GB ENHANCED+ USER MANUAL

Version 1.0
D.S. Baxter

0. Table of Contents GB Enhanced+ Manual - 2

1. Foreword

This manual will attempt to cover all the functions users may encounter while operating GB Enhanced+. The information contained herein doubles as both an operating guide and reference to the emulator's various features.

GB Enhanced+ is the successor to the original GB Enhanced project (note the shiny "+"). It is a Game Boy, Game Boy Color, and Game Boy Advance emulator that aims to provide as many enhancements as reasonably possible. Although there is much work to be done in later releases, such as cheat code support, emulation of the GB Camera and Printer, and implementing pixel shaders, GBE+ is dedicated to pursuing these types of things.

Perhaps most importantly, GBE+ supports an emerging emulation technique for 2D games, what the project refers to as **Custom Graphics**, or **CGFX** for short. Like HD textures for N64 games, CGFX lets users replace in-game graphics with their own pixels, from simple recolors to full-blown high-definition versions. While replacing graphics for 2D systems is not common among other emulators, GBE+ is proud to push the boundaries in this field of emulation.

What started as an idle dream to make my own NDS emulator has become something much more than the humble project I founded in 2012. GBE+ is still in early development, despite its 1.0 release. Many, many rough edges remain. There is still a lot more to go through before I can see my vision completed, however, I believe this project is on the right path. In the coming years, we'll see where the road takes this emulator. In the meantime, I hope this manual will give users an insight to how the programs works, what it's capable of, and where it's going.

D.S. Baxter - aka Shonumi

2. Getting Started

Getting started with GBE+ is relatively simple. The emulator does not have many requirements to build from source, and installation should be simple for most operating systems. Please consider, however, that GBE+ has not been tested on OS X in any way, shape, or form. This is due to the lack of access to the operating system. Users can still build it themselves on OS X.

Currently, GBE+ has minimal hardware requirements. Any recent computer should be able to run the emulator just fine. Certain tasks, such as processing large amounts of CGFX, require more single-threaded processing power. Under some circumstances CGFX may also benefit from more RAM. Generally, however, these scenarios are reserved for intense use of HD graphics. Otherwise GBE+ is not a demanding emulator.

It should be noted, however, that as of 1.0, GBA games in general eat up a lot of CPU resources. This is due to inefficiencies in the GBA core that will be addressed in the very near future.

GBE+ aims to build and run with minimal software dependencies. The recommended minimum version of OpenGL is 2.1. Any computer released in the past decade should support this without any trouble. Future versions of GBE+ will move to OpenGL 3.3 (for backward compatibility) and Vulkan.

Currently, GBE+ supports both 32-bit and 64-bit systems. For the foreseeable future, this will remain the case. For CPU emulation, any dynamic recompilers added to later releases, will only target x64 systems. Nevertheless, GBE+ will technically continue support 32-bit systems at that time through CPU interpreters.

For a general roadmap of where GBE+ will go from here, please see the FAQ in **Section 6** for more details.

2.1 Obtaining, Compiling, & Installing GBE+

For Windows users who do not want to build from source code, please visit the project's GitHub page and check out the 1.0 release on the Release page. Download the zip file and extract it. Simply double-click the executable file *gbe_plus_qt.exe* to run the GUI version of the emulator. For those interested in the command-line version of GBE+, run the *gbe_plus.exe* file from the command prompt. The majority of this manual focuses on the Qt version of GBE+, however, please refer to **Section 2.2** for more details about running the command-line version.

Linux users have to compile the source code themselves. Compiling from source requires prior installation of the following programs and libraries:

- GIT
- CMake
- SDL 1.2
- OpenGL
- Qt4 or Qt5 (optional)

CMake will check to make sure it can find all of the necessary dependencies before the build process begins. Before that happens, however, GIT must retrieve the source, or the source tarball from the Release page must be downloaded and extracted. The following terminal instructions detail how to download the source code through GIT, compile it, and install the emulator:

```
git clone https://github.com/shonumi/gbe-plus.git
cd gbe-plus
mkdir build && cd build
cmake ..
make && make install
```

Note that this will install the very latest source code. GIT can check out specific revisions based on the hash of that commit. Consult the GIT documentation for checking out revisions and the GBE+ GitHub repository for the appropriate hash. Once CMake installs GBE+, the emulator can be called via *gbe_plus* for the command-line version, or *gbe_plus_qt* for the Qt version.

