityHero by Scorp0rX0r - [Download](#)

> EntityHero is a tool to help newer modders into level modding. It helps with organizing entities and makes it easier to move around eventCalls.

> *To install: Download the EntityHero.zip and extract its contents to whatever folder you want. Make sure to copy [oocore_8_win64.dll](#) from DOOM Eternal's installation directory to the folder.*

After you have the tools you need, go to the [Entities & Custom Encounters](#) book to learn how to script encounters.

Text Editor

[Notepad++ is the Recommended Text Editor to Use](#)

.entities files are compressed and must be decompressed in order to edit Entities.

Decompressing Tools:

EntityHero by Scorp0rX0r - [Download](#)

File -> Export to text (make sure the file extension is still .entities)

SAMUEL Asset Extraction Tool by SamPT - [Download](#)

Currently exports textures, .decls, and .entities files.

idFileDecompressor by Zwip-Zap Zapony - Coming Later

Even though the .entities file is decompressed, make sure it still retains the .entities file extension.

When editing a decompressed .entities file while you are in game, you will need to create a folder called "overrides" in the DOOM Eternal installation directory. Structure it like this:

```
overrides/maps/game/<campaign>/<level_name>/<level_name>.entities
```

Example:

```
overrides/maps/game/sp/e1m1_intro/e1m1_intro.entities
```

> This is the level file path for Hell on Earth.

You can place as many level files you like in the overrides folder

With M347h00k installed, you can enter your level and edit the decompressed entities file while you are in-game.

Typing in the console command, `mh_force_reload` will reload the level with the recent changes you made in the text file.

EntityHero

EntityHero by Scorp0rX0r - [Download](#)

“ EntityHero is a tool to help newer modders into level modding. It helps with organizing entities and makes it easier to move around eventCalls.
To install: Extract the files to whatever folder you want. Make sure to copy `oocore_8_win64.dll` from DOOM Eternal's installation directory to the folder.

EntityHero does not require you to decompress .entities files.

You can open the compressed files within the application, but you should be in your chosen level and use

File -> Open from MH to open the M347h00k instance. Be sure so save your instance somewhere. With EntityHero, you can press the button, "Reload level" and it will perform the command "mh_force_reload" for you.

To avoid issues, do most of the typing in a text editor (such as [Notepad++](#)). Copy an entity or eventCall and paste them into the text editor, make your changes, then copy that and paste it back into where it was.

When pasting eventCalls, ignore the "item[#] = {" bracket because it would sometimes throw an error.

Copying from EntityHero to Text Editor:

1. Copy the eventCall selected

> eventCall	ENCOUNTER_SPAWN_DREAD_KNIGHT
> eventCall	ENCOUNTER_SPAWN_ANY
> eventCall	
> eventCall	
▼ eventCall	ENCOUNTER_SPAWN_BARON
eventDef	spawnPossessedAI
▼ args	
eEncounterSpawnType_t	ENCOUNTER_SPAWN_BARON
entity	tut_heavy_open_1
string	main
entity	tut_heavy_open_2
eEncounterSpawnType_t	ENCOUNTER_SPAWN_ANY
string	main
bool	false
> eventCall	
> eventCall	ENCOUNTER_SPAWN_BARON

2. Paste into text editor

```
1 item[20] = {
2     eventCall = {
3         eventDef = "spawnPossessedAI";
4         args = {
5             num = 7;
6             item[0] = {
7                 eEncounterSpawnType_t = "ENCOUNTER_SPAWN_BARON";
8             }
9             item[1] = {
10                entity = "tut_heavy_open_1";
11            }
12            item[2] = {
13                string = "main";
14            }
15            item[3] = {
16                entity = "tut_heavy_open_2";
17            }
18            item[4] = {
19                eEncounterSpawnType_t = "ENCOUNTER_SPAWN_ANY";
20            }
21            item[5] = {
22                string = "main";
23            }
24            item[6] = {
25                bool = false;
26            }
27        }
28    }
29 }
```

3. Make your changes and only copy the following

```
1 item[20] = {
2     eventCall = {
3         eventDef = "spawnPossessedAI";
4         args = {
5             num = 7;
6             item[0] = {
7                 eEncounterSpawnType_t = "ENCOUNTER_SPAWN_BARON";
8             }
9             item[1] = {
10                entity = "tut_heavy_open_5";
11            }
12            item[2] = {
13                string = "possessed";
14            }
15            item[3] = {
16                entity = "tut_heavy_open_5";
17            }
18            item[4] = {
19                eEncounterSpawnType_t = "ENCOUNTER_SPAWN_ANY";
20            }
21            item[5] = {
22                string = "main";
23            }
24            item[6] = {
25                bool = false;
26            }
27        }
28    }
29 }
```

