

**NAME**

**rgbasm** — Game Boy assembler

**SYNOPSIS**

```
rgbasm [ -EhLVvw] [ -b chars] [ -D name[=value]] [ -g chars] [ -i path]
[ -M depend_file] [ -MG] [ -MP] [ -MT target_file] [ -MQ target_file]
[ -o out_file] [ -p pad_value] [ -r recursion_depth] [ -W warning]
file ...
```

**DESCRIPTION**

The **rgbasm** program creates an RGB object file from an assembly source file. The input *file* can be a file path, or **-** denoting **stdin**.

Note that options can be abbreviated as long as the abbreviation is unambiguous: **--verb** is **--verbose**, but **--ver** is invalid because it could also be **--version**. The arguments are as follows:

**-b** *chars*, **--binary-digits** *chars*

Change the two characters used for binary constants. The defaults are 01.

**-D** *name*[=*value*], **--define** *name*[=*value*]

Add a string symbol to the compiled source code. This is equivalent to *name* **EQU** "*value*" in code, or *name* **EQU** "1" if *value* is not specified.

**-E**, **--export-all**

Export all labels, including unreferenced and local labels.

**-g** *chars*, **--gfx-chars** *chars*

Change the four characters used for gfx constants. The defaults are 0123.

**-h**, **--halt-without-nop**

By default, **rgbasm** inserts a **nop** instruction immediately after any **halt** instruction. The **-h** option disables this behavior.

**-i** *path*, **--include** *path*

Add an include path.

**-L**, **--preserve-ld**

Disable the optimization that turns loads of the form **LD** [**\$FF00+n8**],**A** into the opcode **LDH** [**\$FF00+n8**],**A** in order to have full control of the result in the final ROM.

**-M** *depend\_file*, **--dependfile** *depend\_file*

Print make(1) dependencies to *depend\_file*.

**-MG** To be used in conjunction with **-M**. This makes **rgbasm** assume that missing files are auto-generated: when **INCLUDE** or **INCBIN** is attempted on a non-existent file, it is added as a dependency, then **rgbasm** exits normally instead of erroring out. This feature is used in automatic updating of makefiles.

**-MP** When enabled, this causes a phony target to be added for each dependency other than the main file. This prevents make(1) from erroring out when dependency files are deleted.

**-MT** *target\_file*

Add a target to the rules emitted by **-M**. The exact string provided will be written, including spaces and special characters.

**-MT** *fileA* **-MT** *fileB*

is equivalent to

**-MT** 'fileA fileB'.

If neither this nor **-MQ** is specified, the output file name is used.

- MQ** *target\_file*  
Same as **-MT**, but additionally escapes any special make(1) characters, essentially '\$'.
- o** *out\_file*, **--output** *out\_file*  
Write an object file to the given filename.
- p** *pad\_value*, **--pad-value** *pad\_value*  
When padding an image, pad with this value. The default is 0x00.
- r** *recursion\_depth*, **--recursion-depth** *recursion\_depth*  
Specifies the recursion depth at which RGBASM will assume being in an infinite loop.
- V**, **--version**  
Print the version of the program and exit.
- v**, **--verbose**  
Be verbose.
- W** *warning*, **--warning** *warning*  
Set warning flag *warning*. A warning message will be printed if *warning* is an unknown warning flag. See the **DIAGNOSTICS** section for a list of warnings.
- w**  
Disable all warning output, even when turned into errors.

## DIAGNOSTICS

Warnings are diagnostic messages that indicate possibly erroneous behavior that does not necessarily compromise the assembling process. The following options alter the way warnings are processed.

- Werror**  
Make all warnings into errors.
- Werror=**  
Make the specified warning into an error. A warning's name is appended (example: **-Werror=obsolete**), and this warning is implicitly enabled and turned into an error. This is an error if used with a meta warning, such as **-Werror=all**.

The following warnings are “meta” warnings, that enable a collection of other warnings. If a specific warning is toggled via a meta flag and a specific one, the more specific one takes priority. The position on the command-line acts as a tie breaker, the last one taking effect.

- Wall**  
This enables warnings that are likely to indicate an error or undesired behavior, and that can easily be fixed.
- Wextra**  
This enables extra warnings that are less likely to pose a problem, but that may still be wanted.
- Weverything**  
Enables literally every warning.

The following warnings are actual warning flags; with each description, the corresponding warning flag is included. Note that each of these flag also has a negation (for example, **-Wcharmap-redef** enables the warning that **-Wno-charmap-redef** disables). Only the non-default flag is listed here. Ignoring the “no-” prefix, entries are listed alphabetically.

- Wno-assert**  
Warns when **WARN**-type assertions fail. (See “Aborting the assembly process” in `rgbasm(5)` for **ASSERT**).

**-Wbuiltin-args**

Warn about incorrect arguments to built-in functions, such as **STRSUB()** with indexes outside of the string's bounds. This warning is enabled by **-Wall**.

**-Wcharmap-redef**

Warn when re-defining a charmap mapping. This warning is enabled by **-Wall**.

**-Wdiv**

Warn when dividing the smallest negative integer by -1, which yields itself due to integer overflow.

**-Wempty-macro-arg**

Warn when a macro argument is empty. This warning is enabled by **-Wextra**.

**-Wempty-strrpl**

Warn when **STRRPL()** is called with an empty string as its second argument (the substring to replace). This warning is enabled by **-Wall**.

**-Wlarge-constant**

Warn when a constant too large to fit in a signed 32-bit integer is encountered. This warning is enabled by **-Wall**.

**-Wlong-string**

Warn when a string too long to fit in internal buffers is encountered. This warning is enabled by **-Wall**.

**-Wmacro-shift**

Warn when shifting macro arguments past their limits. This warning is enabled by **-Wextra**.

**-Wno-obsolete**

Warn when obsolete constructs such as the **\_PI** constant or **PRINTT** directive are encountered.

**-Wshift**

Warn when shifting right a negative value. Use a division by  $2^N$  instead.

**-Wshift-amount**

Warn when a shift's operand is negative or greater than 32.

**-Wno-truncation**

Warn when an implicit truncation (for example, **db**) loses some bits.

**-Wno-user**

Warn when the **WARN** built-in is executed. (See "Aborting the assembly process" in `rgbasm(5)` for **WARN**).

**EXAMPLES**

You can assemble a source file in two ways.

Straightforward way:

```
$ rgbasm -o bar.o foo.asm
```

Pipes way:

```
$ cat foo.asm | rgbasm -o bar.o -
$ rgbasm -o bar.o - < foo.asm
```

The resulting object file is not yet a usable ROM image—it must first be run through `rgblink(1)` and then `rgbfix(1)`.

**BUGS**

Please report bugs on *GitHub*: <https://github.com/gbdev/rgbds/issues>

**SEE ALSO**

`rgbasm(5)`, `rgbfix(1)`, `rgbblink(1)`, `rgbds(5)`, `rgbds(7)`, `gbz80(7)`

**HISTORY**

**rgbasm** 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/gbdev/rgbds>