2.2 Command-Line Options

The command-line version of GBE+ accepts several parameters. Below are all the valid arguments for the emulator along with a short description of what they do:

-b or **--bios [FILE]:**
This instructs GBE+ to boot a system's BIOS or Boot ROM with the provided file when loading a game. The second argument is the exact path for the BIOS or Boot ROM on the user's computer.

-d or **--debug:**
This starts GBE+ in debug mode. It will pull up the command-line debugger. See **Section 5.7** for details on how to use properly use this version of the debugger.

--opengl:
This forces GBE+ to use OpenGL for all drawing/blitting operations instead of SDL.

--2x, --3x, --4x, --5x, --6x:
Scales the screen by a given factor. Only applicable when OpenGL is enabled.

--sys-auto:
Sets the emulated system type to AUTO. In this mode, GBE+ will automatically determine what system to emulate based on the game.

--sys-dmg:
Sets the emulated system type to DMG (old black and white Gameboy). This option is not valid when running GBA games and is ignored by the emulator.

--sys-gbc:
Sets the emulated system type to GBC. This option is not valid when running GBA games and is ignored by the emulator. DMG games will run as if on a GBC.

--sys-gba:
Sets the emulated system type to GBA. DMG/GBC games will run as if on a GBC, however, like a real GBA, the screen can be stretched horizontally by pressing the L and R triggers.

--multicart:
Forcibly emulates MBC1 games as MBC1M variants. Use this option for games like Mortal Kombat I & II, or Bomberman Collection.

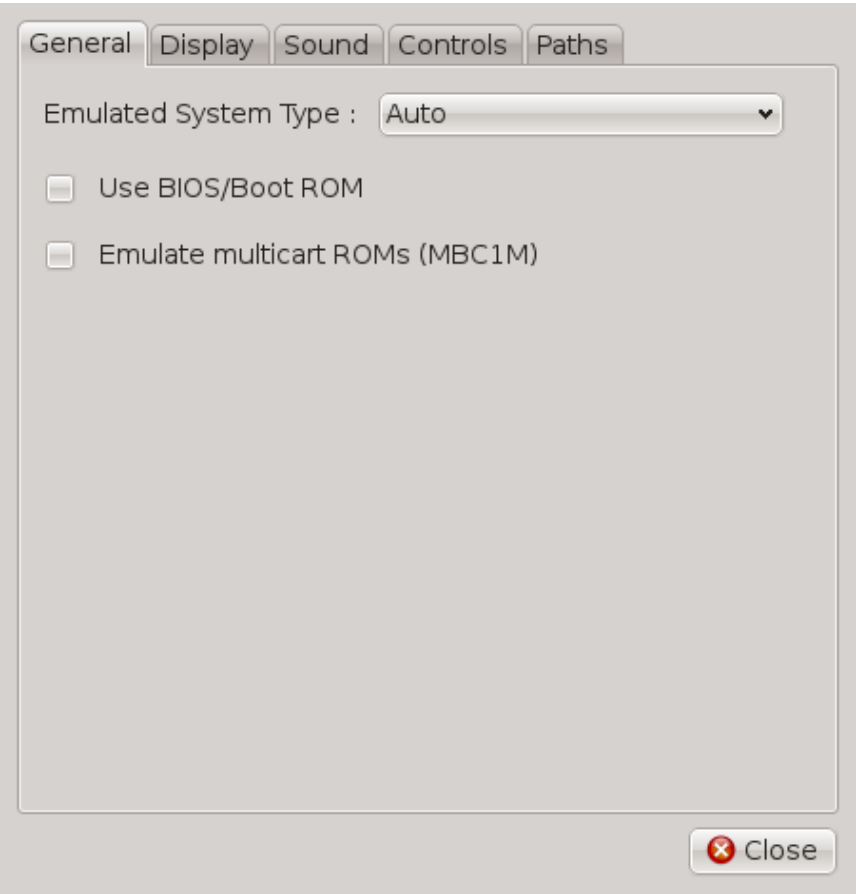
--h, or **--help:**
Displays a brief help message explaining all of the above options.