4. Reinsert the eventCall into EntityHero

- > eventCall ENCOUNTER_SPAWN_DREAD_KNIGHT
- > eventCall ENCOUNTER_SPAWN_ANY
- > eventCall
- > eventCall
- ▼ eventCall ENCOUNTER_SPAWN_BARON

5. If you are replacing an eventCall, delete the original one, but you can always add more eventCalls this way.

Issues with EntityHero:

EntityHero can be a bit weird and is prone to crashes. Be sure to frequently save to avoid losing progress.

- When editing text directly in EntityHero and leaving a field blank (such as a string), the field might auto-generate a garbled mess that will result in the game crashing when reloading the level.

Copy the corrupted entity to a text editor and replace the garbled text with two quotations to fix this.

- When double clicking the parent field of an entity in EntityHero, the text, "entity" would be the only thing showing up. Doing this action also sometimes generates a garbled mess in that field.

Copy the corrupted entity to a text editor and replace the garbled text with `entity` to fix this.

- Sometimes, entities will randomly duplicate its quotations, particularly when relating to Invasion stuff.

This results in crashes when reloading the level, and this issue is common for Base Campaign levels.

A solution is to decompress the entities file to plain text, find where all the duplicate quotations are (use Find and Replace), then re-compress the entities file.

Newer versions of EntityHero tend to mitigate these issues so you may not experience any of these.

Sound Modding

Information for people who are looking to create Music, Sound Effect, and Voice Over mods.

Extracting Game Audio

It is recommended to do all of your extracting in the SAME disk drive (ex: C:\)

Extracting Tools

EternalAudioExtractor by proteh - [Download](#)

Exports WEM & OPUS audio from SND files for Windows OS.

EternalAudioExtractorLinux by PowerBall253 - [Download](#)

Exports WEM & OPUS audio from SND files for Linux OS.

Make sure you have at least 2.00 GB of storage if you plan on extracting music.
550 MB for sound effects, 50 MB for voice overs (for each language).

Extraction Instructions

Instructions are for Windows users.

- Extract the contents of **EternalAudioExtractor** somewhere on your machine.
- Launch either "**Extract_All.bat**" or "**Extract_Single.bat**" depending if you want to extract all sound files or from a single directory.
- Copy and paste the filepath of your DOOMEternal installation directory.
(ex: **C:\Program Files (x86)\Steam\steamapps\common\DOOMEternal**)
- Input the name of the **.snd** file you want to extract (if applicable).
- Input the name of the output directory. Make sure the directory is either empty or nonexistent.
- If you are extracting music files, press **[Y]** to enable automatic .ogg conversion.
- If necessary, press **[Y]** again to enable the extraction of unused sound files.
- If everything looks correct, press **[Y]** to extract its contents.
- Press **[Y]** to close the batch script open the output folder in a new window.

Music files must be in the OGG sound format before they can be edited.

Unused game audio cannot be referenced in-game.

List of Resources

<u>Resource Name</u>	<u>Information</u>
music.snd	Contains WEM music files for Base Game & DLC1
music_patch_1.snd	Contains WEM music files for DLC2
music_patch_2.snd	Contains WEM music files for Horde Mode
sfx.snd	Contains OPUS sound effects files for Base Game
sfx_patch_1.snd	Contains OPUS sound effects files for DLC1
sfx_patch_2.snd	Contains OPUS sound effects files for DLC2
sfx_patch_3.snd	Contains OPUS sound effects files for post-DLC2
vo_English(US).snd	Contains OPUS voice over files for Base Game & DLC1
vo_English(US)_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_French(France).snd	Contains OPUS voice over files for Base Game & DLC1
vo_French(France)_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_German.snd	Contains OPUS voice over files for Base Game & DLC1
vo_German_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_Italian.snd	Contains OPUS voice over files for Base Game & DLC1
vo_Italian_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_Japanese.snd	Contains OPUS voice over files for Base Game & DLC1
vo_Japanese_patch_1.snd	Contains OPUS voice over files for DLC2 and after

vo_Polish.snd	Contains OPUS voice over files for Base Game & DLC1
vo_Polish_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_Portuguese(Brazil).snd	Contains OPUS voice over files for Base Game & DLC1
vo_Portuguese(Brazil)_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_Russian.snd	Contains OPUS voice over files for Base Game & DLC1
vo_Russian_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_Spanish(Mexico).snd	Contains OPUS voice over files for Base Game & DLC1
vo_Spanish(Mexico)_patch_1.snd	Contains OPUS voice over files for DLC2 and after
vo_Spanish(Spain).snd	Contains OPUS voice over files for Base Game & DLC1
vo_Spanish(Spain)_patch_1.snd	Contains OPUS voice over files for DLC2 and after

