Atlan Fundamentals

The basics of using Atlan Mod Loader to make mods

- **Atlan Mod Loader vs Eternal Mod Injector**
- **Adding Files to your Mods**
- **Atlan Config File**
- **Unzipped Mod Folders**

Atlan Mod Loader vs Eternal Mod Injector

Atlan's approach to mod-loading is fundamentally different from Eternal Mod Injector's. The major limitations of Eternal Mod Injector do not apply to Atlan.

Eternal Mod Injector	Atlan Mod Loader
Replaces files inside of the vanilla game archives	Builds it's own resource archive. The vanilla game archives are untouched
Modders had to know what archives contained which assets, and replicate this setup in their mods.	Not an issue. The modded resource archive is loaded globally, so modded assets will always be loaded.
Adding new assets to an archive, instead of replacing existing assets, could cause instability.	Not an issue. Add as many new assets as you want!
Game updates frequently broke mods due to resource archive priorities changing.	Not an issue. The modded resource archive will always have the highest priority.

Atlan does not make your mods immune to game updates. When a game update edits a vanilla file, you should update your mods to use that new version as a baseline. The dangers of outdated mod files vary based on their importance. Consequences can range from nothing happening, to core mechanics breaking, to your game crashing on startup.

What files are modified?

Atlan Mod Loader edits the following files:

```
The game executable file (i.e. DOOMTheDarkAges.exe)
base/packagemapspec.json
base/meta.resources
base/sound/soundbanks/pc/soundmetadata.bin
base/build-manifest.bin (DOOM: Eternal only)
```

The modded resource archive is `base/modarchives/common_mod.resources`

The modded StreamDB archive is `base/modarchives/common_mod.streamdb`

The modded audio archive is `base/sound/soundbanks/pc/ATLANMOD.snd`

Adding Files to your Mods

Before building your mods, you must use **Atlan Resource Extractor** to obtain the game's files. **Please thoroughly read the information on the download page and in it's configuration file.**

To add a vanilla file to your mod:

1. Copy it from the extracted vanilla resources into your mod folder
2. Replicate the filepath produced by Atlan Resource Extractor

Example: Let's say you copy the following vanilla file into your mod folder

```
decls/string/ui/mainmenu/collectible_viewer.decl
```

You must replicate this filepath in your mod folder. This should be it's path when your mod is zipped up.

Fake Directories

A goal of Atlan Mod Loader is improving the developer experience when making mods. One way it does this, is by making directories optional. Instead of creating *literal* directories, add an `@` symbol in the filename. The mod loader translates these symbols into directories for you!

Example: Let's say we want to make a mod using the file:

`decls/string/ui/mainmenu/collectible_viewer.decl`

This file is nested inside of 4 directories! That's a real pain to create and navigate through each time you need to open the file! Instead of creating those directories, just name the file:

`decls@string@ui@mainmenu@collectible_viewer` `.decl`

You can also mix and match! For Example:

1. Have a folder named `decls@strings`
2. Name the file `ui@mainmenu@collectible_viewer` `.decl`

Find a setup and style most convenient for you and the mod you're making!

Example Mod

You can download an **example mod** that illustrates the "Fake Directory" concept. It also includes an example of the **mod config's** aliasing system. All files in this example mod replace the same game asset.

Atlan Config File

Atlan Mod Loader uses `darkagesmod.txt` as the configuration file for DOOM: The Dark Ages mods. If you're making a mod for DOOM: Eternal, you should use `atlanmod.txt` as your file name. Including a config in your mod zip is optional, but highly recommended!

The config file consists of several optional properties.

