Eternal - How to Create Mods

Information for people getting started with making their first mods. Covers sound, string, model modding, and multiplayer mods.

- **Extracting Resources**
- **Creating Mods**
- **File Extensions**
- **EternalMod.json**
- **Command Prompt Tips**

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.

> It is recommend that you extract assets from resources that end in .backup or uninstall all mods before extracting to avoid getting modified files.

Extraction Tools:

EternalResourceExtractor by PowerBall253 - Download
Command-line tool. Exports everything, but textures/models are exported as "headers" only - not for texture mods. You may wish to filter the extraction to .decl files only. 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, models, .decls, and .entities files in a readable format.

[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 6.66 rev 2, EternalResourceExtractor requires 20.0 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
e1m2_battle	Exultia
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	Nekravol
e3m2_hell_b	Nekravol - 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 Holt
hub	Fortress of DOOM
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. You can see the priorities by checking the vanilla version of base/packagemapspec.json. As of update 6.66 rev-2.2, here are the known unusual patch priorities:*

- *gameresources_patch1 is higher than gameresourcces_patch2.*
- *e5m1_spear_patch1 is higher than e5m1_spear_patch2.*
- *e3m1_slayer_patch2 is higher than e3m1_slayer_patch3.*
- *e2m3_core_patch2 is higher than e2m3_core_patch3.*
- *e5m3_hell_patch1 is higher than e5m3_hell_patch2.*

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.

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:
`e1m1_intro_patch1/maps/game/sp/e1m1_intro/e1m1_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.

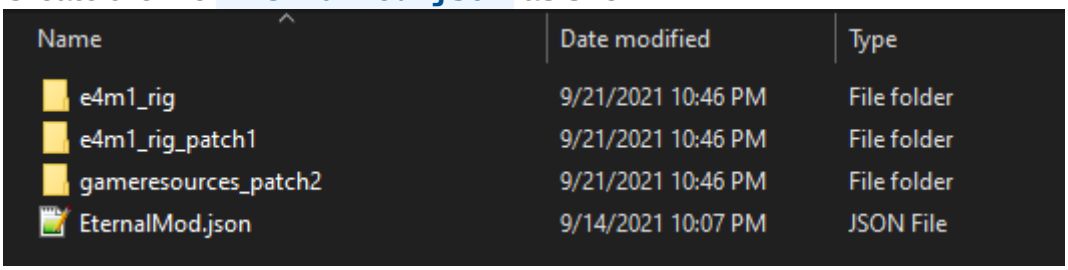
> Despite the name, the ".tga" files used in Doom Eternal are not actually .tga (targa) images. You will need to use a tool such as SAMUEL or VEGA to extract them in a usable format (.dds, .png, or .tiff).

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:



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 conflictions 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**

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.

Create Directory

Rather than making directories one at a time graphically, you may find it easier to create an entire path in one command.

Windows and Linux | mkdir

>

> The base command, mkdir, is the same on both Windows and Linux, however the flags are different.

Syntax:
`mkdir [directory/path]`

Example:
`mkdir warehouse_patch1/generated/decls/material2/art/weapons/supershotgun/skins/techhell/`

> On Linux:
> You need the -p flag to create directories recursively if their parents don't exist. If your Linux distribution does not do so automatically, add an alias to the config file for your favorite shell so that you can just type mkdir without the -p.

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 -ir "ai/fodder/imp_stone" e5m1_spear/generated/decls/*
```

The -ir arguments make the command search sub-directories recursively and ignore case-sensitivity. If you intentionally want a case-sensitive search, you can leave out the -i flag and use just -r.

> 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 /n "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"
```