GBR file specifications

Date: 1 june 1999
By: Harry Mulder (hpmulder@casema.net)

This document specifies the structure and content of a GBR file.

In this document, the following datatype definitions are used:

Word	16-bit hi-endian.
Long	32-bit hi-endian.
String (xx)	C-style string; ie. ends with hex 00, with a maximum length of xx (including end-marker).

Header

Each GBR file should begin with the following header:

Name	Size	Content
Magic marker	3 bytes	“GBO”
Filetype version	1 byte	“0”

Use the *Magic marker* to make sure this is indeed a GBR file; it should always contain **GBO** (in uppercase). The *Filetype version* tells which structure is used in this GBR file. For the structure specified in this document, this number should be **0** (in ASCII, not binary). This number will only change if the complete structure changes, *not* when an object-type is added or changed.

Objects

Directly after the header, zero or more objects will follow, containing the actual information. Each object has the following prefix:

Name	Size	Content
ObjectType	Word	The type of the following object
ObjectID	Word	Unique identifier for this object
RecordLength	Long	Size of this object (in bytes, excluding this header)

This header permits you to ‘walk’ through all objects in the GBR file, using the ones you want and ignoring the ones you don’t need (or know), in the following fashion:

Read the first 8 bytes. If the ObjectType is the one you want, handle it. Else, skip the next ObjectLength bytes. Repeat until EOF().

The EOF means a real filesystem-response; there is no EOF-marker in the file itself.

When writing objects, you should keep the following in mind:

  - When “saving” an existing file, it is best to update it instead of rewriting the whole file; if you do rewrite it, make sure you copy any objects (and extra data in known objects) you don’t know to this new file to retain information stored by other applications (or newer versions of your own).

- Don’t assume any length for an object; *always* use the size specified in the file; so, when writing an object to a GBR file which contains the same object, but with a longer size, don’t decrease it; just change the bytes you want to change, and ignore the rest. When you encounter a shorter object, you can either reshape the whole file to account for the extra bytes (not recommended), or either mark the current object as deleted and add a new one at the end of the file.

As long as you ignore (ie don’t change or delete) unknown or unwanted objects and data, use the RecordLength to determine the size of an object, don’t assume any order in which the objects should appear in the GBR-file and don’t assume all the objects you need are available in the file, then this flexible system gives you up- and downward compatibility with all GBR files of Filetype **0** (with is currently the only one available).

Objecttypes

The following objecttypes are currently in use:

Producer (Hex 0001)

Name	Size	Content
Name	String (30)	Name of application.
Version	String (10)	Version of application.
Info	String (80)	Extra information about the application.

This object contains some information about the application that made the GBR file.

TileData (Hex 0002)

Name	Size	Content
Name	String (30)	Name of tileset.
Width	Word	Width of each individual tile.
Height	Word	Height of each individual tile.
Count	Word	Total number of tiles in this tileset.
ColorSet	4 bytes	Colorset which applies to this tileset.
Data	x bytes	Actual tileset data.

This object contains a tileset.

The layout of the Data-array is depending on the Width and Height; if the tilesize is 8x8, the first 8*8 bytes hold the first tile, the next 8*8 the next, etc. if the size is 32x32, the first 32*32 bytes contain the first tile, etc.

Each byte in Data is $00-$03; remap these through the ColorSet to get the real color (the ColorSet is like the Object-Palette in the Gameboy itself).

The Name-field can be used as a label to the tileset (for user-friendliness; internal reference to objects should be done via ObjectID’s).

TileSettings (Hex 0003)

Name	**Size**	**Content**
TileID	Word	ObjectID of object to which this object refers (usually a Tiledata object).
Simple	Boolean	Extra resulting tiles are shown.
Flags	Byte	Display settings: Hex 01 Grid is shown. Hex 02 Nibble markers are shown.
LeftColor	Byte	Currently selected color for left mouse-button.
RightColor	Byte	Currently selected color for right mouse-button.
GBTD 0.9		
SplitWidth	Word	Number for horizontal tiles for Split-Paste&Copy.
SplitHeight	Word	Number for vertical tiles for Split-Paste&Copy.
SplitOrder	Long	Order used by Split-Paste&Copy: 1 Left to right, top to bottom 2 Top to bottom, left to right
1. GBTD 1.0		
ColorSet	Byte	Selected color set: 1 Pocket 2 GameBoy 3 GBC 4 SGB
1. GBTD 1.1		
Bookmarks	Word(3)	Tiles which are bookmarked.
GBTD 2.0		
AutoUpdate	Boolean	AutoUpdate is/isn’t active.

This object contains various screen-settings; although these settings are targeted for GBTD, other applications will most likely be able to use some of these settings for comparable functions.

