
rg.el Documentation

Release 2.0.1

David Landell

Feb 14, 2021

Contents

1 Usage	3
1.1 Installation	3
1.2 Searching	4
1.3 Results buffer	7
1.4 Search management	9
2 Configuration	11
2.1 Customization	11
2.2 Faces	13
2.3 Configuration functions	14
2.4 Configuration macros	14
2.5 Customizing the menu	16
3 Contribute	17
3.1 Pull requests	17
3.2 Tests	17
3.3 Documentation	18
4 License	21
4.1 GNU General Public License	21
Index	35

rg.el is an Emacs search package based on the [ripgrep](#) command line tool. It allows you to interactively create searches, doing automatic searches based on the editing context, refining and modifying search results and much more. It is also highly configurable to be able to fit different users' needs.

Throughout this manual this emacs package will be referred to as *rg* while the command line utility will be referred to as *ripgrep*.

If you are used to built-in Emacs `rgrep` command, transitioning to *rg* should be simple. *rg* provides a lot of extra features but the basics are similar.

The big benefit of using *ripgrep* instead of *grep* as a backend is speed. Especially when searching large source code repositories where *ripgrep* really shines. Please read [this blog post](#) for some speed comparisons with other tools.

1.1 Installation

This version of *rg* is supported on GNU Emacs 25.1 or later on Linux systems. It might work on older Emacsen and on other systems but such configurations are not tested. Patches for other OS:es are welcome.

MELPA

Packages are published on [MELPA Stable](#) and [MELPA](#). From within Emacs, run `M-x package-install [RET] rg [RET]` to install from those sources.

Enable default key bindings:

```
(rg-enable-default-bindings)
```

The above will enable the default key map under the default prefix key `C-c s`.

Manual

Releases can alternatively be downloaded from [GitHub](#) and installed manually. Put all elisp files in main directory in your load path and `require` the package in your init file.

```
(require 'rg)
(rg-enable-default-bindings)
```

You would also need to make sure all package requirements are met. For this version these are:

- **wgrep** 2.1.10
- **transient** 0.1.0
- **emacs** 25.1

`rg` is using autoloaded symbols which means it's also possible to defer loading if you have autoloading setup. That usually comes out of the box with `package-install`.

Lazy loading

For lazy loading you don't want to call directly into the package during startup. Use a setup similar to this instead:

```
(global-set-key (kbd "C-c s") #'rg-menu)
(with-eval-after-load 'rg
  ;; Your settings goes here.
)
```

If you don't want to use the transient menu interface, the following is needed to achieve lazy loading:

```
;; Workaround for emacs' lack of autoloaded keymaps.
;; This is essentially what use-package do.
(defun rg-autoload-keymap ()
  (interactive)
  (if (not (require 'rg nil t))
      (user-error (format "Cannot load rg"))
      (let ((key-vec (this-command-keys-vector)))
        (global-set-key key-vec rg-global-map)
        (setq unread-command-events
              (mapcar (lambda (ev) (cons t ev))
                      (listify-key-sequence key-vec))))))
(global-set-key (kbd "C-c s") #'rg-autoload-keymap)
(with-eval-after-load 'rg
  ;; Your settings goes here.
)
```

wgrep

This package use `wgrep` for editing capabilities in the `rg` results buffer. No setup is needed.

Interaction with the `ripgrep` configuration file

The `ripgrep` binary allows using a `configuration file` to set default values for command line flags. This package requires specific command line flags to function correctly and using a `ripgrep` configuration may conflict with these requirements. Therefore the configuration file is ignored by default. This can be changed by the `rg-ignore-ripgreprc` setting.

Note: Using the `ripgrep` configuration file may break functionality of this package if you are not careful.

1.2 Searching

Searching is done by invoking one of the different frontend commands. This package is built around recursive search based on three parameters; a single `directory`, `file type` filter, and a search `pattern`. These three parameters can interactively be selected or figured out automatically by the package, depending on which command that is used.

The underlying *ripgrep* binary has the file type filter concept built in. You have a high level of control over which files to search and which to ignore. This is partly what makes it so fast, ignoring uninteresting files.

In addition to the base parameters there are a lot of options that control how a search is done. These are typically selected from the *rg-menu* interface.

