NAME

rgblink — Game Boy linker

SYNOPSIS

rgblink [-t] [-w] [-d] [-m *mapfile*] [-n *symfile*] [-O *overlayfile*] [-o *outfile*]
[-p *pad_value*] [-s *symbol*] [-l *linkerscript*] *file ...*

DESCRIPTION

The **rgblink** program links objects created by *rgbasm*(1) into a single Game Boy ROM file.

By default, ROM0 sections created by the assembler are placed in the 16KiB bank 0, and ROMX sections are placed in any bank except bank 0. If your ROM will only be 32KiB, you can use the `-t` option to override this.

Similarly, WRAM0 sections are placed in the first 4KiB of WRAM bank 0 and WRAMX sections are placed in any bank except bank 0. If your ROM doesn't use banked WRAM you can use option `-w` option to override this.

Also, if your ROM is designed for DMG, you can make sure that you don't use any prohibited section by using the option `-d`, which implies `-w` but also prohibits the use of VRAM bank 1.

The arguments are as follows:

`-m` *mapfile*
: Write a mapfile to the given filename.

`-n` *symfile*
: Write a symbol file to the given filename.

`-O` *overlayfile*
: The ROM image to overlay sections over. When an overlay ROM is provided, all sections must be fixed. This may be used to patch an existing binray.

`-o` *outfile*
: Write ROM image to the given filename.

`-p` *pad_value*
: When padding an image, pad with this value. The default is 0x00.

`-s` *symbol*
: ???

`-w`
: Expand the WRAM0 section size from 4KiB to the full 8KiB assigned to WRAM and prohibit the use of WRAMX sections.

`-d`
: Enable DMG mode. Prohibit the use of sections that doesn't exist on a DMG, such as WRAMX and VRAM bank 1. This option automatically enables `-w`.

`-t`
: Expand the ROM0 section size from 16KiB to the full 32KiB assigned to ROM and prohibit the use of ROMX sections. Useful for ROMs that fit in 32 KiB.

`-l` *linkerscript*
: Specify a linkerscript file that tells the linker how sections must be placed in the ROM. This file has priority over the attributes assigned in the source code, but they have to be consistent. See *rgblink*(5) for more information about its format.

EXAMPLES

All you need for a basic ROM is an object file, which can be made into a ROM image like so:

```
$ rgblink -o bar.gb foo.o
```

The resulting bar.gb will not have correct checksums (unless you put them in the assembly source). You should use *rgbfix*(1) to fix these so that the program will actually run in a Game Boy:

```
$ rgbfix -v bar.gb
```

SEE ALSO

rgbasm(1), *rgblink*(5), *rgbfix*(1), *rgbds*(5), *rgbds*(7)

HISTORY

rgblink was originally written by Carsten Sørensen as part of the ASMotor package, and was later packaged in RGBDS by Justin Lloyd. It is now maintained by a number of contributors at https://github.com/rednex/rgbds.