TileExport (Hex 0004)

Name	**Size**	**Content**
TileID	Word	ObjectID of object to which this object refers (usually a Tiledata object).
FileName	String (128)	Filename of export file.
FileType	Byte	Type of export file: Hex 00 RGBDS Assembly file Hex 01 RGBDS Object file Hex 02 TASM Assembly file Hex 03 GBDK C file Hex 04 Binary file
SectionName	String (20)	Name of section to use.
LabelName	String (20)	Label to use for data.
Bank	Byte	Bank in which data should be stored.
TileArray	Boolean	Should the tiles be stored as one, or each separate.
Format	Byte	Format of exported data: Hex 00 Gameboy 4 color

		Hex 01 Gameboy 2 color Hex 03 Byte per color
Counter	Byte	Type of counter to include: Hex 00 None Hex 01 Byte-count as Byte Hex 02 Byte-count as Word Hex 03 Byte-count as Constant Hex 04 Tile-count as Byte Hex 05 Tile-count as Word Hex 06 Tile-count as Constant Hex 07 8x8-count as Byte Hex 08 8x8-count as Word Hex 09 8x8-count as Constant
From	Word	Start exporting from this tile.
Upto	Word	Export upto this tile.
Compression	Byte	The following compression-type should be used: Hex 00 None Hex 01 GB-Compress
		GBTD 1.3
IncludeColors	Boolean	Include palette colors Yes/No
		GBTD 1.4
SGBPalettes	Byte	Include SGB palettes in following fashion: 1 None 2 Constant per entry 3 2 Bits per entry 4 4 Bits per entry 5 1 Byte per Entry
6 GBCPalettes	Byte	Include GBC palettes in following fashion: 7 None 8 Constant per entry 9 2 Bits per entry 10 4 Bits per entry 11 1 Byte per Entry
		1. GBTD 1.5
MakeMetaTiles	Boolean	Convert to Metatiles Yes/No
MetaOffset	Long	Index offset
MetaCounter	Byte	Index counter type: 1 None 2 Byte-count as Byte 3 Byte-count as Word 4 Byte-count as Constant 5 Tile-count as Byte 6 Tile-count as Word 7 Tile-count as Constant 8 8x8-count as Byte 9 8x8-count as Word 10 8x8-count as Constant
		1. GBTD 1.8
Split	Boolean	Split data Yes/No
BlockSize	Long	Block size
SelTab	Byte	Currently selected tabpage (GUI).

This object stores the export-settings for a Tiledata object.

TileImport (Hex 0005)

Name	**Size**	**Content**
TileID	Word	ObjectID of object to which this object refers (usually a Tiledata object).
FileName	String (128)	Filename of import file.
FileType	Byte	Type of import file: 1 GBE file 2 Binary 8x8 tiles
3 FromTile	Word	First tile from import file.
ToTile	Word	First tile in current application.
TileCount	Word	Total amount of tiles to import.
ColorConversion	Byte	Import colors: 0 by colors 1 by index
GBTD 1.5		
FirstByte	Long	First byte to use
	Byte	Binary file type: 1 Byte per pixel 2 bits per pixel 3 Gameboy VRAM

This object stores import-settings.

Palettes (Hex 000D)

Name	**Size**	**Content**
ID	Word	ObjectID of object to which this object refers (usually a Tiledata object).
Count	Word	Number of GBC Colors.
Colors	TGBColorSets	Actual colors for GBC.
SGBCount	Word	Number of SGB Colors.
SGBColors	TGBColorSets	Actual colors for SGB.

This object stores the actual palette colors for both the GBC and SGB.

The TGBColorSets works as follows:

For Count (or SGBCount)	
	Color 0 Color 1 Color 2 Color 3

Where ‘Color x’ is a 4-byte RGB value, as used by Windows.

TilePal (Hex 000E)

Name	Size	Content
ID	Word	ObjectID of object to which this object refers (usually a Tiledata object).
Count	Word	Number of tile entries in ColorSet (usually tile count).
ColorSet	Long(x)	Palette map for GBC.
SGBCount	Word	Number of tile entries in SGBColorSet (usually tile count).
SGBColorSet	Long(x)	Palette map for SGB.

This object stores the Palette maps for both the GBC and SGB.

The length of ColorSet is determined by Count; the length of SGBColorSet is determined by SGBCount. Both ColorSet and SGBColorSet have the following structure:

ColorSet[n] = Pal number for tile n.

Deleted (Hex 00FF)

If you see an object of this type, it is deleted and should be ignored.

Additions

You are free to add new objects, data-fields and extra options in indexed data-fields as long as these additions do not conflict with current specifications. Also, please report them to me *as soon as possible*; they will then be added to these specifications. This way, others can start using them too, and clashes are kept to a minimum.

Contacting me

If you have any further questions about these specifications, feel free to contact me at *hpmulder@casema.net*. Also, the latest version of these specifications can be found at my web-site, which is located at *http://www.casema.net/~hpmulder*.

History

Date	**Changed**
22 november 1997	First release.
1 june 1999	Updated for various objects.