3. Configuration

GBE+ has many options that will affect how the program runs games. There are 5 major areas of configuration: General Settings, Display Settings, Sound Settings, Control Settings, and Paths. The following sections detail what these options do for the Qt version of the emulator.

3.1 General Settings

This section deals with miscellaneous settings that don't belong to any particular category.



Emulated System Type:

Forces GBE+ to emulate a certain system. The following options are:

Auto - In this mode, GBE+ will automatically determine what system to emulate based on the game.

Game Boy [DMG] - Emulates the DMG (old black and white Gameboy). This option is not valid when running GBA games and is ignored by the emulator.

Game Boy Color [GBC] - Emulates the GBC. This option is not valid when running GBA games and is ignored by the emulator. DMG games will run as if on a GBC.

Game Boy Advance [GBA] - Emulates the GBA. DMG/GBC games will run as if on a GBC, however, like a real GBA, the screen can be stretched horizontally by pressing the L and R triggers.

Use BIOS/Boot ROM:

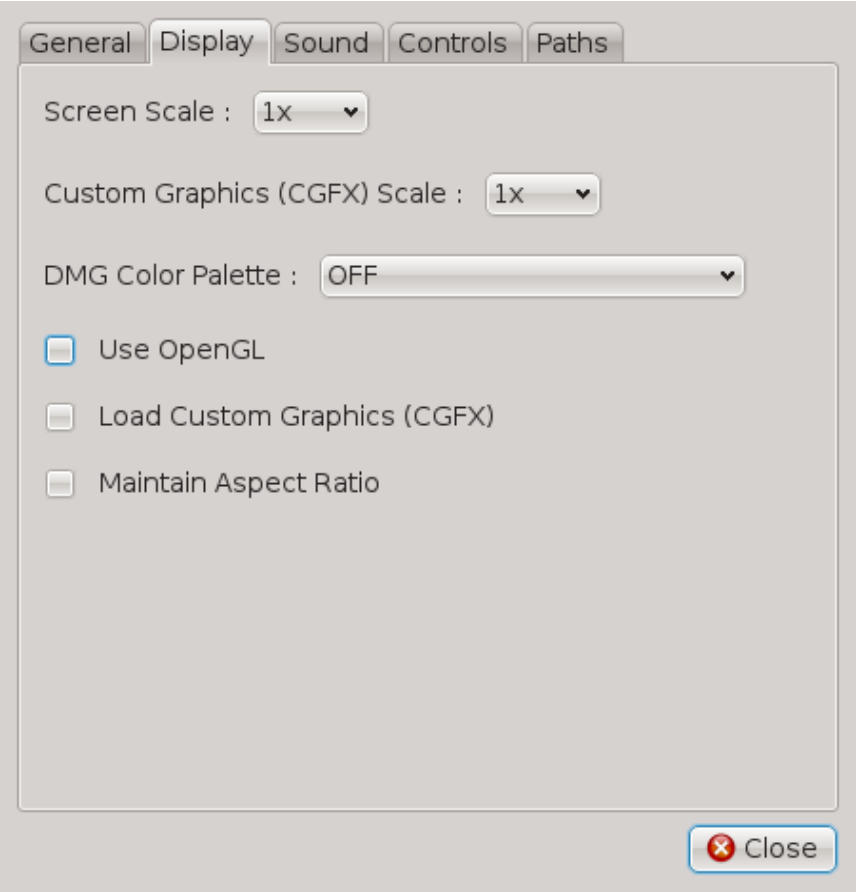
This instructs GBE+ to boot a system's BIOS or Boot ROM. When checking this option, users must ensure that the proper paths to the DMG, GBC, or GBA files are configured. See **Section 3.5** for more details on configuring BIOS and Boot ROM files.