1.2.1 Case sensitivity

Considering case when searching is an important feature of any search tool. This package gives you a lot of control over how to handle case sensitive and case insensitive search. It can be forced to **on** or **off** and set to **smart case**. The latter is similar to the *ripgrep* `--smart-case` flag but is not using the flag directly. One thing to note about this is that the case insensitive setting controls the behavior when starting a new search. In the results buffer the setting is fixed to **on** or **off** but can be toggled easily with a key binding. See *rg-ignore-case* customization for the details of the configuration.

1.2.2 Interactive search

Two commands implements fully interactive search, where all the base parameters are selected from the mini buffer.

C-c s r (rg)

This command prompts for *query*, *file type* and *directory* and tries to suggest reasonable default values. The *query* string is interpreted as a regular expression. Default for *query* is the thing at point and for *directory* it is the current directory. If the type of the currently visited file is recognized, the corresponding *file type alias* is suggested as the *file type* parameter.

Invoking this command with the *universal argument* will trigger confirmation and potential modification of the *full command line* that will invoke the *ripgrep* binary.

C-c s t (rg-literal)

This command works in the same way as *rg* but interprets the *query* string literally and not as a regular expression.

Invoking this command with the *universal argument* will trigger confirmation and potential modification of the *full command line* that will invoke the *ripgrep* binary.

1.2.3 Project search

A common scenario is to search through a whole project while visiting a file in the project. This essentially means identifying the project root and use that as the top *directory* when invoking the *ripgrep* binary. *rg* supports several ways of identifying a project. Emacs' major project packages are supported including *projectile*, *find-file-in-project* and builtin *project.el*. If none of these are used, the fallback is Emacs' `vc-backend`.

C-c s p (rg-project)

Search in the current project. The *directory* is selected via one of Emacs' project packages while *query string* and *file type* are prompted for. The *query string* is interpreted as a regular expression.

1.2.4 Do what I mean

The **DWIM** family of search commands tries to be smart by figure out the search parameters from the context without prompting. Thanks to *ripgrep*'s speed, this allows for new ways of searching by invoking a *dwim* command and then *refine* the search from the results buffer.

These commands use the word (with the definition of word depending on context) under cursor as the *query* string. The *file type* parameter is taken from the type of the currently visited file. If the current file type can not be identified

all file types known to *ripgrep* are used. The fallback can be customized with *rg-default-alias-fallback*. The *directory* parameter varies between these commands.

M-x rg-dwim-project-dir

Do a **DWIM** search in the current *project*.

M-x rg-dwim-current-dir

Do a **DWIM** search in the current directory.

M-x rg-dwim-current-file

Do a **DWIM** search in the current file. The *current file* in this context is actually a file *pattern* exactly matching the current file name in a search starting from current directory. Most of the time this means a single file but if there are multiple files with the same name in a sub directory, those will be searched as well.

C-c s d (rg-dwim)

This command combines all the **DWIM** commands to one. The default search is in the *project dir*. With one *universal argument* *current directory* is used and with double *universal arguments* a *file search* is done.

1.2.5 File type aliases

File type aliases are used in *ripgrep* to filter out the files to search in. The *ripgrep* binary comes with a default set of aliases that can be extended or overridden from this package by customizing *rg-custom-type-aliases*.

An alias is a mapping between a name and a list of **glob patterns** matching the files of interest. Selecting an alias when searching is done with completing read of the defined aliases. It is also possible to enter a custom glob pattern if there is no suitable alias defined for the file type.

rg defines some internal aliases:

Name	Meaning
all	all defined types including <i>rg-custom-type-aliases</i>
everything	all files. No filtering on type is done.
custom	used internally in this package for mapping custom glob patterns.

Warning: Do not use any of the internal aliases in *rg-custom-type-aliases*. That would interfere with the package internal usage.

1.2.6 The menu

The global *prefix key* may be bound to a transient prefix command, which in normal words mean that the key binding will popup a menu. This package is using the same **popup menu backend** as the *magit* package. If you are familiar with *magit* this should feels like home.

The menu is mostly interesting when you want to give specific command line flags to the *ripgrep* binary. When you just want to do a quick search based on the defaults the menu basically acts as a normal keymap.