Sound Modding Guides

- [Creating Sound Mods](#)
- [Sound File Extensions](#)

Creating Sound Mods

See [Extracting Game Audio](#) before proceeding.

Instructions are for Windows users.

Audio Editing

Download an audio-editing tool of your choice, such as Audacity - [Download](#)

1. Import the OGG or OPUS files and edit them in your audio-editing tool.
2. Make sure that your edits match the start and end length of the audio track.
3. **SFX & VO:** Convert all of the audio you want to export to OPUS.
(Export Audio -> OggOpus - 64 kbps for high quality)
4. **MUSIC:** Convert all of the audio you want to export to WAV. (32 bit for high quality)

FFmpeg (SFX & VO)

You need the FFmpeg extension for Audacity to import/export OPUS audio files - [Download](#)

1. Install **ffmpeg-win-#.#.#.exe** -> Follow the installation instructions
2. Open Audacity -> Edit -> Preferences -> Libraries
3. FFmpeg Library: Locate... -> Browse... -> Find **avformat-55.dll** -> OK

Sometimes, Audacity will automatically find avformat-55.dll

Wwise Launcher (MUSIC)

Create an audiokinetic account (use free trial) and install Wwise Launcher - [Download](#)

1. Go to **Wwise Launcher** -> **WWISE** -> **Launch Wwise** (64-bit) -> **New** (Project Launcher) -> Provide it a name -> Select **None** (Import assets to project) -> **OK**

2. **Close** (License Manager) -> **Property Editor** (SHIFT + K) -> **Source Settings** -> Default Conversion Settings ... -> Factory Conversion Settings / Vorbis / **Vorbis Quality High** -> **OK** -> **OK**
3. **Audio File Importer** (SHIFT + I) -> Add Files... -> Select your WAV files -> **Open** -> **Import**
(Replace any if needed)
4. **Project** -> Convert All Audio Files... -> Check **Windows** (If it is not already checked) -> **OK**
5. **File Manager** (SHIFT + F1) -> Right Click **Project Folder Directory** -> Open **Containing Folder** -> .cache\Windows\SFX (It is recommended to bookmark this directory)
6. Be sure that the exported **WEM** files end with **_3F75BDB9**

It is recommended to repeat **Step 2 EVERY TIME** you launch Wwise, in case the default Factory Conversion Settings are automatically restored.

Injecting Audio

1. In your DOOMEternal\Mods folder, create a folder called named based on where the file was originally found.
(**ex: music_patch_1**)
2. Copy your converted WEM file(s) in there and make sure they have the same name as extracted files.
(**ex: samuelsbase_music_main_heavy_0_0_id#15098487.wem**)
3. Run the **EternalModInjector** like normal.

Sound Modding Guides

- [Extracting Game Audio](#)
- [Sound File Extensions](#)

Sound File Extensions

These are the sound file types you will be interacting with.

.ogg

Extracted music files from "**music**" and "**music_patch_1**" will be converted to this format if they are to be edited.

OGG files **CANNOT** be re-injected into DOOM Eternal.

Example:

cultist_base_music_main_heavy_0_id#156134790.ogg

.opus

Extracted sound effects & voice over files from "**sfx**", "**sfx_patch_1**", "**vo_English(US)**", etc will be in this format.

OPUS files **CAN** be re-injected into DOOM Eternal.

Example:

ammo_bfg_id#373311939.opus

.wav

Edited music files must be converted into the WAV format before they can be converted into WEM files.

WAV files **CANNOT** be re-injected into DOOM Eternal.

Example:

cultist_base_music_main_heavy_0_id#156134790.wav

.wem

After the edited music files have been converted into the WEM format, they **CAN** be re-injected into DOOM Eternal.