Emulate multicart ROMS (MBC1M):

Forcibly emulates MBC1 games as MBC1M variants. Use this option for games like Mortal Kombat I & II, or Bomberman Collection. Please note, however, that only a rare handful of games need this option. It is not necessary for most games and can cause issues if left checked for the wrong ones. If unsure, leave unchecked.

3.2 Display Settings

Display settings deal with how GBE+ draws things on screen.



Screen Scale:

Determines what factor to scale the original game screen. Factors of 1x to 10x are applicable. Unlike the command-line version, the Qt version can scale the image regardless of whether OpenGL is used or not.

Custom Graphics (CGFX) Scale:

Determines what the input scale for CGFX is. *When loading CGFX into the emulator, this option must be set to the correct scale.* For example, if a user makes HD graphics that are 4x the size of the original graphics, this option must be set to 4x. Mismatching the scale will result in graphical errors when using CGFX. This scale is also multiplied by the Screen Scale to get the final screen size.

DMG Color Palette:

DMG games can be colorized with special palettes. **OFF** emulates standard grayscale colors. **DMG - Classic Green** emulates the old-school green LCD. The other options emulate palettes from the GBC Boot ROM.

Use OpenGL:

Use OpenGL for all drawing/blitting operations. This option is faster than using software, especially when increasing the Screen Scale.

Load Custom Graphics (CGFX):

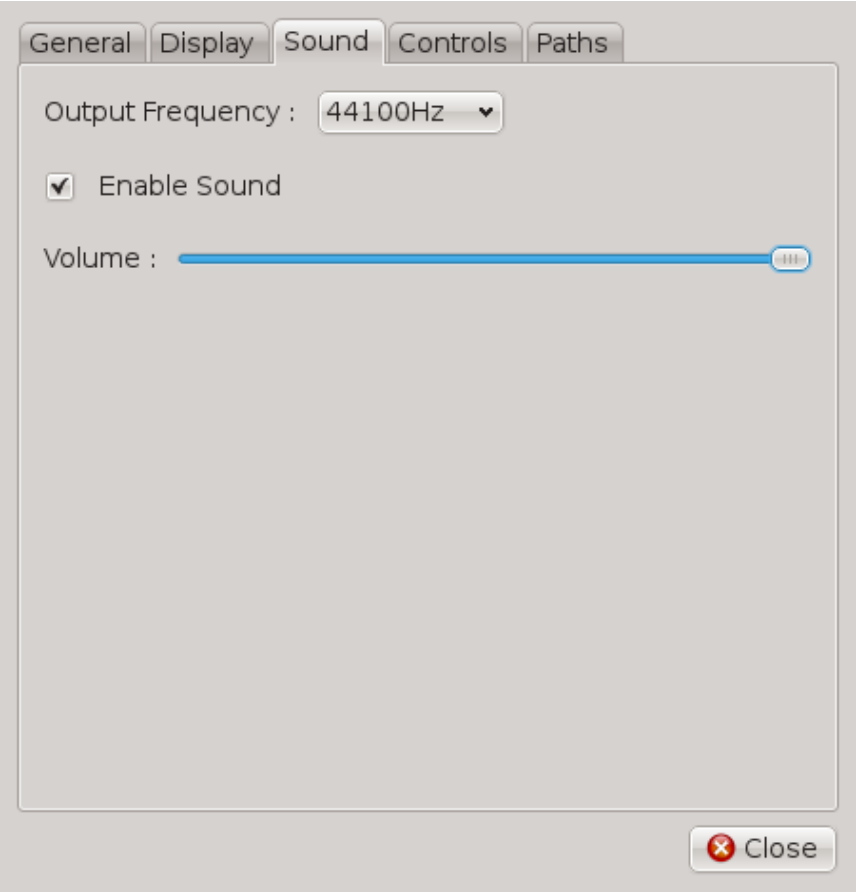
Loads Custom Graphics into the emulator. This option must be enabled in order to play using CGFX. Additionally, a manifest file must be selected in Paths. For full details on how to set up CGFX, refer to **Section 4.4**.

