How to Create Custom Assets

Information for people who want to learn how to create their own custom assets.

- **Custom AI**
    - **Creating Custom AI - decls**
    - **Creating Custom AI - entities**
    - **Creating Custom AI - textures**
- **Moving Assets**
    - **Assets Info Redirect Links**

Custom AI

Information about how to create and manage custom AI variants.

Creating Custom AI - decls

This is an ADVANCED GUIDE. You are expected to have a decent understanding of decls, textures, and entities.

See the Following Books Before Proceeding

- **Beginner's Guide: How to Create Mods**
- **Eternal Texture Mods: A Comprehensive Guide**
- **Entities & Custom Encounters**

Basics

For starters, do all of your editing on an existing AI, so you know what the results will look like before you spend a large sum of time managing an entirely new AI variant.

For your custom AI, you want to copy all the decl files that will be changed. The **entitydef** file is the most important and mandatory for almost all custom AI. Paste the vanilla decls into a **resource patch with a very high priority**. Patch priority changes between each update, so you will most likely need to change it after each update. As of Update 6.66, **e4m2_swamp_patch1** is a great patch to use.

Be sure to rename the decls too, so they do not replace properties of the vanilla AI.

After making changes to the decls, you will need to link them to the other modified decls if necessary. Make sure to do so for all of the modified decls. For example, to link the demon's FX, go to the entitydef file and change:
fxDecl = "character/stonegargoyle/gargoyle";
to
fxDecl = "character/stonegargoyle/stonegargoyle";

The stonegargoyle FX decl is a modification to the vanilla gargoyle decl.

For every added asset, you will need to add them to the **EternalMod\assetsinfo** JSON file. This must be done in BOTH the directory that contains all the added decls AND the level patch that holds your entities file(s). More info on this soon.

After everything is linked, move on to the level that you want to spawn your new custom AI.

Custom AI

Creating Custom AI - entities

See **Creating Custom AI - decls** before proceeding

idAI2 - Reference

Reference the original **idAI2** file for the vanilla AI and copy it into a separate text file. Like the the **entitydef** decl file, this links all the decls together in a usable format. You will need to copy all of the decl references in this entity, just like for the **entitydef** file.

materialRemap is used to apply the custom textures to the new demon. Under **renderModelInfo**, a **materialRemap** parameter can be added (if it does not already exist). The following is an example.

```
renderModelInfo = {
	model = "md6def/characters/monsters/gargoyle/base/gargoyle.md6"; // base gargoyle model to use
	materialRemap = {
		num = 2;
		item[0] = {
			from = "models/monsters/gargoyle/gargoyle"; // replace existing texture
			to = "models/customization/monsters/mancubus/set12/mancubus_guns_set12"; // to this
		}
		item[1] = {
			from = "models/monsters/gargoyle/gargoyle_body";
			to = "models/customization/monsters/mancubus/set12/mancubus_body_set12";
		}
	}
}
```

Notice that the **set12** skin for the the Mancubus is used. See **Creating Custom AI - textures** for more information.

idTarget_Spawn - Reference

The **idTarget_Spawn** entity is used to spawn the custom AI. You will need more per custom AI that you want to spawn. Make sure to change the spawn positions for each. Always reference the idAI2 entity when spawning the custom AI, so you would not accidentally spawn the vanilla AI.

Troubleshooting

The process of getting custom AI working is a rather challenging task. You are bound to encounter some issues. Double check all of your **modified** decls and reference them with their **vanilla** counterparts. Same applies to **entities**. Ask yourself the following:

- Are you spawning your AI correctly?

If not, check the **idAI2** and **idTarget_Spawn** for typos. The **encounter manager** can also have syntax errors.

- Are the textures loading consistently?

If not, check the **materialRemaps** or **TGA** files for typos. Also, check if the textures are added correctly in the assetsinfo JSON file.

- Does your AI function as intended?

If not, check the **decls** for typos or mislinks. The **entitydef** file might be the root of your problems if there is a mislink.

Double check BOTH assetsinfo JSON files for typos. Most crashes occur from syntax errors.

See Also

- **Entities**

Custom AI

Creating Custom AI - textures

See **Creating Custom AI - decls** before proceeding

Creating Custom Textures

While adding custom textures into the game is possible, it is not the optimal choice. Even if the added textures are in a relatively high patch directory, there is the chance that some of them will not load properly, especially if there are a lot of added assets in the mod(s).

Instead, it is recommended to replace the textures of an existing playable demon skin. Since injecting gameplay-altering mods disable Battlemode entirely, there is little concern for distorting playable demon skins. You will want to edit the regular demon skin like normal and replace its name with the cosmetic skin that you want to replace. Make sure to include the ENTIRE file extension (ex: **mancubus_body_set12.tga$streamed$mtlkind=albedo**)

Also, be sure to replace ONLY the matching **$streamed$** TGA extension. In the case of Gargoyles, replace any **.tga$streamed$mtlkind=albedo** file with that of the same extension for a **set12** mancubus skin, for example. Most cosmetic skins are located in **warehouse\models\customization\monsters**

If your extracted textures do not have the **$streamed$** TGA extension, then reference here - **Resource Data**

See Also

- **Eternal Texture Mods: Beginner's Guide**

Moving Assets

Information about moving assets from and to other patches.

Assets Info Redirect Links

assetsinfo JSON

Format Overview

Reference: "resources" Array