Dark Ages Level Modding

- **Eternal vs Dark Ages**
- **Essential Tools**
- **Hot Reloading**
- **MapExecutions Decls**

Eternal vs Dark Ages

This page covers the critical differences between Eternal and Dark Ages entities files.

It is assumed you have some experience with editing Doom Eternal entities files. While important differences do exist between the files, there are substantial similarities to Eternal. Check out the book on **Eternal Level Modding** if this subject is new to you. Most of the knowledge you'll gain there is applicable to Dark Ages.

Serialization

In Doom Eternal, entities files are plaintext and serialized by the game at runtime. In Dark Ages, they are already serialized. See **Serialization and Atlan Mod Packager** for the rules and restrictions they follow, compared to regular decls.

Submaps

The first difference you'll notice in a Dark Ages entities file, is that every entity has a number assigned to it. This value is it's submap index.

```
entity 3 {
<entity data>
}
entity 4 {
<entity data>
}
entity 4 {
<entity data>
}
```

Doom Eternal entity files were a single, long, uninterrupted list of entities. In Dark Ages, entities are split into multiple lists based on the submap they're located in. Each list starts with it's own `idWorldSpawn` entity named `world`.

idTech8 implements a "World Composition" system that divides the level's world into submaps, loaded and unloaded at will. Think about idTech7's **refmaps** - except they're no longer just a tool for developer workflow.

Due to this World Composition system, Dark Ages does not load every entity on map load. Only entities from the active submaps are loaded. The active submaps depend on your location in the level.

Identifying Submaps

Now that we know what submaps are, how do we figure out what region of a level they belong to? Many mapentities contain a couple dozen submaps, so it's important to figure this out.

There are a few techniques to help you figure things out.

1. Compare what entities are close by to a given location in the world. (Use EntitySlayer's spawnPosition filter!)
2. Check the names of the other entities in the submaps
3. Check the `entityPrefix` variable in the submap's `world` entity.

The following world entities come from `m4_siege.mapentities`

```
// Prefix: cin_locations_atlan_cockpit
// Conclusion: This submap is for the cinematic at the end of Siege Part 2
// (Many submaps exist solely for cinematics. This makes sense, since many cinematics
// load entirely different worlds you don't see during gameplay.)
entity 1 {
	entityDef world {
		inherit = "entityDef/worldspawn";
		expandInheritance = false;
		systemVars = {
			entityType = "idWorldspawn";
		}
		edit = {
			entityPrefix = "cin_locations_atlan_cockpit";
			levelSoundState = "soundstate/level_mix/none";
			globalAIsettings = "default";
			automapDecl = "automap/default";
		}
	}
}

// Prefix: s4_battlegrounds_north_cave
// Conclusion: This submap is for one of the 2 detached cave segments in Siege Part 1
// Further analyze the submap's other entities to determine which cave it's for.
entity 14 {
	entityDef world {
		inherit = "entityDef/worldspawn";
```

```
	expandInheritance = false;
	systemVars = {
		entityType = "idWorldspawn";
	}
	edit = {
		entityPrefix = "s4_battlegrounds_north_cave";
		levelSoundState = "soundstate/level_mix/none";
		globalAIsettings = "default";
		automapDecl = "automap/default";
	}
}
}
```

Adding New Entities

When adding new entities to a file, you must assign them a submap index based on what part of the level they'll be appearing in. Use the above techniques to identify the submaps relevant to the part of the level you want to edit. Then, give your modded entities the appropriate submap index.

Only use a submap index that already exists in the vanilla file.

Make sure a submap's `world` entity stays listed before any of it's other entities

Submap 0

Entities in most submaps will be loaded/unloaded based on where you are in the level. But Submap 0 functions as the "persistent level". Entities placed in submap 0 will always be loaded and active.

Multi-Level Entities

As a consequence of this enhanced loading system, levels are a lot bigger. So big, that sometimes two different levels' entities are contained in the same file! For example:

- *Siege Part 1* and *Siege Part 2* are both part of `mapentities/maps/game/sp/m4_siege/m4_siege.mapentities`.
- *Abyssal Forest* and *Ancestral Forge* are both part of `mapentities/maps/game/sp/m5_forge/m5_forge.mapentities`

Layers

DOOM Eternal made constant use of the layer's system. It still exists in Dark Ages, but only for specific purposes.

If an entity has a layer, it will look something like this:

```
entity 14 {
	layerIndex = 1;
	layers = {
		"spawn_target_layer"
	}
    entityDef some_entity { ... }
}
```

Dark Ages entities can only be added to ONE layer. The `layerIndex` property is important. You may think of it as the "ID" for a layer. It must be included when specifying a layer. **The same layer may have a different layerIndex value across different submaps.** The consequences of using a layer whose layerIndex is not already defined in the submap, is unknown.

End-of-File Data

If you scroll down to the bottom of a mapentities file, you'll see a massive blob of encoded data:

```
headerchunk {
	<massive amount of alphabetical data>
}
```

This is the header chunk of the original file. It is there intentionally, and **you must not touch it**. This data is critical for correctly serializing the file.

Essential Tools

Tools you should use when making Dark Ages level mods.

EntitySlayer

An editor for Doom Eternal and Dark Ages mapentities files. **Download EntitySlayer here.**

EntitySlayer has a number of useful tools that streamline the level modding process. Recent updates have added a diff-checking system that automates updating your level mods after a game update!

Check out EntitySlayer's **README** for more information.