Maintain Aspect Ratio:

Forces GBE+ to maintain the same aspect ratio as the original system regardless of the window size. Leaving this option unchecked will let GBE+ fill in the available window space.

3.3 Sound Settings

Sound settings determine how GBE+ will process audio.



Output Frequency:

Determines the final output frequency of all sound. The available options are **11025Hz**, **22050Hz**, **44100Hz**, and **48000Hz**. With higher frequencies, GBE+ will produce better audio quality. Please note, changing this option only takes effect when booting or resetting a game. If unsure, please leave this option at its default setting.

Enable Sound:

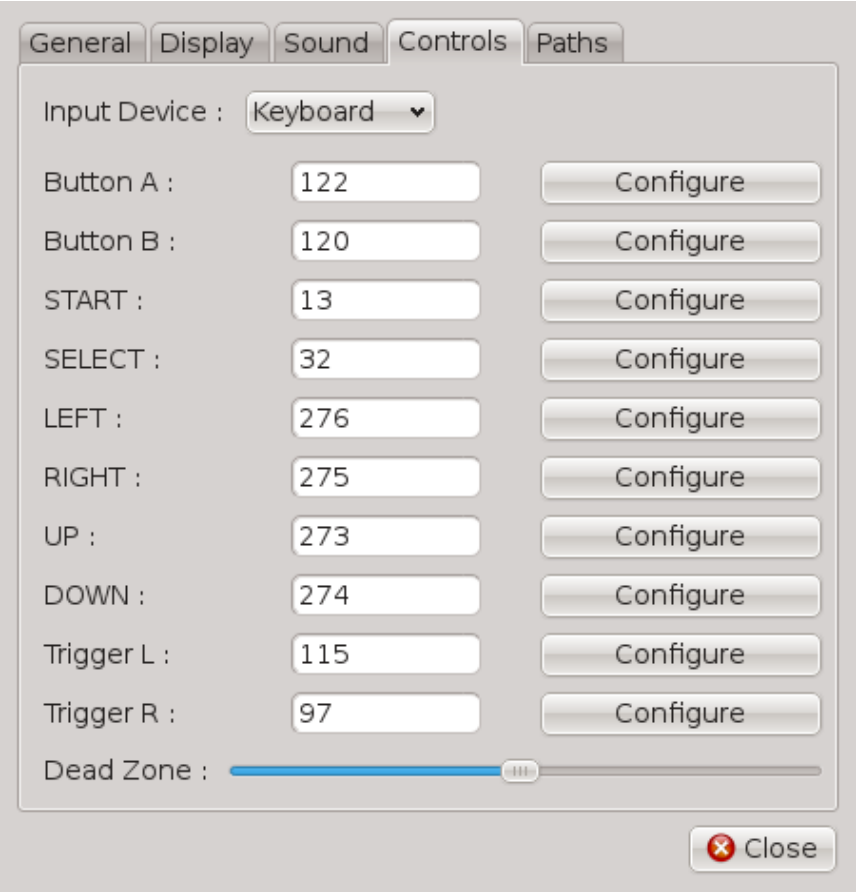
Checking this option enables sound output. Unchecking this option will mute any sounds from GBE+

Volume:

This slider controls the master volume for all sound output from GBE+. Turning it all the way down to zero will effectively mute the emulator.

3.4 Controls Settings

Controls settings allow users to configure input for the emulator via the keyboard or joysticks



Input Device:

Selects the input device to configure. Both keyboards and joysticks can be configured. This option will contain a list of all available joysticks GBE+ can detect. Please note that both keyboard and joysticks can be used at the same time, regardless of what this option is set to.