Pressing the *rg-menu prefix key* will popup the menu where command line flags can be selected before triggering the wanted search function. The menu can be customized via the transient API as usual. This package contains some shortcuts to directly add a new command to the menu when defining the command via the *rg-define-search* macro.

```
(rg-define-search rg-word
  :format literal
  :flags ("--word-regexp")
  :menu ("Custom" "w" "Word"))
```

The `:menu` keyword in the above invocation will trigger insertion of a new menu item bound to key `w` with description **Word**. The new menu item will be put under the **Custom** group. This group is not available in the original menu so it will be created.

The menu can be triggered from the *results buffer* with the `m` key. The commands in the menu differs, depending on from where it's triggered but the available options are the same. The menu does not show all options by default. The visible options can be controlled by the transient suffix levels documented [here](#).

1.3 Results buffer

The results of a search is shown in the results buffer. This buffer displays search parameters, the full command line and the output of the *ripgrep* binary. It supports basic navigation between search results editing of the file contents directly from the search buffer and also modification of the current search. The results buffer is a modified *compilation* buffer and some key bindings and functionality is inherited from the parent and from *grep mode*.

1.3.1 Navigation

Navigation works mostly as in *grep/compilation* buffers.

M-n (compilation-next-error)

Move to next line with a match.

M-p (compilation-previous-error)

Move to previous line with a match.

n (next-error-no-select)

Move to next line with a match, show that file in other buffer and highlight the match.

p (previous-error-no-select)

Move to previous line with a match, show that file in other buffer and highlight the match.

M-N (rg-next-file)

Move to next file header if the results is grouped under a file header (See *rg-group-result*).

M-P (rg-prev-file)

Move to previous file header if the results is grouped under a file header (See *rg-group-result*).

} (compilation-next-file)

Move first match in previous file.

{ (compilation-previous-file)

Move last match in previous file.

RET (compile-goto-error)

Visit match in file.

1.3.2 Refine search

From the results buffer it's easy to change the search parameters. Some bindings toggle a flag while others allow you to interactively change the *base parameters*.

- d** (**rg-rerun-change-dir**)
Interactively change search *directory*.
- f** (**rg-rerun-change-files**)
Interactively change searched *file types*.
- t** (**rg-rerun-change-literal**)
Interactively change *search string* interpret the string literally.
- r** (**rg-rerun-change-regexp**)
Interactively change *search string* interpret the string as a regular expression.

Tip: `rg-rerun-change-regexp` and `rg-rerun-change-literal` are used for switching between regular expression and literal search. So for quick switching between search modes with the same search string, just press the respective key and then RET.

- g** (**rg-recompile**)
Rerun the current search without changing any parameters.
- c** (**rg-rerun-toggle-case**)
Toggle case sensitivity of search. The state of the flag is shown in the **[case]** header field.
- i** (**rg-rerun-toggle-ignore**)
Toggle if ignore files are respected. The state of the flag is shown in the **[ign]** header field.

Tip: It is possible to create and bind your own toggle flags with the macro `rg-define-toggle`.

- m** (**rg-menu**)
Fire up *the menu* for full access to options and flags.

1.3.3 Full command line search

Some search commands (See `rg` or `rg-literal`) allow you to edit the final command line before invoking the search by giving a *universal argument*. This can be used to invoke features of the `ripgrep` binary that is not supported in this package's interface. This could be specific flags, searching in multiple directories etc.

Note: Using full command line search will disable refinement of the search from the result buffer.

1.3.4 History navigation

Each search result is stored in the search history, which is a per results buffer property. History can be navigated back and forward, the forward history is cleared when a new search is done.

- C-c <** (**rg-back-history**)
Navigate back in history.
- C-c >** (**rg-forward-history**)
Navigate forward in history.

Tip: The key bindings here are slightly inconvenient so invoking this via *the menu* by pressing `m b` and `m w` is more ergonomic.

1.3.5 Edit and apply (wgrep)

The results buffer supports inline editing via the `wgrep` package. This is setup automatically when `rg` is loaded.

e (wgrep-change-to-wgrep-mode)

Make the search results editable by enabling `wgrep` mode. When done press `C-c C-c` to commit your changes to the underlying files or `C-c C-k` to drop the changes.