Kaibz Mod

Kaibz Mod is the Dark Ages equivalent to Doom Eternal's Meathook mod. It adds a number of console commands useful to level modding. **Download Kaibz Mod here.**

Command	Description
k_spawninfo	Copies the player's spawnPosition and spawnOrientation to your clipboard. This command is equivalent to `mh_spawninfo` from Doom Eternal's Meathook mod. Currently, this command only works when playing as the Slayer.
k_activeEncounters	Prints a list of active idEncounterManagers to the console. This is extremely helpful for figuring out what encounter managers are orchestrating the current combat encounter.
k_noclip	Toggles noclip
k_hotReload	See **Hot Reloading** for instructions on the proper usage of this command.
setviewpos <x> <y> <z>	Teleports you to the desired location. The vanilla `teleportposition` command does not work in Dark Ages. Use this instead. (This is a vanilla command, but it's included here for reference)

Connecting EntitySlayer to Kaibz Mod

Doom Eternal's Meathook mod possesses an RPC interface. This lets other processes communicate with the game to execute console commands. EntitySlayer uses this interface to enable several convenient editor features, like setting an entity's spawnPosition from the player's current position.

Kaibz Mod offers equivalent functionality using a Windows Pipe interface. To enable this feature:

1. Launch the game, and open the Kaibz Mod menu by pressing the `F8` Key.
2. Open the `MISC.` menu. In that menu, set `Enable Mod Interface` to `YES`
3. You're good to go! Kaibz Mod saves your settings, so you don't need to turn the interface on every time you launch the game.

> Kaibz Mod does not support every feature offered by Meathook. A small number of options in EntitySlayer still won't work because of this.

Hot Reloading

How to use Atlan Mod Loader and Kaibz Mod to reload modded entities files without restarting the game.

> This is an experimental feature. It may or may not be broken on Linux. If you attempt this procedure, please report whether it's successful or not.

The biggest problem with editing level mods is iteration time. Each time you want to test new changes, you must re-install the mod and restart the game. It's a time-consuming process that grinds down development. But with the power of Kaibz Mod and Atlan Mod Loader, this inconvenience can be eliminated!

Install Kaibz Mod

You must install **Kaibz Mod** to perform hot reloading! It is recommended you enable it's Mod

Interface while developing level mods. See **Connecting EntitySlayer to Kaibz Mod** to learn how to enable the mod interface.

Before Launching the Game

Atlan Mod Loader will engage "Hot Reload Mode" when an **unzipped and unpackaged** .mapentities file gets loaded. Let's say you want to edit the mission *Hebeth*. Your mods folder might look like this.

```
- Steam/steamapps/common/DOOMTheDarkAges/
   - mods/
      - my_hebeth_mod/
         - mapentities@maps@game@sp@m2_hebeth@m2_hebeth.mapentities
         - darkagesmod.txt
```

Once this is setup, run Atlan Mod Loader and launch the game.

> While hot reloading, you can have other mod files installed besides the mapentities you're editing. However, you must NOT edit any of these files. You must NOT remove any of these files, or add any new files. If you want to do any of these things, you must shutdown the game then re-run the mod loader. Otherwise, hot reloading will break, and your game will most likely crash.

Reloading the File

Let's say you've made some changes to the Hebeth mapentities file and you're ready to reload it. Run the console command `k_hotReload` to reload the map!

Notes:

1. This may alt-tab you out of the game for several seconds. The alt-tabbing is most severe when running the game in `Full Screen` mode. If it gets annoying, try changing to `Borderless Windowed` mode.
2. You may notice a command prompt briefly pop up and close. This is Atlan Mod Loader running due to Kaibz Mod.

Quick Test

The first time you attempt hot reloading, you should do something simple to verify it works on your system. The following code snippet will place a floating text entity at the start of Hebeth.

1. Paste this into Hebeth's mapentities file. **Make sure you paste it somewhere AFTER submap 0's world entity.**
2. Edit the entity's `edit/headerText/text` property.
3. Trigger a hot reloading using the above procedure.
4. Verify that the text you see in-game has changed to reflect your edits.

```
entity 0 {
	entityDef hot_reload_test {
		inherit = "entityDef/gui/text";
		expandInheritance = false;
		editorVars = {
			placeable = false;
		}
		systemVars = {
			entityType = "idGuiEntity_Text";
		}
		edit = {
			headerText = {
				text = "Hot Reload Test!";
```

```
	}
	canvasFile = "ui/shapes/worldgui_text";
	useSWFTransform = true;
	swfScale = 0.016667;
	spawnPosition = {
		x = -1245.702881;
		y = -1230.202271;
		z = -77.942009;
	}
	spawnOrientation = {
		mat = {
			mat[0] = {
				x = -0.000000;
				y = -1.000000;
			}
			mat[1] = {
				x = 1.000000;
				y = -0.000000;
			}
		}
	}
	flags = {
		noknockback = false;
	}
	renderModelInfo = {
		model = "editors/models/gui_text.lwo";
		scale = {
			y = 15.000000;
		}
	}
	clipModelInfo = {
		type = "CLIPMODEL_NONE";
	}
  }
 }
}
```

Reloading the File (Without Kaibz Mod)

> Hot Reloading without Kaibz Mod is NOT RECOMMENDED! This section is here purely for documentation purposes! Please use the above method instead of this one!

To Hot Reload without using Kaibz Mod, perform the following steps:

1. Alt-Tab out of the game.
2. Run Atlan Mod Loader and wait for it to complete
3. Quit out to main menu, then load back into Hebeth using Mission Select.
4. The changes you've made to the file should be reflected in-game.

MapExecutions Decls

Location: `rs_streamfile/generated/decls/mapexecution/`

These decls are new to idTech8, and define object pools for hundreds of different entity types. It's unknown how important they are compared to the `aipoolnumbers` decls. If you ever run into a problem with entities not spawning or the game crashing when too many of an entity are spawned, consider experimenting with these files.