Buttons:

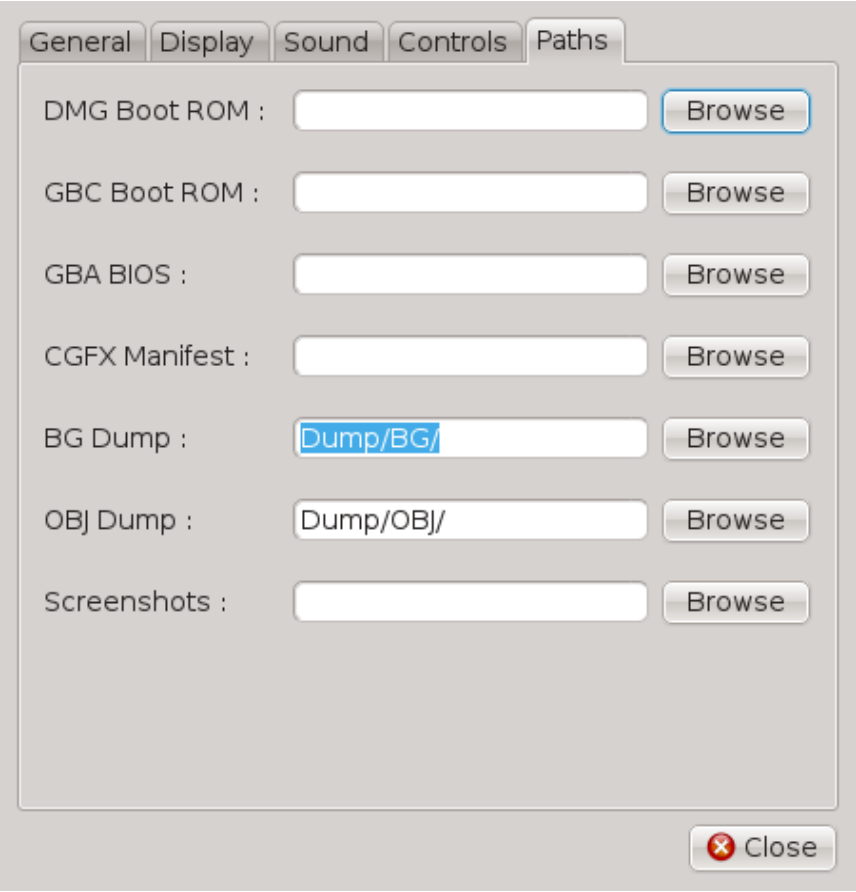
These are the individual buttons that can be configured. Press the **Configure** button for GBE+ to map input from a device. For joysticks, there is a 3 second delay before configuration is processed. During this time, please continue holding down on the joystick for GBE+ to finish mapping.

Dead Zone:

Configures the dead zone for joystick axes. A smaller dead zone means less tilt on an axis is needed to trigger an input. A larger dead zone means more tilt on an axis is needed to trigger an input. If unsure, please leave this option at its default setting.

3.5 Paths Settings

Paths settings determine the location GBE+ will look to for important files such a screenshot directories, or the location of BIOS or Boot ROMs. To set a location, click the **Browse** button and choose a folder or file from the pop-up.



DMG Boot ROM:

This is the original Game Boy's Boot ROM file. Although booting this will have no effect on overall emulation, this is will let users see the scrolling "Nintendo" logo.

GBC Boot ROM:

This is the Game Boy Color Boot ROM file. Booting up this will let users choose color palettes when running DMG games, and some DMG games (like *Metroid II: Return of Samus*) will have custom palettes.

GBA BIOS:

This is the Game Boy Advance BIOS file. Booting this up will let users see the GBA's "Nintendo" logo and the boot animation. Currently, all BIOS functions are high-level emulated, so GBE+ does not actually run the code from the GBA BIOS. In the future, low-level emulation of the BIOS will be possible.

CGFX Manifest:

This is the manifest file GBE+ will use when loading Custom Graphics or when dumping them via the Advanced Menu. See **Section 4** for more details about Custom Graphics.

BG Dump:

This is the folder GBE+ will look to when loading or dumping Background Tiles for Custom Graphics. By default, this is located in the **data** folder for GBE+. This folder *must* be located within the **data** folder for GBE+ to correctly find the image files.

