GBM file format specifications

By Harry Mulder (hpmulder@casema.net)

This document specifies the structure and content of a GBM file, as generated by Gameboy Map Map Builder ("GBMB") version 1.0 and higher. This document does NOT specify the format of versions before 1.0.

In this document, the following datatype definitions are used:

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

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.

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*.

Date	Revision
October 1999	● Extended for GBMB 1.8 (Object 0x0003)
● March 1999	● Extended for GBMB 1.2 ● Specified *Map Tile Data*-format
● Januari 1999	● Initial release

Contents

1 General layout

Each GBM-file starts with a header which can be used to determine if this file is indeed a GBM-file, and of the correct version.

After the header, zero or more *Objects* are stored in sequence, whereby each *Object* contains its own size. This way you can 'walk' through the available *Objects*, using the ones you want and ignoring the unknown.

It is important to keep in mind that the order of the actual *Objects* in a file is completely random.

1.1 Header

Each GBM file should begin with the following header:

Name	**Size**	**Content**
Magic marker	3 bytes	"GBO"
Filetype version	1 byte	"1"

Use the *Magic marker* to make sure this is indeed a GBM file; it should always contain **GBO** (in uppercase). The *Filetype version* tells which structure is used in this GBM file. For the structure specified in this document, this number should be **1** (in ASCII, not binary). This number will only change if the complete structure changes, *not* when an *Object* is added or changed.

1.2 Objects

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

Name	Size	Content
Marker	String (6)	"HPJMTL"
ObjectType	Word	The type of the following object
ObjectID	Word	Unique identifier for this object
MasterID	Word	The master-object of this sub-object
CRC	Unsigned long	CRC of object
ObjectLength	Unsigned long	Size of this object (in bytes, excluding this header)

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

Read the first 22 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.

Marker can be used to determine corruption in a file; note that the current version of GBMB does not check for this. However, to be compatible with upcoming version, be sure to insert the specified characters (in uppercase).

The same applies to *CRC*; this can also be used to determine corruption. But again, the current version of GBMB does not support this. Setting this field to 0x00000000 will tell upcoming versions that CRC is not calculated for this *Object*.

MasterID is used by *Sub-objects*. If the current *Object* is not a *Sub-object*, this field will contain 0x0000.

1.3 Sub-Objects

Some *Objects* are *Sub-Objects*; this means they apply to another *Object* found in the file. These are generally used to store variable-length information, and to keep loosely-bound information close together.

All *Sub-objects* have a *MasterID* which points to the *Object* to which it is bound. This master generally contains information about the actual contents of the *Sub-Object.*

Note that apart from the usage of *MasterID*, *Sub-objects* should be regarded as normal *Objects*. Also, their order in the file is as random as any other *Object*, so a situation where a *Sub-object* is found before its master is perfectly valid.

1.4 Notes

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 GBM 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 *ObjectLength* to determine the size of an object, don't assume any order in which the objects should appear in the GBM-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 GBM files of Filetype **1**.

2 Object types

This chapter contains a list of all Objects used by a GBM-file. If an Object requires a value in the MasterID, the value is specified. If not, fill it with 0x0000.

2.1 Producer (0x 0001)

Name	Size	Content
Name	String (128)	Name of application
Version	String (10)	Version of application
Info	String (128)	Extra information about the application

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

2.2 Map (0x 0002)

Name	Size	Content
Name	String (128)	Name of the map
Width	Integer	Width of the map, in tiles
Height	Integer	Height of the map, in tiles
PropCount	Integer	Number of properties
TileFile	String(256)	Full path to the GBR-file used for this map
TileCount	Integer	Number of tiles in GBR file
PropColorCount	Integer	Number of Property colors

This object contains basic information about a map, and is the master for various *Objects* contains the actual data.

Name is currently not used by GBMB; set the first character to 0x00 for forward compatibility.

TileCount does not actually refer to the number of tiles in the GBR-file, as this can be easily changed without the GBM-file "knowing" about it. It is used to determine the size of tile-related information in the GBM-file, mainly the *default properties*.

2.3 Map Tile data (0x 0003)

Name	Size	Content
Data	Record	The tile(number) used for each location, including extra information.