Example:

cultist_base_music_main_heavy_0_id#156134790.wem

.snd

These contain the audio files. They **CANNOT** be directly modified by a conventional audio editor. Instead, tools are used to extract individual audio files.

Example:

music.snd

.pck

These contain the audio resources. They **CANNOT** be directly modified by a conventional audio editor. Instead, [Fusion Tools](#) can directly modify PCK files.

Example:

[mus.pck](#)

If a **WEM** file has been exported correctly (Vorbis Quality High), they should have "**_3F75BDB9**" before its WEM extension. You will need to delete the "**_3F75BDB9**" in the WEM file's name before you can inject it into DOOM Eternal.

PCK files modified by [Fusion Tools](#) are not yet supported by [DEternal_loadMods](#), which is a program that allows certain mods type to be injected.

See [Creating Sound Mods](#) for more information.

Sound Modding Guides

- [Extracting Game Audio](#)
- [Creating Sound Mods](#)

String Modding

Information for people who are looking to create mods that affect Text.

Extracting Strings

See [Extracting Resources](#) before proceeding. You must have **gamerresources.resources** extracted.

A [string](#) is a sequence of characters (text) that appears literally.

Extracting Tools

BlangJsonGenerator by PowerBall253 - [Download](#)

Exports BLANG files into JSON, a format that can be edited with raw text editor, such as [Notepad++](#).

BlangParser by proteh - [Download](#)

Directly modifies BLANG files with a simple GUI.

Currently, there are no released packages for BlangParser. You will either need to find a compiled version in the 2016+ Modding Discord server or compile it with a C# compiler.

Each BLANG file usually range from 1.5 MB to 2 MB of storage.

Finding BLANG files

- As of Update 6.4, the highest-priority string files are located in **gamerresources_patch2\strings** from your resource output directory.
- Copy any language BLANG file of choice and paste it somewhere safe. This will be the string file that you will modify.

Extraction Instructions ([BlangJsonGenerator](#))

It is **recommended** to create a folder containing both "**BlangJsonGenerator.exe**" and the unmodified **BLANG** file.

- Run "**BlangJsonGenerator.exe**".

- Select option 1.
- Input the name of the **.blang** file. (ex: **english.blang**)
- Input the language name. (ex: **english**)
- Input the name of the output file. (ex: **english.json**)
- Close the executable.
- You may now edit each string value. See [Creating String Mods](#) for more information.

The output folder **must** end with **.json**

List of Resources

<u>Resource Name</u>	<u>Information</u>
english.blang	Contains strings for the English language
french.blang	Contains strings for the French language
german.blang	Contains strings for the German language
italian.blang	Contains strings for the Italian language
japanese.blang	Contains strings for the Japanese language
latin_spanish.blang	Contains strings for the Latin American Spanish language
polish.blang	Contains strings for the Polish language
portuguese.blang	Contains strings for the Portuguese language
russian.blang	Contains strings for the Russian language
simplified_chinese.blang	Contains strings for the Simplified Chinese language
spanish.blang	Contains strings for the European Spanish language
traditional_chinese.blang	Contains strings for the Traditional Chinese language

String Modding Guides

- [Creating String Mods](#)
- [String Customization](#)
- [String File Extensions](#)

Creating String Mods

See [Extracting Strings](#) before proceeding.

Make sure to backup the unmodified **BLANG** file.

Strings can be added to **BLANG** files, but you must reference them with **decls** files.

String Editing (BlangJsonGenerator)

