Grim Dawn Modding Suite

Version 0.9

Developed by Cenorayd

Changelog:

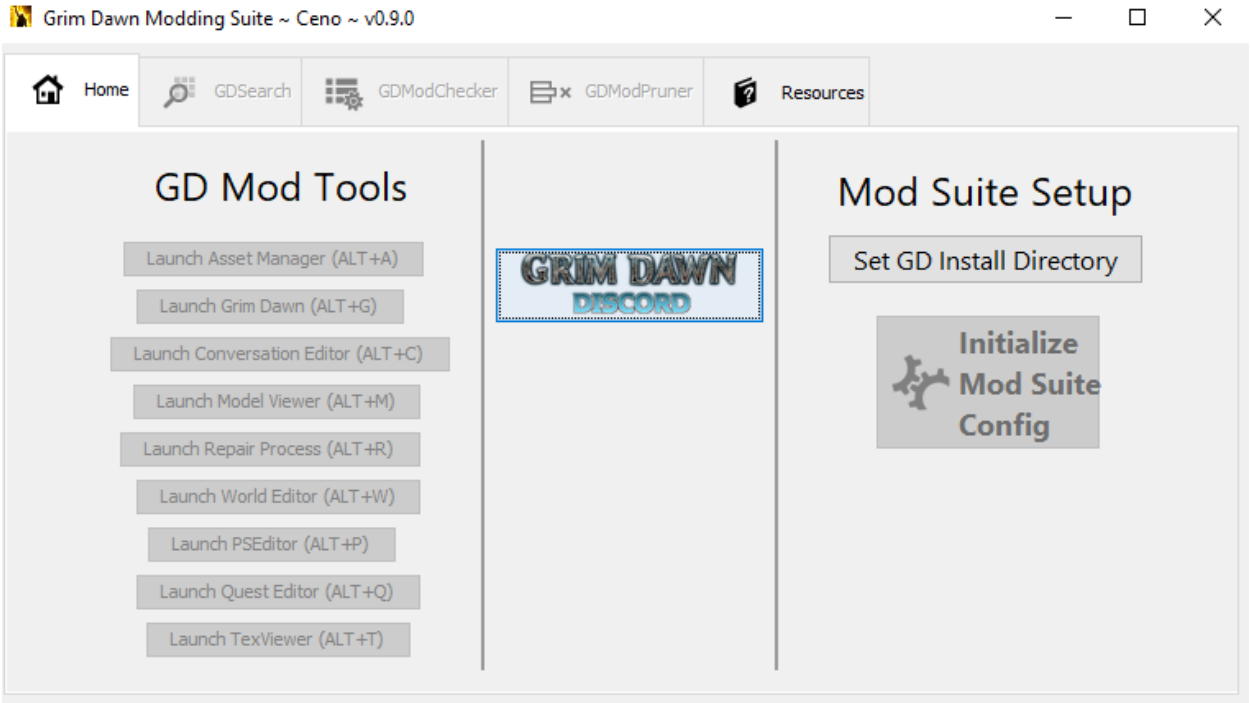
V0.9 – Initial Release of Documentation

Preamble

Welcome to the documentation file for the unofficial Grim Dawn Modding Suite (GDMS), developed by Cenorayd (Ceno). This file is intended to serve the purpose of both an introductory tutorial and a reference manual for using the application. GDMS is a Java application (requiring an updated version of Java 21). Finally, GDMS is designed to integrate with Grim Dawn data/mod files configured such that one's working directory is also their source directory. Support for separate directories thereof is TBD at a later time.

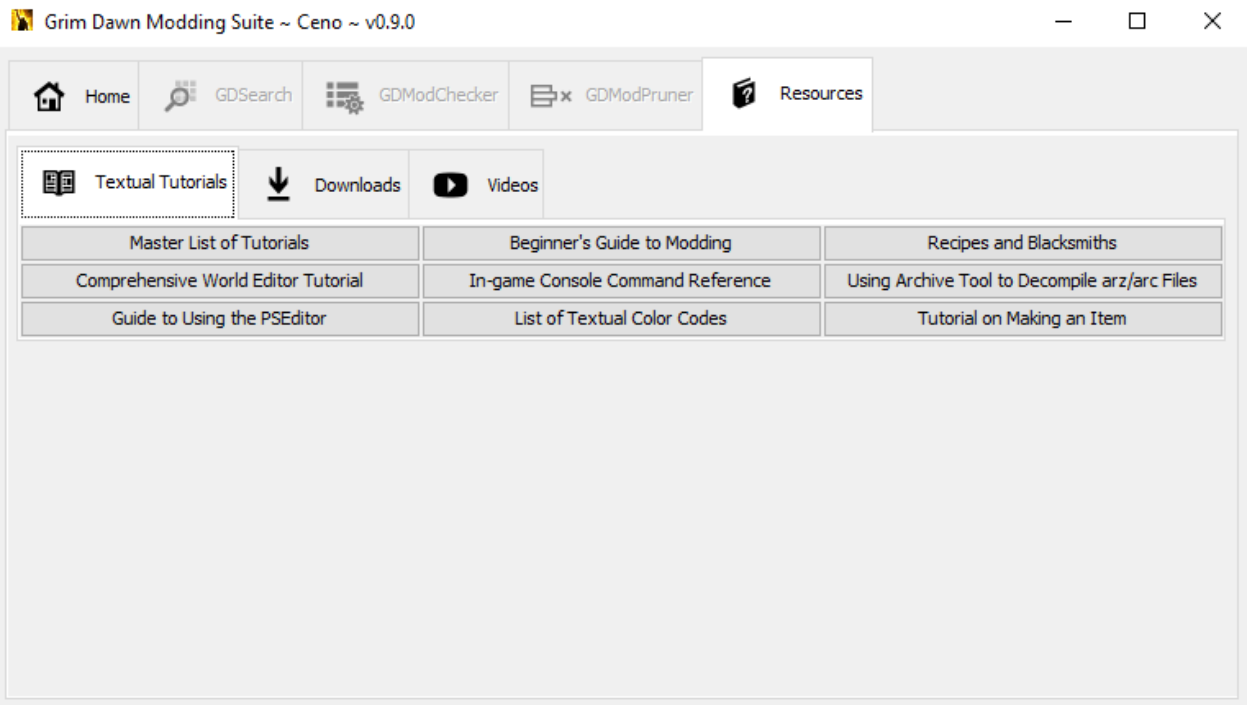
Using the Program

When one first launches a fresh download of GDMS, they will be met with a screen like this:



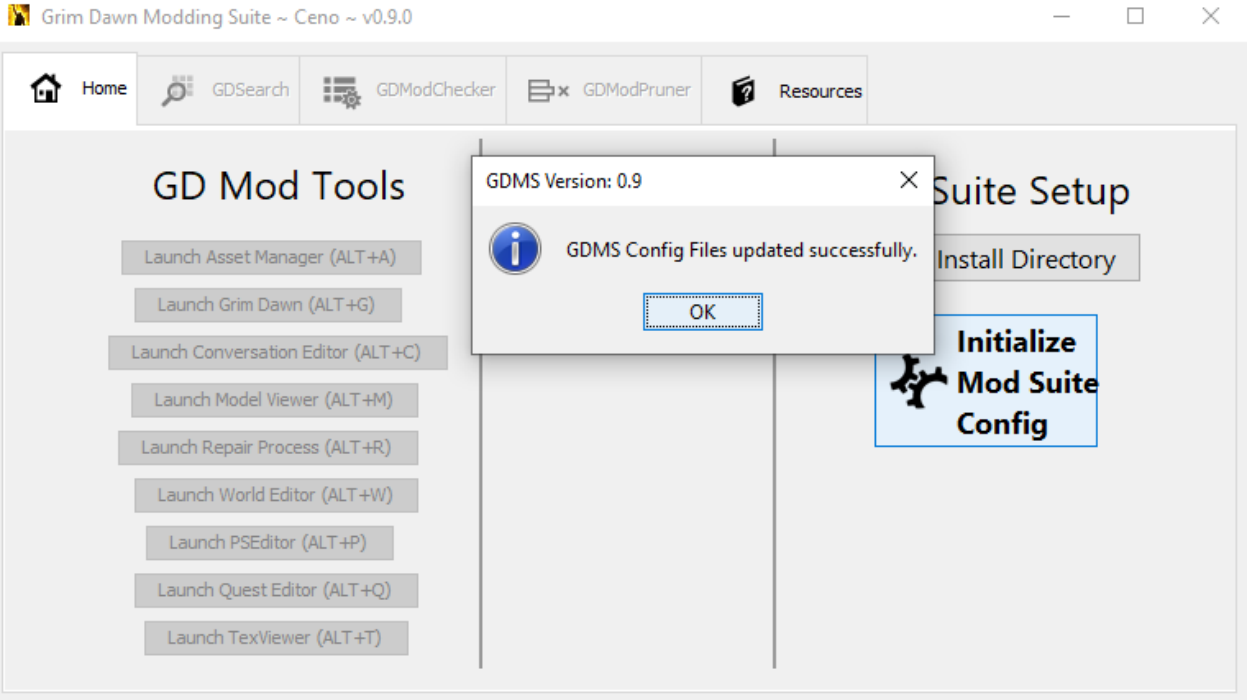
(1 – home.PNG)

Note that most features are inaccessible at first—only the Home and Resources tabs can be accessed, as well as a link to the (unofficial) Grim Dawn Discord where users can frequently find me and other modders with which to chat. The Resources tab includes links to many 3rd-party modding links, most of which lead to the official Grim Dawn forums, that offer further tutorials or downloads to expedite the modding process:



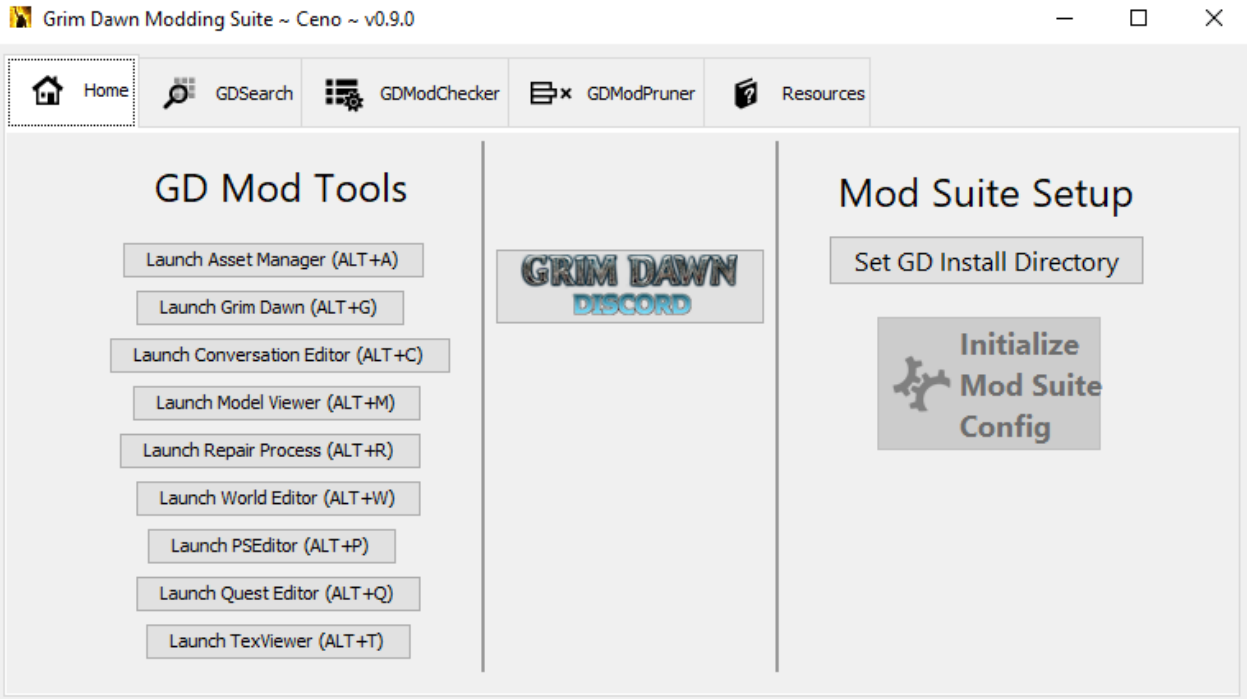
(2 – resources.PNG)

In order to begin utilizing the program in full, one must first tell GDMS where their Grim Dawn Installation Directory is on their PC by clicking the "Set GD Install Directory" button and navigating to the directory in question. Upon doing so, the "Initialize Mod Suite Config" button becomes available. Clicking it will make the program pause for a moment as it processes the necessary data for its continued use. When it is done, users will be given the following popup:



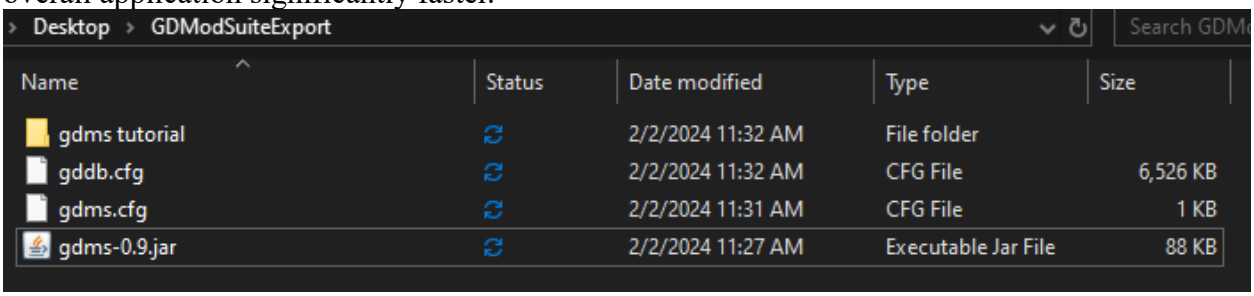
(4 – home.PNG)

And the rest of GDMS will become operable:



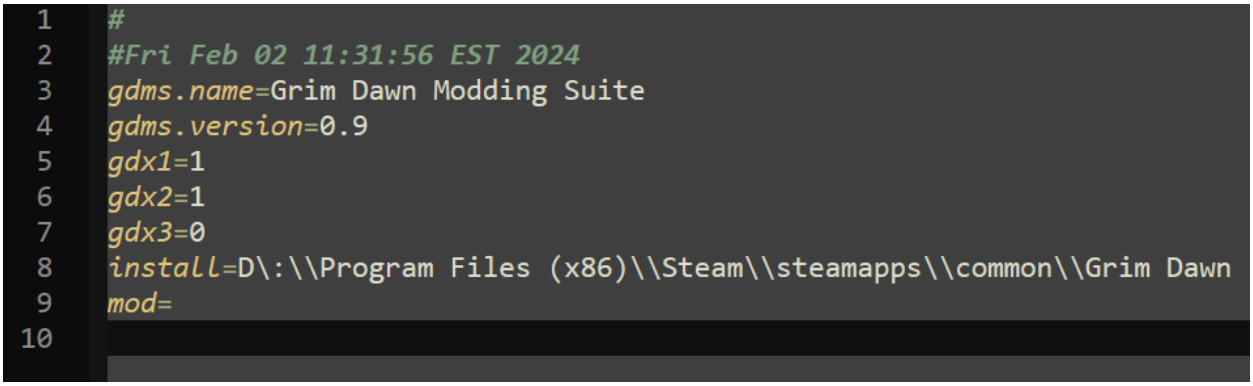
(5 – home.PNG)

Externally, GDMS will have created two configuration files in the directory in which it is being run from. It is NOT recommended to edit these files outside of the program; they will be used for all remaining program features and will be changed as necessary by GDMS itself. These files are simply used to expedite the process of using GDMS between sessions and make the overall application significantly faster.



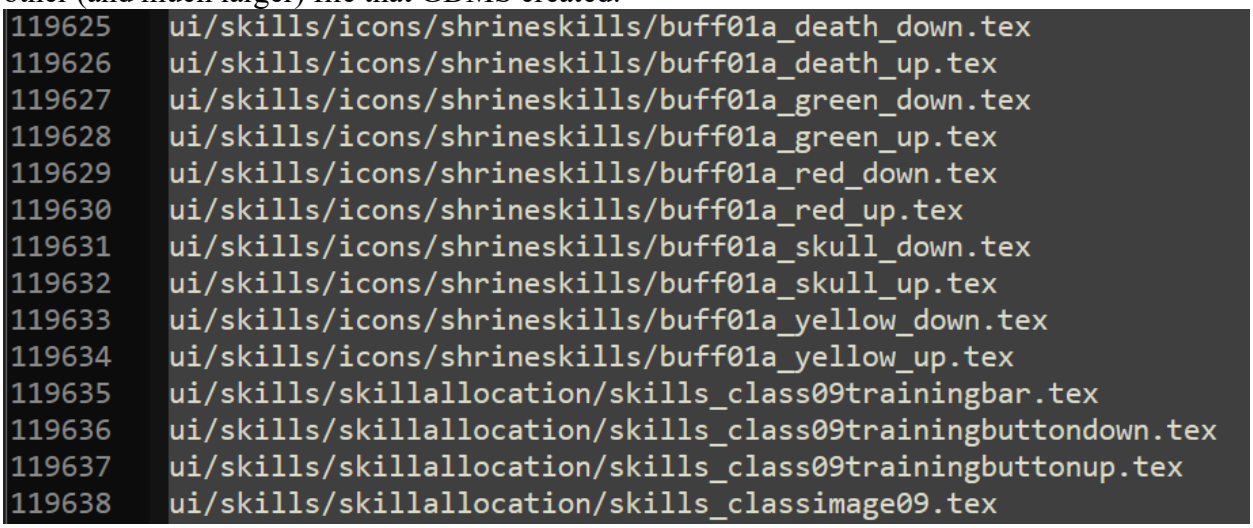
(6 – gdms files.PNG)

Let's look inside these files to understand their contents and why they are important to keep intact:



(7 – gdms cfg.PNG)

gdms.cfg is a configuration file containing relevant working info. It is used to verify the integrity of GDMS itself as well as inform the user of the application's version number on certain menus. More importantly, it describes which files GDMS should be working with—note that GDMS identified which Grim Dawn expansions are currently installed and extracted on my system. (gdx1 and gdx2 are installed, gdx3 is not) This can be changed in certain menus to enable or inhibit GDMS from interacting with certain expansion content. While editing this file externally is not recommended, it should not be too dangerous if one wishes to pre-configure GDMS before use. Note that the "mod" field is, at this time, empty. Let's take a quick look at the other (and much larger) file that GDMS created:

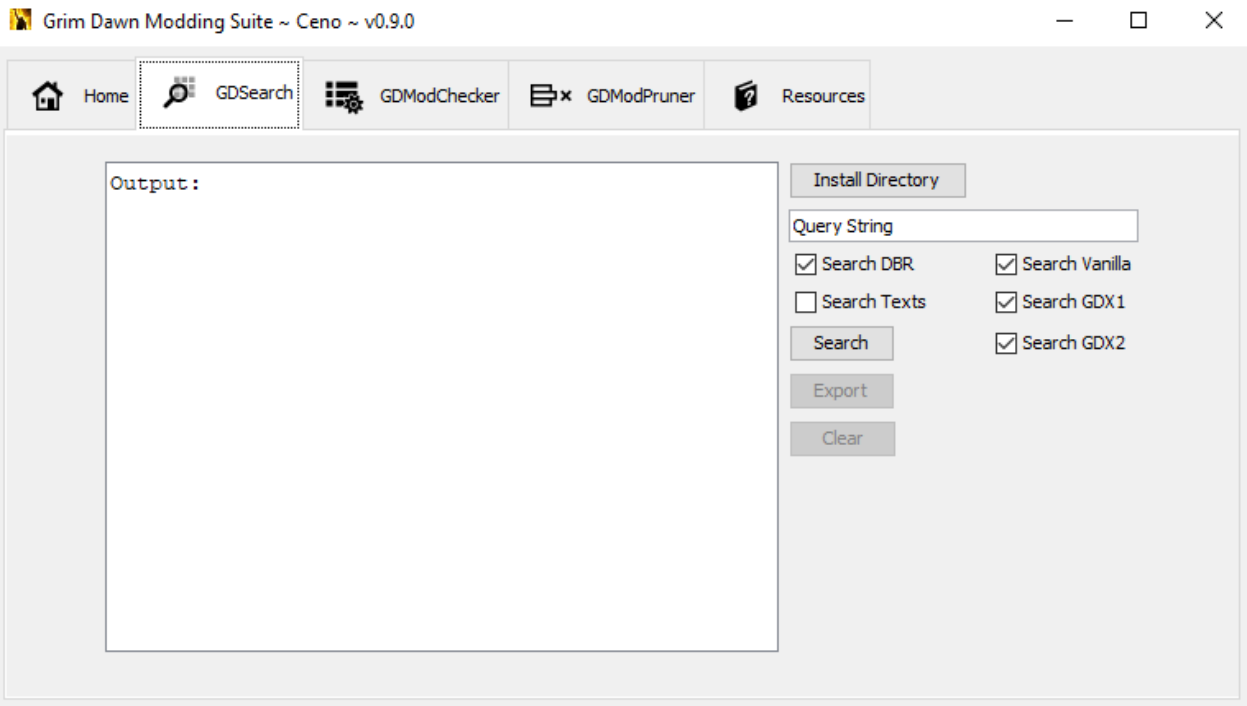


(8 – gddb cfg.PNG)

gddb.cfg is a *long* file listing the contents of all database and resource files used in Grim Dawn made by Crate Entertainment. This file is referenced as a quick, in-place means of comparing mod contents to those present in unmodded files. It is *extremely* recommended to ***never*** edit this file. If one changes their Grim Dawn installation, either by moving it or installing new content from Crate Entertainment (such as a new expansion), this file will be regenerated by GDMS whenever the "Initialize Mod Suite Config" button is pressed on the Home menu.

Let's move on to actually utilizing the newly-enabled features offered within GDMS, starting with GDSearch.

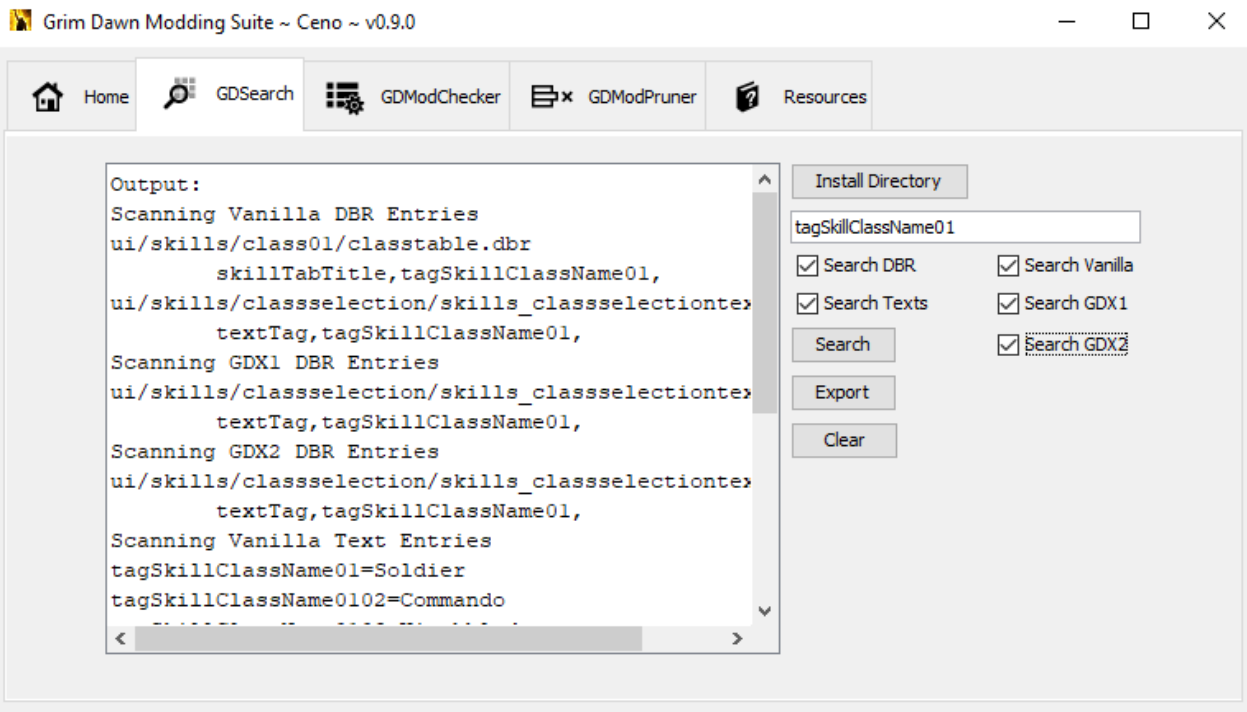
GDSearch



(9 – gdsearch.PNG)

GDSearch is a tool used for searching for specific strings within the content provided by Crate Entertainment. Note that it does *not* search through modded content. One can search through .dbr (database record) and .txt files in vanilla Grim Dawn content, GDX1 content, GDX2 content, or GDX3 content. Note that the checkboxes for expansion content only become visible if the expansion is detected by the GDMS initialization process; I do not have GDX3 installed, so the checkbox for GDX3 is not displayed.