1.4 Search management

The result buffer is named `*rg*` and `rg` reuse the same result buffer for new searches. If you want to store a search while continuing doing new searches there are two ways of doing that.

s (rg-save-search)

Save the search buffer by renaming it to a unique new name. This is available both outside and inside a result buffer. Outside of the result buffer it's bound to `C-c s`.

If you want to keep all search buffers until manually killed you can use this snippet in your init file.

```
(defadvice rg-run (before rg-run-before activate)
  (rg-save-search))
```

S (rg-save-search-as-name)

Save the search buffer and interactively give it a specific name. This is available both outside and inside a result buffer. Outside of the result buffer it's bound to `C-c s S`.

The default buffer name can be customized with `rg-buffer-name`. This setting considers dir local variables and it's even possible to use a function to get a really dynamic setup.

Having a lot of search buffers floating around can easily get messy. To help keeping this under control there is a search manager. The manager is simply a modified `ibuffer` that lists all the results buffers, shows some data about the searches and make it possible to kill of some unused etc.

L (rg-list-searches)

Open the search manager. This is available both in result buffer and globally bound to `C-c s l`.

C-c s k (rg-kill-saved-searches)

Kill all saved searches except for the one that matches `rg-buffer-name`. This is available both in result buffer and globally bound to `C-c s k`.

Warning: If you have a dynamic `rg-buffer-name` setup, only one buffer that matches your current criteria (dir locals or project for instance) will be kept. So be careful when killing saved searches to avoid losing important search results.

2.1 Customization

Customization is done via the Emacs customization system. The group `rg` is the main group of the package.

```
M-x customize-group [RET] rg [RET]
```

rg-executable [(executable-find "rg")]

The *ripgrep* executable to use. Could be an absolute path or just the base name if the executable is in the path. The default is using `executable-find` to locate the command. If you want to use this package with tramp it might be better to set it to just “`rg`” in order to let the OS find the binary where it’s invoked.

rg-custom-type-aliases [(("gyp" . "*.gyp *.gypi"))]

An association list that maps file type aliases to a space delimited string with file globs. These are combined with the *ripgrep* builtin file aliases.

Example:

```
(setq rg-custom-type-aliases
  '(("foo" . "*.foo *.bar")
    ("baz" . "*.baz *.qux")))
```

You may also add lambdas to `rg-custom-type-aliases` to add aliases dynamically based on mode, directory, project, etc.

```
(add-to-list
 'rg-custom-type-aliases
 (lambda ()
  (when (in-frontend-app)
   (cons "ui" "*.js *.hbs *.json"))))
```

rg-default-alias-fallback ["all"]

This setting controls the default alias used when no alias can be recognized for the current buffer. `all` or `everything` are reasonable values for this variable.

rg-command-line-flags [*nil*]

A list of command line flags that will be appended to the *ripgrep* command line. Must either be a list of flags or a function that returns a list of flags.

rg-group-result [*t*]

Controls the layout of the results buffer. If non *nil*, each file name is displayed once and matches are grouped under that filename instead of repeating the filename on each match. This is essentially the layout of the `--no-heading` *ripgrep* command line flag.

rg-show-columns [*nil*]

Controls if column numbers are used in the search result.

rg-ignore-case [*case-fold-search*]

Setting that controls if case sensitive search is made or not. It can essentially be **on**, **off** or **smart**. The **smart** setting will trigger an analyze of the search string and if it's all lower case, the search will be case *insensitive*, otherwise it will be case *sensitive*. The following values are valid:

- **case-fold-search** - A non *nil* value of *case-fold-search* will trigger smart case behavior.
- **smart** - Smart case behavior.
- **force** - Always ignore case.
- **nil** - Always consider case.

rg-hide-command [*t*]

Hide most of command line by default. This is enabled by default and can be set to *nil* to show full command line. This can be toggled in the results buffer by clicking on the command line.

rg-keymap-prefix ["*C-c s*"]

This variable sets the default prefix used for the global key bindings. Note that *rg-enable-default-bindings* needs to be invoked for the bindings to be enabled.

rg-use-transient-menu [*t*]

Controls whether *rg-menu* will be used by default or not. It's also possible to enable the menu explicitly with

```
(rg-enable-menu)
```

rg-show-header [*t*]

Controls if the search info header is shown in the result buffer. This is enabled by default but can be disabled by setting this variable to *nil*.

rg-buffer-name ["*rg*"]

Controls the name of the results buffer. It may be *string* or *function*. This name will be surrounded by *** to yield the final buffer name so if this setting is *foo* the buffer name will be **foo**. One useful case of using it is to have separate result buffers per project. One can set this variable in 'dir-locals' file or set it to function.