OBJ Dump:

This is the folder GBE+ will look to when loading or dumping Sprite Tiles for Custom Graphics. By default, this is located in the **data** folder for GBE+. This folder *must* be located within the **data** folder for GBE+ to correctly find the image files.

Screenshots:

This is the folder GBE+ will use to store screenshots.

3.6 The .ini File

GBE+ can be configured through a .ini text file. This file includes nearly all of the same options as the Qt options previously described. For both the SDL and Qt versions of GBE+, the .ini file will be loaded and automatically set up any options. The .ini file and its format are self-documented, so please refer to the default .ini file to see how it works and how to edit it.

GBE+ will always search for a .ini file in the same folder as the emulator itself. If no such file exists, GBE+ will search in the **data** folder for its .ini file. If no .ini file exists in the **data** folder either, GBE+ will its default settings.

For the Qt version, GBE+ will always update and save the .ini file. This way, changes to settings are remembered during the next play session.

4. Custom Graphics (CGFX)

Custom Graphics (referred to as CGFX) are an exciting new way of modding in-game graphics, much like custom textures used on other platforms. This section details how GBE+ can extract and replace tiles for backgrounds and sprites.

4.1 Overview

CGFX works by first grabbing tile data from a game, which GBE+ offers numerous tools for achieving. Once the graphics have been extracted or “dumped”, they are edited to look like something else. The changes can be simple 1:1 recolors of black-and-white-only games such as the screenshot above, or they can be bigger, high-definition versions. Once those changes are made, GBE+ can automagically load the graphics and draw them on-screen when running the game.

4.2 Dumping Graphics

In order to replace a game's original graphics with CGFX, the original graphics must first be extracted from the game. This process is known as "dumping" or "ripping". GBE+ provides a dedicated user interface for this very purpose. **At this time, it is important to note that CGFX only work for GB and GBC games.** Support for GBA games will come in later releases. The guide is applicable to the Qt version only.

Before beginning, please make sure the paths for the BG and OBJ dump folders are correct. Go to **Options -> Paths** to check or change any settings. **The BG and OBJ dump folders must always be in GBE+'s data folder**. By default, these two folders are data/Dump/BG and data/Dump/OBJ, but they can be changed, moved or renamed (so long as they remain in the data folder).

When dumping custom graphics, a manifest file should be specified. This is simply a text file that will tell GBE+ where to look for custom graphics and how it should load and handle them. Go to **Options -> Paths** and make sure the CGFX Manifest points to a valid text file. When starting new CGFX projects, the manifest may be left as an empty text file.

To get started, load a game from the menu. Once the game starts, click on **Advanced -> Custom Graphics...** The screen below will appear:

Use advanced menu:

Every time graphics are dumped, GBE+ will let users specify additional options through an advanced menu. See **Section 4.4** for an extended explanation of the advanced menu and its features.

Automatically dump OBJ tiles:

GBE+ will automatically dump sprites as users play the game. This option is not recommended for beginners, as a lot of data can be generated. It can be hard to keep track of what graphics GBE+ is extracting, making editing harder as well. Use this only if you know what you're doing.

Automatically dump BG tiles:

GBE+ will automatically dump background tiles as users play the game. This option is not recommended for beginners, as a lot of data can be generated. It can be hard to keep track of what graphics GBE+ is extracting, making editing harder as well. Use this only if you know what you're doing.

Ignore blank/empty tiles when dumping:

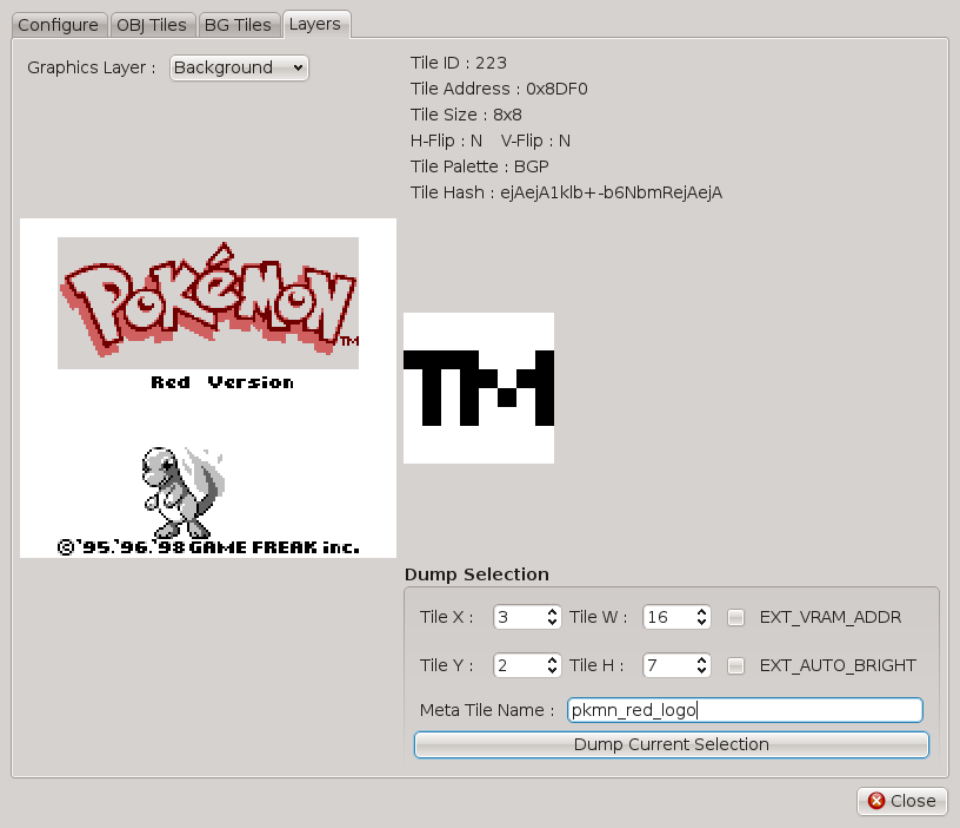
Useful for the above automatic dumping options. Sometimes games generate graphics that are blank tiles (solid colors), that GBE+ may mistake for graphics that might be relevant. This option will prevent GBE+ from dumping them at all.

The automatic options only require users to check the boxes, close the Custom Graphics dialog, and run the game. This is generally not the recommended way to dump graphics because a lot of graphics can be dumped at once. This makes editing things quite hard in some cases. A better approach is to dump only a few graphics at a time manually. There are tabs specifically for dumping BG and OBJ tiles. Clicking on one of the tiles will dump it. For even greater control over dumping graphics, the Layers tab provides a better interface.

This is the recommended method of dumping graphics due to its ease of use. Here, users can switch between the Game Boy's Background, OBJ, and Window layers. The present screen will be displayed. Use the mouse to hover over a specific tile. A magnified version will appear on the right along with detailed data about the tile itself. Clicking here will dump the tile.

4.3 Dumping Meta-tiles

Meta-tiles offer users a way to dump a section of the screen all at once, then save the results as a single image. Take, for example, the POKEMON logo from Pokemon Red. The logo itself is just short of 100 8x8 background tiles. Normally, it would take many people hours to dump each tile and edit them by hand. As a convenience, GBE+ will let users save that logo as a single bitmap file which can be edited and loaded later.



To create a meta-tile, simply go to the Layers tab and go to **Dump Selection**. In this area are parameters to control which section will be dumped. The X and Y boxes determine where the on-screen selection will begin. The W and H boxes determine the dimensions of the selection. The selection will automatically highlight which area is going to be dumped. Additionally, **EXT_VRAM_ADDR** and **EXT_AUTO_BRIGHT** options are available (see **Section 4.4** for more details). Once the selection has been made, enter in a name for the file. **This file will automatically appear in data/Dump/BG and will automatically be saved as a .BMP file.** All the necessary manifest entries are written when saving as well.

At this time, dumping meta-tiles is only supported for Background tiles, not OBJs. OBJs can still take advantage of meta-tiles if dumped manually and if the manifest file is edited by hand. See **Section 4.5** for more info about the manifest file.