Upon entering a Query String in the textbox above and hitting the "Search" button, GDMS will perform a scan of that string within any files earmarked for searching through. The results will be displayed in realtime in the large textbox labeled 'Output:'. For instance, if looking for "tagSkillClassName01" (the text tag for the Soldier mastery name) in all installed Grim Dawn content, the output would look like this:

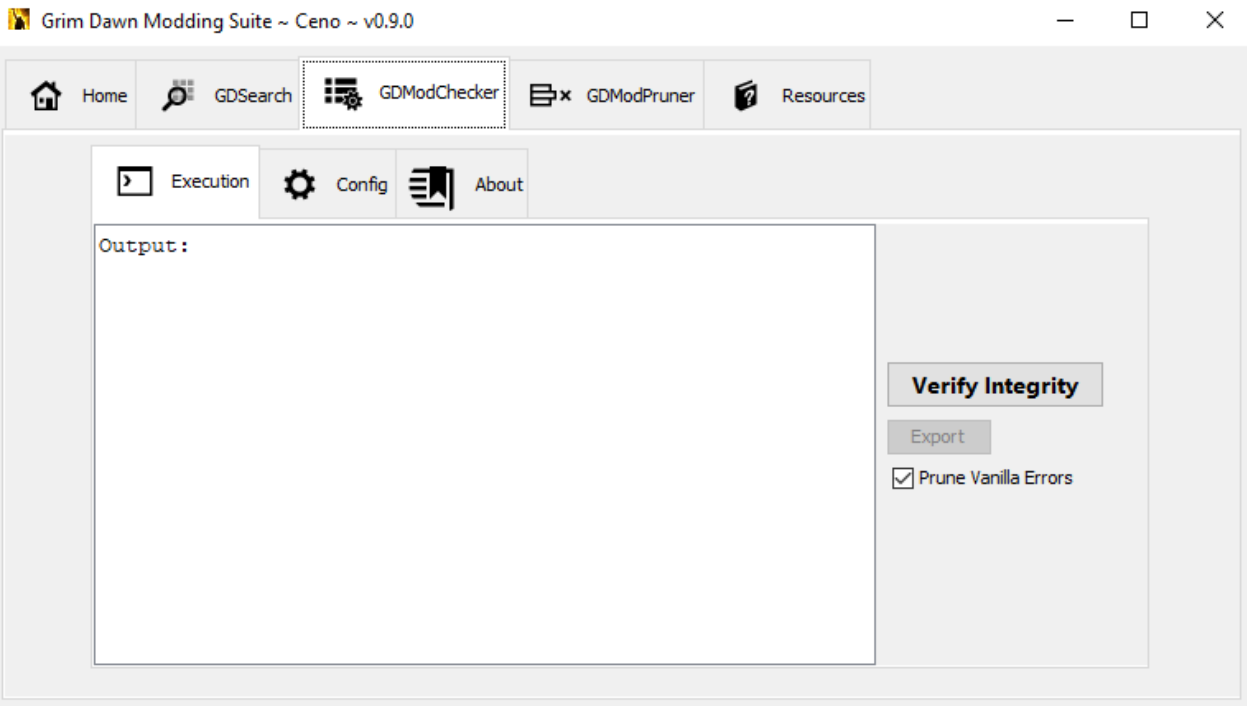


(9.1 gdsearch example.PNG)

The Output displays both the file names – in the case of .dbr files – in which the Query String was identified as well as the full line containing that string. Note that in the image above, a bug had been present which omitted the 'records/' (or any database subdirectory) part of a filename—this has been corrected in the release version.

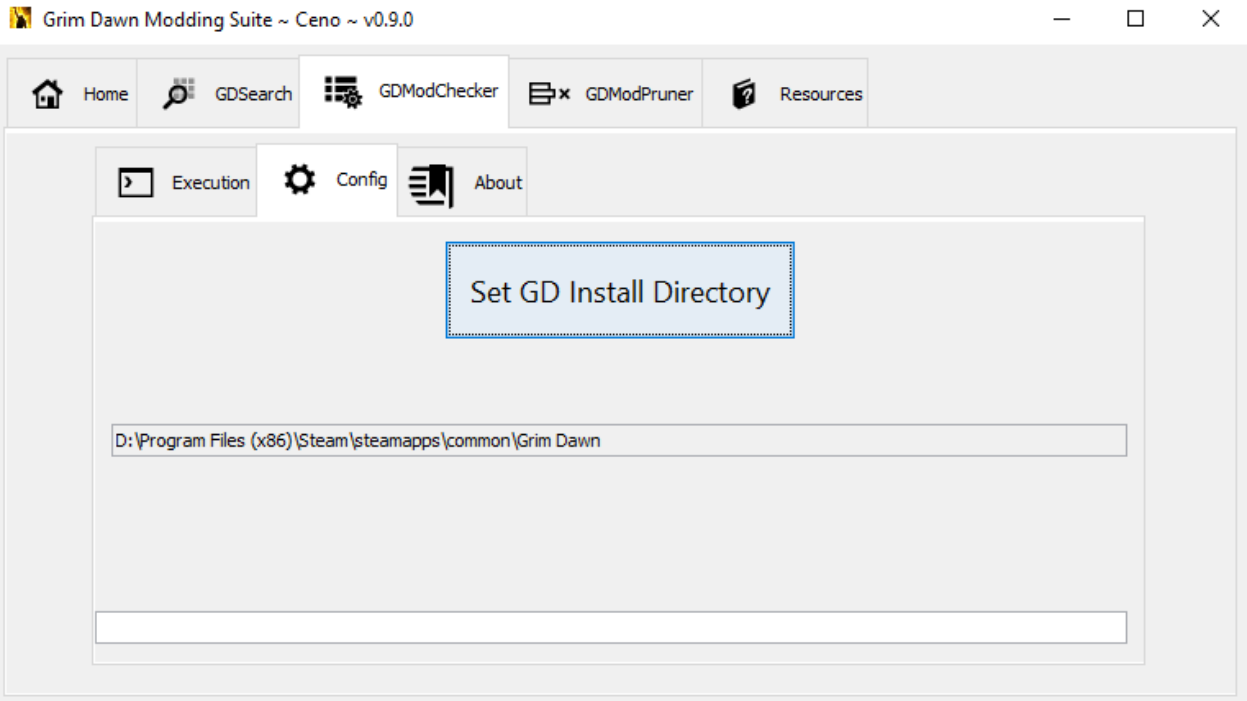
If one so desires, they can export the results of a finished search by clicking the Export button. GDMS will create a text file containing the search output within the 'search/' subdirectory wherever the application is installed. Alternatively, one can clear the Output textbox with the Clear button, though this is not strictly necessary, as performing new searches will refresh the Output textbox anyway.