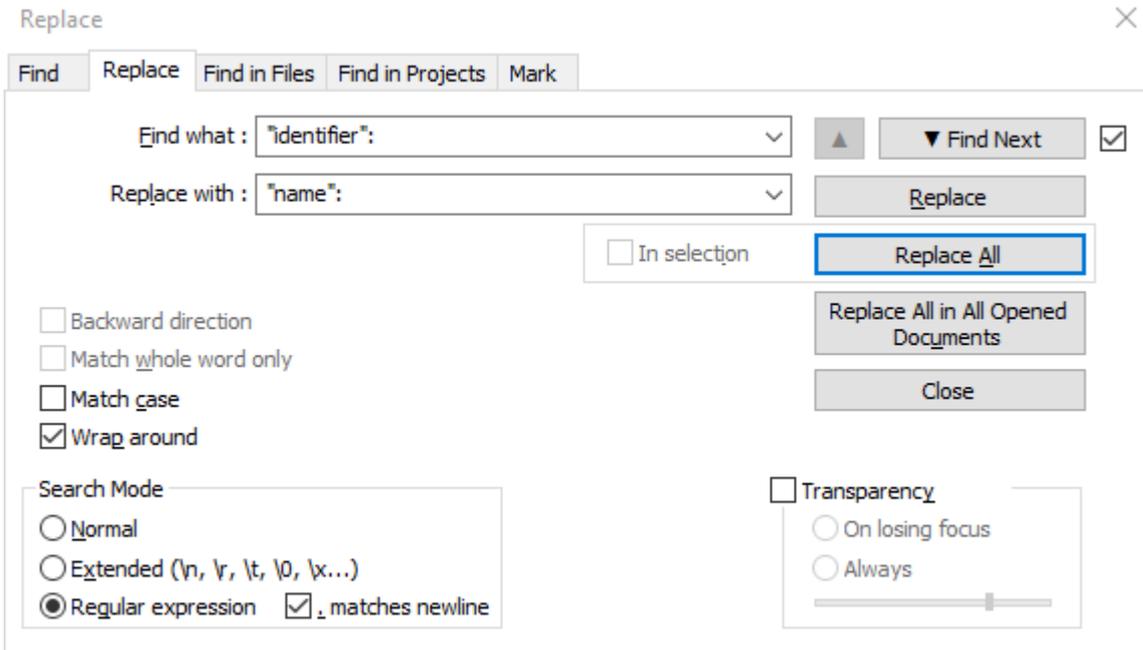
Any unused string can be **deleted**, so it can be **compatible** with other mods that affect other strings.

- Create a backup for the string JSON file.
- Open the string JSON file. (ex: **english.json**)
- The strings should be formatted like this:

```
1  {
2  "strings": [
3  {
4      "identifier": "#STR_SWF_ACCEPT",
5      "text": "Accept",
6      "modified": 0
7  },
8  {
9      "identifier": "#STR_SWF_CANCEL",
10     "text": "Cancel",
11     "modified": 0
12  },
13  {
14     "identifier": "#STR_SWF_CLOSE",
15     "text": "CLOSE",
16     "modified": 0
17  },
```

- If they are not, see the [Extracting Strings](#) page, under **Extraction Instructions: (BlangJsonGenerator)**

- Go to the Replace tab (CTRL + F) -> Replace -> Search Mode: Check **Regular**

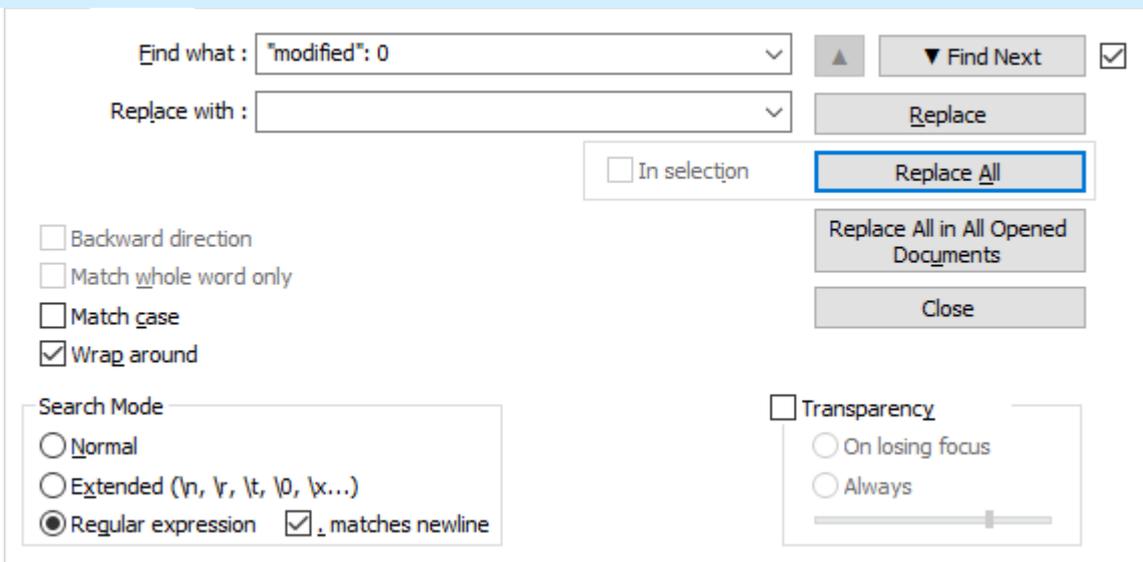


1) Replace All: 12958 occurrences were replaced in entire file

"identifier":