ModInfo Chunk

The ModInfo properties are the only ones that should be included in *every* config file. Here is an example:

```
modinfo = {
    name = "Your Awesome Mod"
    game = "Doom The Dark Ages"
    author = "Your Name Here"
    description = "Your Mod Description"
    version = "3.0"
    loadPriority = 1
    requiredVersion = 1
}
```

Property Name	Description
name	Your mod's name. Displayed in the Mod Manager
game	What game this mod is made for. Atlan will not install *DOOM: Eternal* mods in *DOOM: The Dark Ages* and vice versa
author	Author information. Displayed in the Mod Manager
description	Your mod's description. Displayed in the Mod Manager
version	Your mod's version. Displayed in the Mod Manager
loadPriority	If multiple mods edit the same file, their load priority is used to help resolve conflicts. Mods with a lower load priority override mods with a higher load priority

requiredVersion	The version of *Atlan Mod Loader* you must be running to load this mod. • Dark Ages Decl mods require a minimum version of 1 • Dark Ages Entitydef and Logic Decl mods require a minimum version of 2 • Dark Ages Mapentities mods require a minimum version of 4 • Dark Ages Sound Mods require a minimum version of 5 • Dark Ages Texture Mods require a minimum version of 6 • Dark Ages Font Mods require a minimum version of 7 • Dark Ages Mods utilizing assets-info require a minimum version of 9 • All DOOM: Eternal mods require a minimum version of 10

Aliasing Chunk

The aliasing block is optional. It is meant to help with mod files whose paths are extremely long and convoluted. You may choose to name the files something simple, then alias them to their real asset paths.

The format for an alias is `"Real Path In Zip File" = "Game Asset Path"`

```
aliasing = {
    "my_first_decl.decl" = "decls/ability_dash/my_ability_dash_decl.decl"
    "other/my_second_decl.decl" = "decls/ability_dash/default.decl"
}
```

Assets, MapSpec, and MapResources Chunks

These chunks are used for assets info manipulation. These are advanced properties that require some experience to use correctly. Their usage is explained in-depth on **this page**

Load Level Chunk

This chunk enables idStudio level mods that were installed by Atlan. You can use it to load any number of level mods.

```
// This example causes two idStudio level mods to be loaded
// Their respective installation paths would be:
// 1. base/modarchives/coop_hordamex/
// 2. base/modarchives/pvp_hurricane/

load_level {
	coop_hordamex
    pvp_hurricane
}
```

Atlan Mod Packager requires this property to be set when packaging an idStudio level mod.

Total Overhaul Property

This property should only be used under very rare circumstances. Does your mod touch *hundreds* of files or edit dozens of levels? Does this make it fundamentally incompatible with other mods of similar size? You may wish to designate it as a "total overhaul" mod. Atlan will only allow one Total Overhaul mod to be loaded at a time.

```
modinfo {
	total_overhaul
}
```

Elevated Gameplay Property

This property only exists for DOOM: Eternal multiplayer gamemode mods. It shouldn't be used anywhere else. When set, only your mod can load gameplay-affecting files. Any gameplay-affecting files in other mods will fail to load. Only one Elevated Gameplay mod can be loaded at a time. The intent of this feature is to prevent players from accidentally cheating in competitive modded multiplayer by unknowingly loading other gameplay-affecting mods.

```
modinfo = {
	elevated_gameplay
}
```

Unzipped Mod Folders

When developing your mods, you don't need to zip and package them each time you want to test them! To promote fast iteration, Atlan Mod Loader will load unzipped mod files according to the following rules:

1. Each folder inside your mods folder is considered an unzipped mod
2. If a folder's name begins with a `$` then it is ignored.

For example, if your mods folder has the following contents inside it:

```
Steam/steamapps/common/DOOMTheDarkAges/
   - mods/
      - MyTextureMod/
            - darkagesmod.txt
            - MyImage.png
      - CombatBalance/
            - DeclA.decl
            - DeclB.decl
            - darkagesmod.txt
      - $RandomStuff/
            - Stuff.decl
      - CoolMod.zip
      - AwesomeMod.zip
```

When Atlan Mod Loader is ran, it will load two zipped mods, and two unzipped mods. Since the unzipped mod folder `$RandomStuff` begins with a `$` it will be ignored!

Other Benefits

Unzipped Mods have the following benefits:

1. Dark Ages: Unserialized Mapentities, logic decls, and entitydefs will be serialized
2. Eternal: Uncompressed entities files will be compressed.
3. Both Games: Raw PNG files will be encoded into image assets. The encoding uses a lower compression level than Atlan Mod Packager, so this process will happen quicker!

Zipped mods do not receive these benefits. Atlan Mod Loader expects zipped mod files to be serialized, compressed, or encoded in advance! Remember to use Atlan Mod Packager before distributing mods containing these files!