GDModChecker



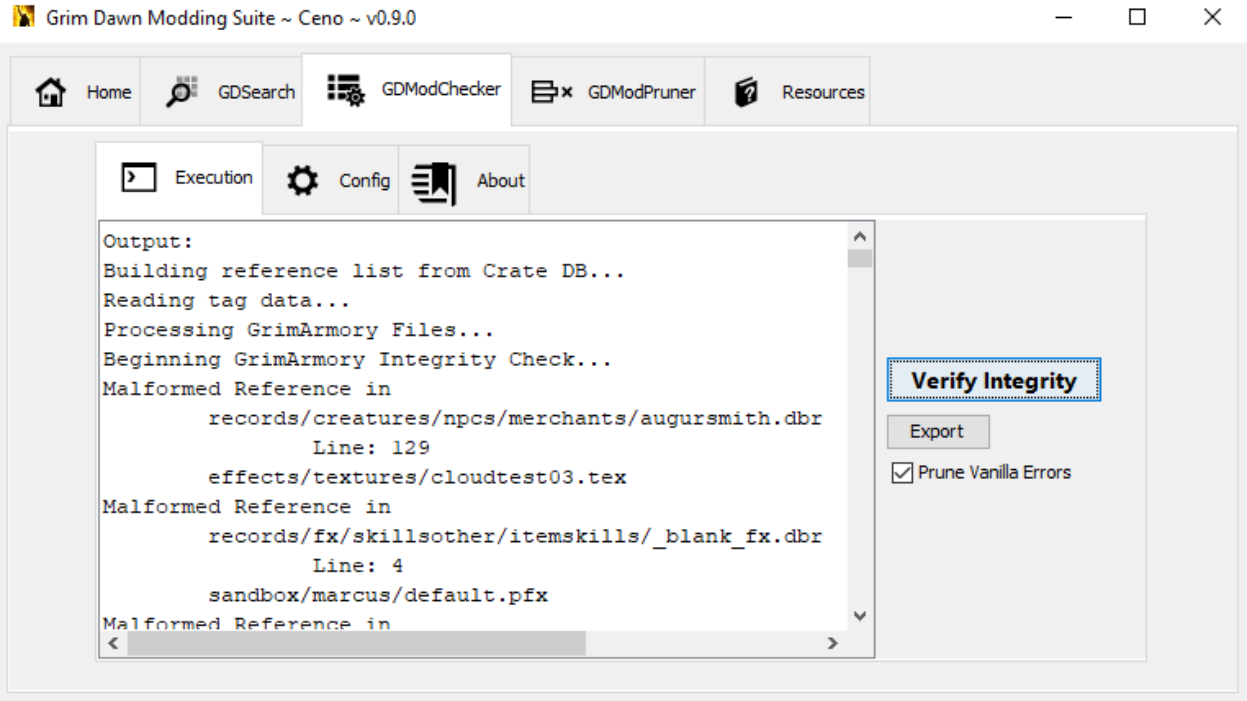
(10 – modcheck exe.PNG)

GDModChecker is an application which scans the database content of a selected mod for possible mistakes, such as Malformed References (a link to a file which does not exist/has its path misspelled) or Missing Tags. GDModChecker, and GDModPruner like it, looks a little bit different from the layout of GDSearch—notably in its separation of Execution, Config, and About menus. Let's start with the Config menu, as the program cannot operate successfully without one key piece of information…



(11 – modcheck cfg.PNG)

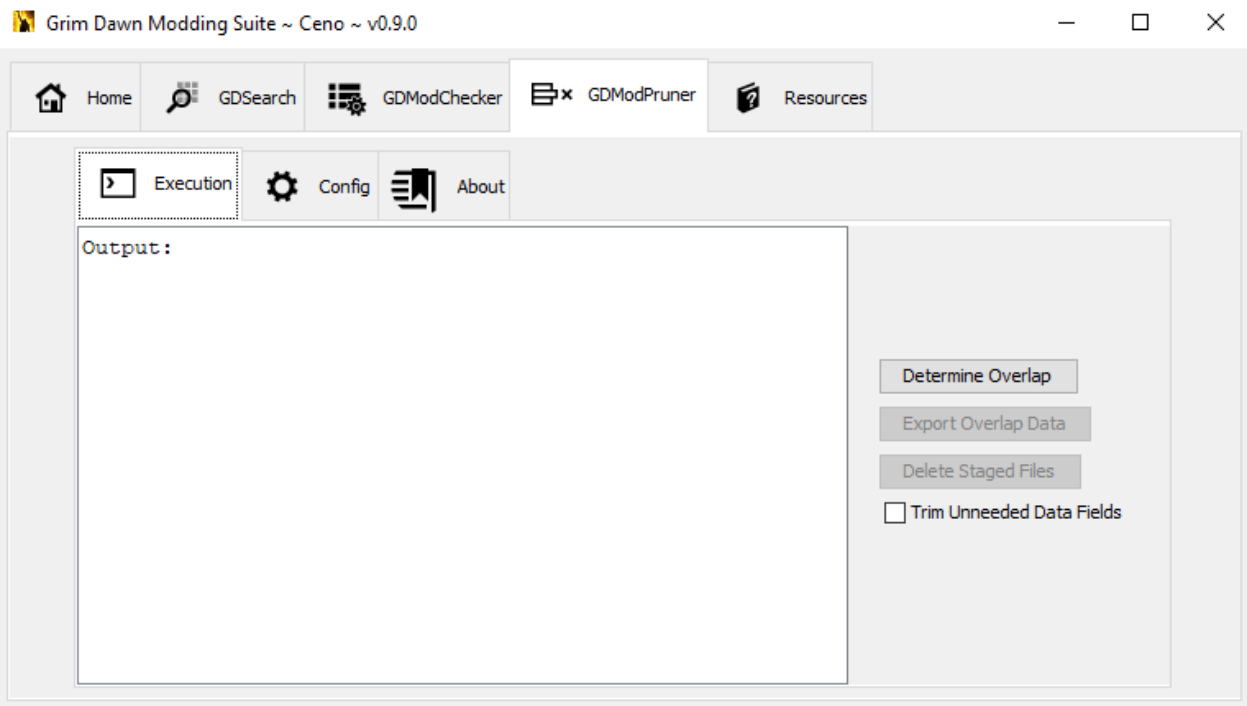
You'll notice that, for display purposes, our Installation Directory is displayed in the center of the Config menu. If we wish to change this, a button is available to do so—though one may prefer to do so from the Home tab. More importantly for the GDModChecker, however, is the empty textbox at the bottom of the Config menu, where one should type the name of the mod (as it is displayed in the Grim Dawn/mods/ folder) which they wish to check the integrity of. For the purposes of this example, I will type the name of my own mod, GrimArmory. Upon doing so and returning to the Execution menu, we are ready to actually use GDModChecker, by pressing the "Verify Integrity" button. Depending on whether we want to ignore errors that occur in both our files and Crate's files—meaning, errors we inherited by modifying existing Crate files rather than making new files ourselves—we can check the 'Prune Vanilla Errors' checkbox.