"name":

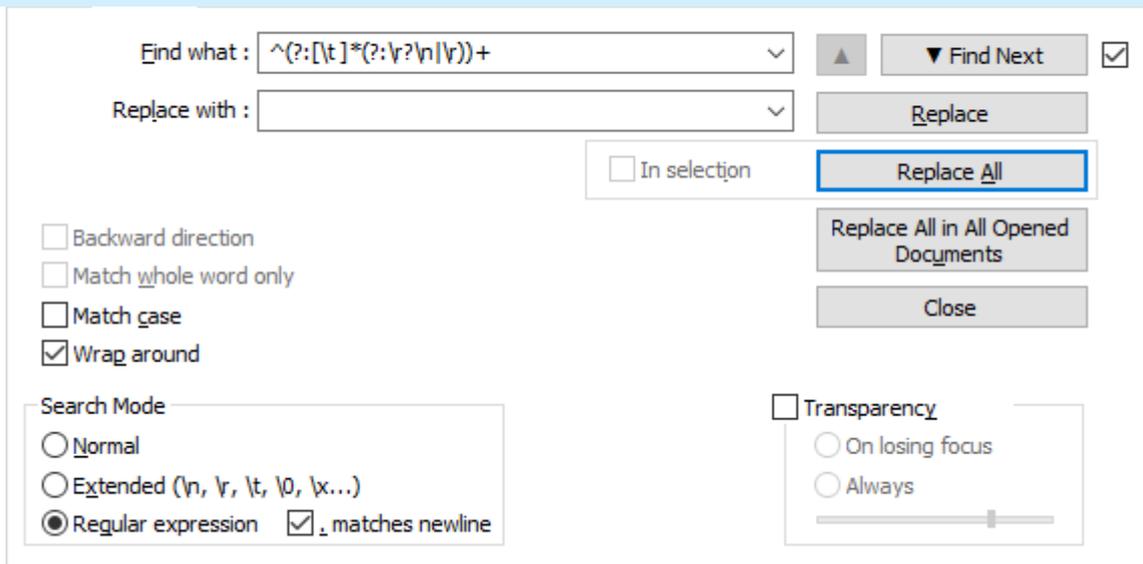
The "Replace with : " box is empty for screenshot 2.



2) Replace All: 12958 occurrences were replaced in entire file

"modified": 0

The "Replace with : " box is empty for screenshot 3.



3) Replace All: 12958 occurrences were replaced in entire file

`^(?:[t]*(?:r?\n|\r))+`

- The results should be formatted like this:

```
1  {
2  "strings": [
3  {
4      "name": "#STR_SWF_ACCEPT",
5      "text": "Accept",
6  },
7  {
8      "name": "#STR_SWF_CANCEL",
9      "text": "Cancel",
10 },
11 {
12     "name": "#STR_SWF_CLOSE",
13     "text": "CLOSE",
14 },

```

- The JSON file is now in a format that can be properly re-injected into DOOM Eternal. Save a backup of this JSON file as well, so you would not need to go through this process again (before the next DOOM Eternal update).
- Find the strings that you wish to modify. (CTRL + F)
- After modifying the strings of your choice, delete all other **unmodified** strings.

- Save your **modified** JSON file with the same name as it was exported. (ex: **english.json**)
- Create the following folder directories in your **DOOMEternal\Mods** folder:

gameresources_patch1\EternalMod\strings

- Copy your **modified** JSON file into the strings folder.
- Run the **EternalModInjector** like normal.

