**Welcome to Dragon Force Edit!**

Most recent version: **2.4**

**Main Program**: <http://www.mediafire.com/download/vu49s8mkutcc8yx/DFedit2.4.7z>

**Sample Files**: <http://www.mediafire.com/download/ueu8b4nixlgyrnl/DFedit_Sample_Files.7z>

**Email**: DragonForceEdit@gmail.com

Dragon Force Edit (DFedit) is a utility to edit the save files for the Game Dragon Force. Specifically, these save files are created by the Sega Saturn emulator known as SSF. The Hook Backup Library option needs to be checked under Options -> Program4 of SSF in order for it to generate the proper save files.

*Please remember that DFedit is a learning exercise for me. I have only done GUI programming twice before, both of them were very small school assignments (a Keno game in Java and a small, simple C++ program using FLTK). So this was an entirely new experience for me and I have done countless hours of Petzold book reading, Googling, MSDN Win32 information lookup, and more. As a result, it is obviously not as clean and polished as it could be. But I continually refactor this code as I learn more and get more comfortable and experienced with Win32 and GUI programming in general.*

DFedit lets you do all kinds of things! Some of the things you can do are:

* Edit almost anything relating to Generals. There are about 45 different things you can change.
* Edit things relating to castles such as who owns the castle, who is the leader, which generals are stationed there, which captives are held there, how many troops the castle currently has, and what level the castle is.
* Editing things relating to divisions such as who owns the division, who the division leader is, which generals are members of that division, who the division captives are, where the division is located, where the division’s goal is, and the division status (charging, running away, victory pose, etc).
* Mass Editing! Meaning, you can edit Generals or Castles in large groups! For Generals you can edit all Generals of a specific class or manually choose as many Generals as you want. For castles you can pick which specific castles to edit, or you can choose all of the castles owned by a specific monarch.
* You can edit your item inventory. This is the item inventory that you see during Domestic Affairs.
* Kingdom Status gives you a basic overview of each kingdom.

DFedit is hand written in pure Win32 using the Code::Blocks IDE with the TDM-GCC version of the MinGW compiler suite. The program separates the file editing code from the GUI creation and management code. There should not be any Win32 code of any kind in any of the file editing classes. These classes/files are:

* DragonForce.cpp
* General.cpp
* Castle.cpp
* Division.cpp
* FileReader.cpp
* FileWriter.cpp

The GUI classes however, do interact directly with the file editing code (the classes listed above) through the public portion of the classes. Most of the content in the file editing classes are public, but I am in the process of refactoring the above classes so that the data will be private and thus accessed through setters/getters. The classes/files that run the GUI are:

* DFGui.cpp
* FileMenu.cpp
* resources.h
* resource.rc
* MainTabParent.cpp
* GenTabWndCtrls.cpp
* CasTabWndCtrls.cpp
* DivTabWndCtrls.cpp
* MassEditTabWndCtrls.cpp
* InvTabWndCtrls.cpp
* KingdomsTabWndCtrls.cpp

The names of the files should be pretty well self documenting, but I’ll give short descriptions of them anyway.

* FileReader and FileWriter interact with the save file directly. Functions performed are as you would expect, given the names of the files.
* DragonForce is the main class that glues all of the other file editing classes and code together.
* General is the class that contains all of the data relevant to the generals in the game. Stats, map location, what kingdom they are a part of, etc.
* Castle is the class that contains all of the data relevant to the castles in the game. Castle level, troops count, which generals are stationed there, who owns the castle, etc.
* Division is the class that contains all of the data relevant to the armies that march around on the map. Who owns the division, their destination, goal, next path point, previous path point, generals, captives, X/Y coordinates, etc.

GUI classes:

* DFGui creates and manages the main window itself. WinMain, WinMain’s WndProc, initialization of common controls, etc.
* FileMenu creates and runs the File portion of the menu bar at the top of the program.
* resource.h has all of the ID #defines for the GUI elements.
* resource.rc has typical resource file stuff: menu bar, program icon, About dialog, etc.
* MainTabParent is the main tab control. This is where new tabs are added to the GUI.
* GenTabWndCtrls creates and runs the Generals tab. Has its own WndProc.
* CasTabWndCtrls creates and runs the Castles tab. Has its own WndProc.
* DivTabWndCtrls creates and runs the Divisions tab. Has its own WndProc.
* MassEditTabWndCtrls creates and runs the Mass Edit tab. Has its own WndProc.
* InvTabWndCtrls creates and runs the Inventory tab. Has its own WndProc.
* KingdomsTabWndCtrls creates and runs the Kingdoms Status tab. Has its own WndProc.

**Flow of Execution**

Forthcoming. Here is where I will explain as much of the step by step flow of the program as I can.