(12 – modcheck exe example.PNG)

As with GDSearch, results from an integrity check of our mod (GrimArmory, in this example), are displayed in realtime in the Output textbox. A report of the sum total of errors of each respective type (Malformed References, Missing Tags, etc.) is given at the bottom of the Output textbox. As with GDSearch, errors are reported with the filename in which the error occurs, the line number of the error, and the specific triggering error in question. As with GDSearch, we can also export the results of our integrity check to a local text file; these files will be created in the 'check/' subdirectory.

GDModPruner

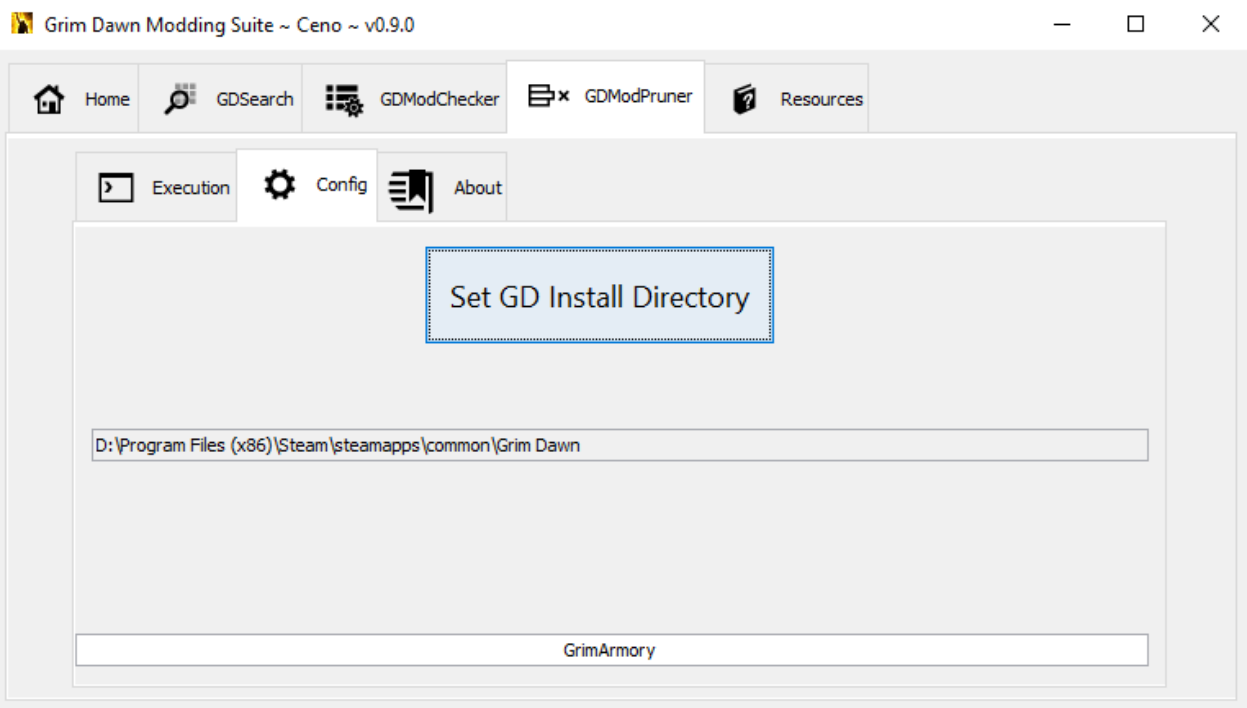


(13 – modprune exe.PNG)

GDModPruner is an application which scans mod files and stages those which do not change anything from their vanilla counterparts for deletion. It optionally also trims irrelevant datafields from database records, shrinking overall mod size without compromising mod functionality. As of GDMS Version 0.9.2, GDModPruner also compares mod resource files against those designed by Crate Entertainment, and stages them for deletion alongside overlapping database records.

Usage of GDModPruner is considered volatile, in the sense that it can and will change or remove files in your mod! Therefore, it is heavily recommended to make a backup of your mod files before use.

Much like GDModChecker, GDModPruner has an important Config menu that needs to be filled out prior to use. However, because we last verified the integrity of the GrimArmory mod in the exploration of GDModChecker, GDModPruner's Config menu is already updated with the same mod being targeted for pruning:



(14 – modprune cfg.PNG)