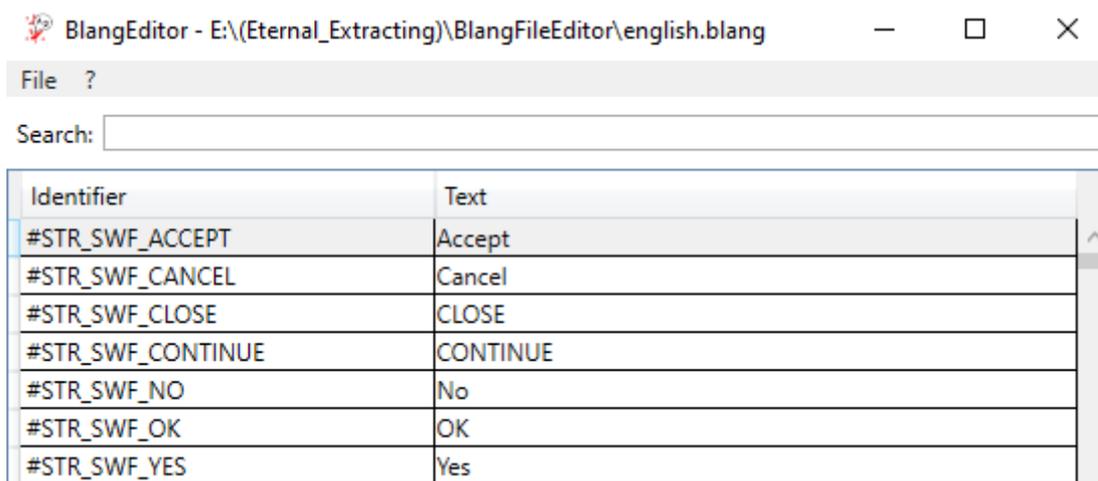
The commas "," at the end of each string name and text are unnecessary and can be deleted. However, they do not impact the syntax in any way.

Since JSON files can be edited with raw text editors, you can execute scripts or **regular expressions** to edit strings more efficiently.

String Editing (BlangParser)

The entire **BLANG** file will be modified, which means it will be **incompatible** with other string mods.

- Extract the contents of **BlangEditorv#.#.zip**
- Open **BlangEditor.exe** you should be prompted with the following:



- You may edit any string **Identifier** or **Text** you wish.
- Upon highlighting a string line, you may select the **DELETE** key to remove it.
- To add a string, scroll down to the bottom of the list and add your own **Identifiers** and **Text** lines.

- After modifying the strings go to **File** -> **Save As...** -> Give it the same name as its original BLANG file.
(ex: **english.blang**)
- Create the following folder directories in your **DOOMEternal\Mods** folder:

gameresources_patch2\strings

- Copy your **modified** BLANG file into the strings folder.
- Run the **EternalModInjector** like normal.

It is not recommended to delete existing string lines because DOOM Eternal will fail to recognize them.

String Modding Guides

- [Extracting Strings](#)
- [String Customization](#)
- [String File Extensions](#)

String Customization

Strings can be customized to help specific characters stand out.

String customization applies to both **BLANG** files and **idGuiEntity_Text** entities.

Available Colors

Start with ^ then a character from the list below, then end with ^7

Ex: "^5sample text^7" (without quotations) This will make the text teal.

- ^a - Gold
- ^b - Light Blue
- ^c - Green Yellow
- ^d - Orange
- ^z - White
- ^0 - Black
- ^1 - Red
- ^2 - Green
- ^3 - Yellow
- ^4 - Blue
- ^5 - Teal
- ^6 - Pink
- ^7 - White
- ^8 - Gray
- ^9 - Black

In general, helpful tips are colored teal, dangerous threats are colored red.

Spacing

- \n - New Line
- \t - Tab

Examples

- ^2Green Yellow^7 -> Green Yellow
- ^1Red^7 ^aGold^7 -> Red Gold
- ^1R^7^da^7^3i^7^2n^7^5b^7^bo^7^6w^7 -> Rainbow

This is a `^5sample^7` tip -> This is a **sample** tip

First line\nSecond line -> **First line**

Second line

Sample\tText -> **Sample** **Text**

Up to 9 separate color identifiers can be present in one string line.
Any additional colorization will cause some to be ignored.

See Also

- [idGuiEntity_Text](#)

String Modding Guides

- [Extracting Strings](#)
- [Creating String Mods](#)
- [String File Extensions](#)

String File Extensions

These are the string file types you will be interacting with.

.blang

String files that **CANNOT** be read by a raw text editor.

BLANG files **CAN** be re-injected into DOOM Eternal.

Example:

english.blang

.json

Extracted string files that **CAN** be read by a raw text editor.

JSON files **CANNOT** be re-injected into DOOM Eternal.

Example:

english.json

String Modding Guides

- [Extracting Strings](#)
- [Creating String Mods](#)
- [String Customization](#)

Texture Modding

Information for people who are looking to create mods that affect textures.

Texture Modding

Texture Modding Redirect Links

Eternal Texture Modding Guides

[Beginner's Guide](#)

[Intermediate Guide](#)

[Advanced Guide](#)