MasterID: Map

This object contains a matrix with one record for each map-location. The dimensions of the matrix are determined by the *Width* and *Height*-fields of the master.

The layout of a record:

Bits	Content	
0-9	Tile-number (0..767)	
10-14	GBC palette	*(GBMB 1.8 and higher)*
15	*Reserved, set to 0*	
16-18	SGB palette	*(GBMB 1.8 and higher)*
19-21	*Reserved, set to 0*	
22	Tile is flipped horizontally (1), or not (0)	
23	Tile is flipped vertically (1), or not (0)	

Note that a record for this type is stored in the following fashion:

Byte #0								Byte #1								Byte #2							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Both *GBC palette* and *SGB palette* use the following format:

1 Use the default palette

<>0 Use palette#-1

2.4 Map properties (0x 0004)

Name	Size	Content
Data	Record	Basic information about each property

MasterID: Map

This object contains an array with one record for each map property. The dimension of the array is determined by the *PropCount*-field of the master.

The layout of a record:

Name	Size	Content
Type	Integer	Type of property
Size	Integer	Maximum size of the property
Name	String(32)	Name of the property

Note that there is currently only one type, which is 0x00000000.

2.5 Map Property data (0x 0005)

Name	Size	Content
Data	Record	The actual property data for each location

MasterID: Map

This object contains a 3 dimentional array with one record for each map-location, which contains the value selected for each property, for each location. The dimensions of the matrix are determined by the *PropCount*, *Width* and *Height*-fields of the master.

The layout of a record is a Word containing the value.

2.6 Map Default property value (0x 0006)

Name	Size	Content
Data	Record	The default property values for each tile.

MasterID: Map

This object contains a matrix which contains the default value for each property, in combination with each tile in the tile-list. The dimensions of the matrix are determined by the *PropCount* and *TileCount*-fields of the master.

The layout of a record is a Word containing the value.

2.7 Map settings (0x 0007)

Name	Size	Content
FormWidth	Integer	Width of the window on screen
FormHeight	Integer	Height of the window on screen
FormMaximized	Boolean	The window is maximized (or not)
InfoPanel	Boolean	Infopanel is shown (or not)
Grid	Boolean	The grid is shown (or not)
DoubleMarkers	Boolean	The DoubleMarkers are shown (or not)
PropColors	Boolean	Property colors are shown (or not)
Zoom	Word	The currently selected zoom-level.
ColorSet	Word	The currently selected Color set.
Bookmarks	Word(3)	Bookmark-storage
BlockFillPattern	Integer	Last used BlockFill-pattern
BlockFillWidth	Integer	Width of last used Blockfill
BlockFillHeight	Integer	Height of last used Blockfill

MasterID: Map

This object contains information about various GUI-settings of GBMB.

2.8 Map Property colors (0x 0008)

Name	Size	Content
Data	Record	Basic information about each Property color

MasterID: Map

This object contains an array with one record for each Property color. The dimension of the array is determined by the *PropColorCount*-field of the master.

The layout of a record:

Name	Size	Content
Property	Integer	Selected property
Operator	Integer	Selected operator
Value	Integer	Selected value

2.9 Map Export settings (0x 0009)

Name	Size	Content
FileName	String (256)	Full path of export-file
FileType	Byte	Selected export type
SectionName	String (40)	Section
LabelName	String (40)	Label
Bank	Byte	Bank
PlaneCount	Word	Plane count
PlaneOrder	Word	Plane order
MapLayout	Word	Map layout
Split	Boolean	Should data be split yes/no
SplitSize	Integer	Block size for spliting
SplitBank	Boolean	Should blocks start on a new bank
SelTab	Byte	Currently selected tab (on screen)
PropCount	Word	Number of export properties
		GBMB v1.2 and higher
TileOffset	Word	Offset to add to each tilenumber

MasterID: Map

This object contains information about the export-settings.

2.10 Map Export Properties (0x 000A)

Name	Size	Content
Data	Record	Basic information about each export property

MasterID: Map Export settings

This object contains an array with one record for each Export property. The dimension of the array is determined by the *PropCount*-field of the master.

The layout of a record:

Name	Size	Content
Property	Integer	Selected property
Size	Integer	Selected size

2.11 Deleted (0x FFFF)

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