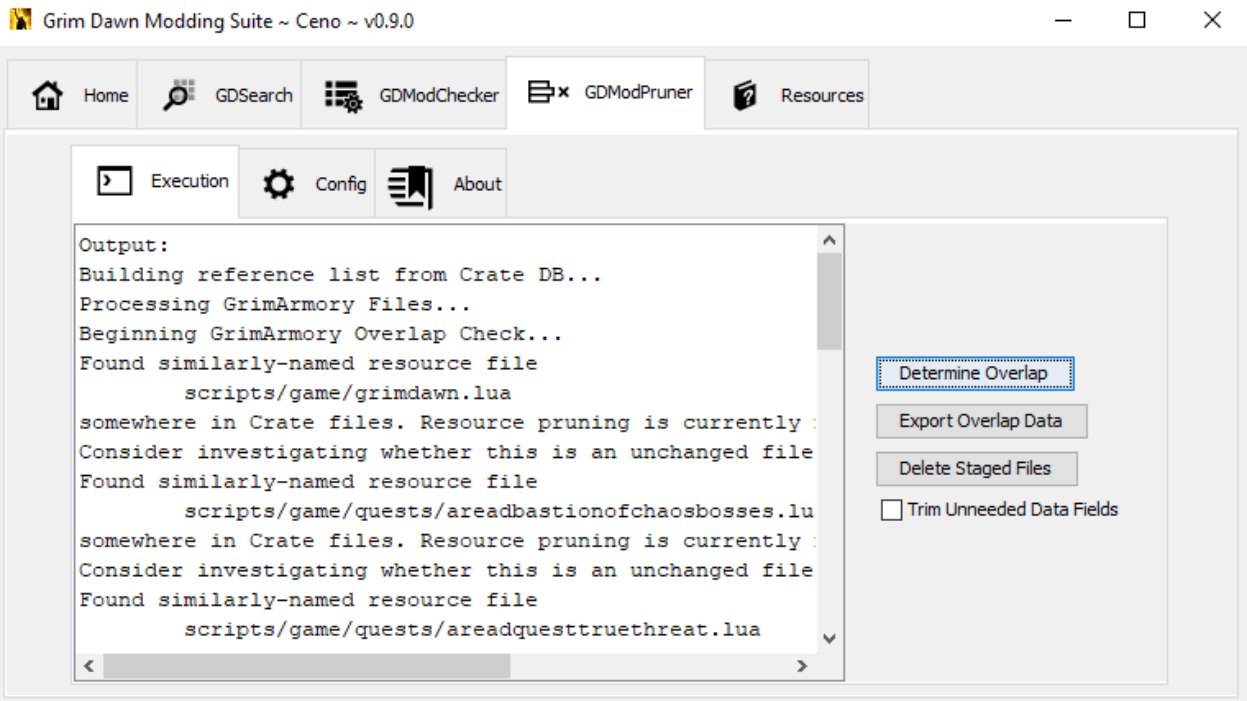
In this way, these two applications communicate a bit of data with each other to streamline their use. Moreover, one may also observe that our local configuration file, gdms.cfg, has been updated with the GrimArmory mod name as well:

```
#Fri Feb 02 11:42:25 EST 2024
gdms.name=Grim Dawn Modding Suite
gdms.version=0.9
gdx1=1
gdx2=1
gdx3=0
install=D\:\\Program Files (x86)\\Steam\\steamapps\\common\\Grim Dawn
mod=GrimArmory
```

(17 – gdms cfg updated.PNG)

This means that if we were to close and relaunch GDMS, this setting would be saved, and we could pick up on working with GrimArmory's files where we left off. With that out of the way and GDModPruner properly configured, let's get on to actually using this application.

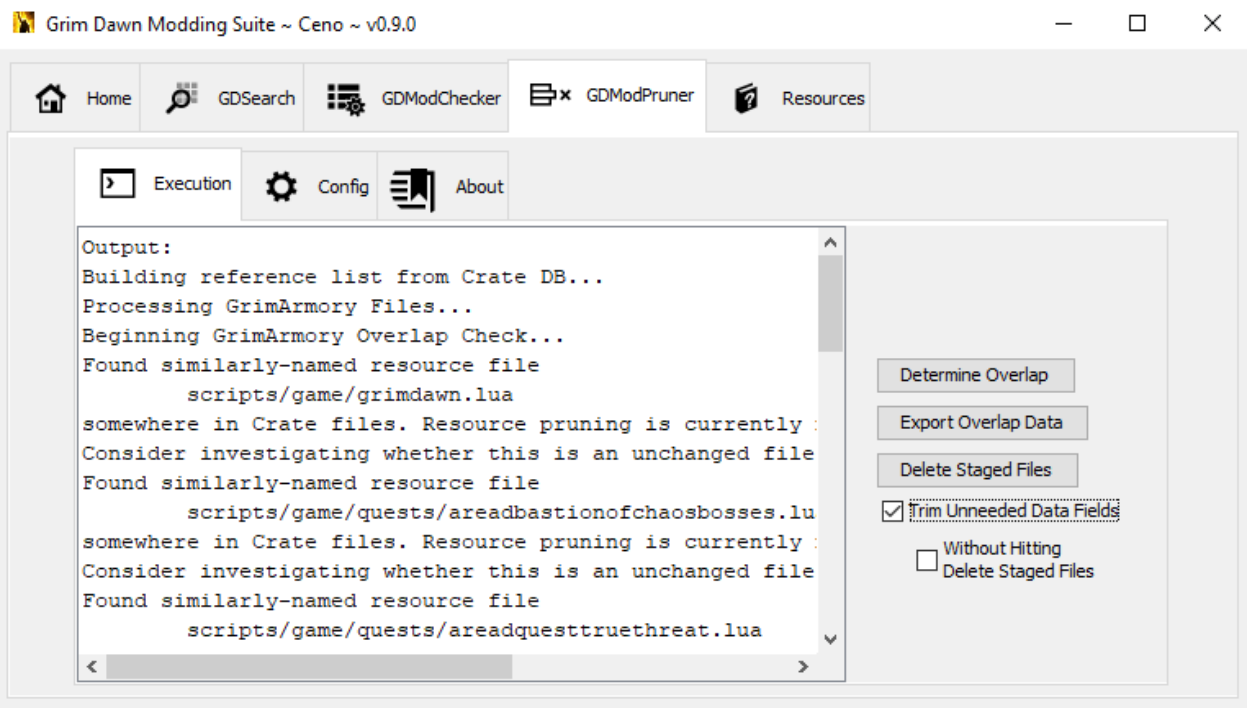
By returning to the Execution menu and clicking the "Determine Overlap" button, GDMS will check to see whether our mod's files are duplicates of any existing files from Crate Entertainment, not merely by filename but also by the contents within. GrimArmory has no such overlap in this example, but we do see some similarly-named resource files being mentioned, which as of v0.9 of GDMS, GDModPruner is unequipped to handle but does identify properly:



(15 – modprune exe example.PNG)

As with GDSearch and GDModChecker before it, we can export the results of our overlap scan to a local file. GDModPruner creates these files in the 'prune/' subdirectory. Had GDModPruner found any files with identical contents to their Crate-counterparts, the application would have staged them for deletion. GDModPruner will not actually delete those files until the "Delete Staged Files" button is pressed, which does bring up a confirmation (Yes/No) dialog window warning users that the process is irreversible. If users press Yes on that window, any files staged for deletion will be removed from the mod. Pressing No does nothing.

By default, GDModPruner will not change or remove any files on the system without user input. This includes trimming extraneous data fields from mod files to save space. To enable this feature, tick the "Trim Unneeded Data Fields" checkbox prior to pressing the "Determine Overlap" button; when staged files are later deleted (with a Yes confirmation), trimmed files will replace the existing mod files. By ticking the checkbox, a user will also reveal a secondary checkbox, which bypasses the need to press the "Delete Staged Files" button to trim extraneous data fields from the mod:



(16 – modprune exe hidden.PNG)

If this second checkbox is also ticked, extraneous data fields will be trimmed from mod files during the process of the overlap check, after hitting the “Determine Overlap” button. Again, caution is advised when using this application, as files in your mod can and will be changed!