Example, this function will set results buffer name based on 'project-current':

```
(defun my-rg-buffer-name ()  
  (let ((p (project-current)))  
    (if p  
      (format "rg %s" (abbreviate-file-name (cdr p)))  
      "rg"))))
```

rg-ignore-ripgrepc [*t*]

Controls if the *ripgrepc* file should be ignored or not. If *nil*, the config file will be used, otherwise it will be ignored. The default is to ignore this file in order to avoid that conflicting settings have impact on this package's behavior. Setting this to *nil* may affect core functionality of this package. Especially changing colors can affect parsing of the output and result in a broken results buffer.

2.1.1 Position numbers alignment

When operating *rg* in grouped output mode (*rg-group-result* is non nil), it's possible to control how the line and column numbers are displayed in the result buffer.

Example settings:

```
(setq rg-align-position-numbers t)
(setq rg-align-line-number-field-length 3)
(setq rg-align-column-number-field-length 3)
(setq rg-align-line-column-separator "#")
(setq rg-align-position-content-separator "|")
```

Will yield the following format:

```
File: matched_file.foo
  1#  2|match1
888# 10|match2
```

rg-align-position-numbers [t]

Setting this to *t* will align line and column numbers in columns padded with white space.

rg-align-line-number-field-length [4]

Defines the length of the line number field.

rg-align-column-number-field-length [3]

Defines the length of the column number field.

rg-align-line-column-separator [" "]

Separator string used between line and column numbers. *nil* means use default separator from *ripgrep*.

rg-align-position-content-separator [" "]

Separator string used between the position numbers and matched content. *nil* means use default separator from *ripgrep*.

2.2 Faces

All faces are in the subgroup *rg-face* of the main group *rg*.

```
M-x customize-group [RET] rg-face [RET]
```

2.2.1 Results buffer

rg-match-face [match]

Face used to highlight matches in result.

rg-error-face [compilation-error]

Face used to highlight errors when invoking *ripgrep*.

rg-context-face [shadow]

Face used to highlight context lines in *ripgrep* output when `--context-lines` flag is used.

rg-info-face [compilation-info]

Face used to highlight general info in results buffer. For instance the number of matches found.

rg-warning-face [compilation-warning]

Face used to highlight warnings in the *ripgrep* output.

rg-filename-face [**rg-info-face**]

Face used to highlight filenames in the output.

rg-file-tag-face [**rg-info-face**]

Face used for the `File:` tag in grouped results output.

rg-line-number-face [**compilation-line-number**]

Face used on line numbers.

rg-column-number-face [**compilation-column-number**]

Face used on column numbers.

rg-match-position-face [**default**]

Face added to file positions. This is the start of a matching line and depending on configuration may be, file name, column number and line number.

2.2.2 Header line

rg-toggle-on-face [**rg-file-tag-face**]

Face used for flags that are toggled on.

rg-toggle-off-face [**rg-error-face**]

Face used for flags that are toggled off.

rg-literal-face [**rg-filename-face**]

Face used the on the `literal` marker in the header line.

rg-regexp-face [**compilation-line-number**]

Face used the on the `regexp` marker in the header line.

2.3 Configuration functions

(**rg-enable-default-bindings** *&optional, prefix*)

Enable the default keyboard bindings for the package with prefix key. If `rg-use-transient-menu` is on this will enable the menu instead of activating the global bindings. If `prefix` is not provided `rg-keymap-prefix` will be used.

(**rg-enable-menu** *&optional, prefix*)

Enable the `rg-menu` with prefix key. This bypass `rg-use-transient-menu` setting. If `prefix` is not provided `rg-keymap-prefix` will be used.

(**rg-use-old-defaults**)

This function is provided to keep backwards compatibility with versions older than 2.0.0. In this version default settings as well as key bindings changed and to bring back the old defaults call this function in your init file.

2.4 Configuration macros

(**rg-define-toggle** *flag, &optional, key, default*)

This is a macro that can be used to define custom `ripgrep` flag toggling functions in the result buffer. The macro takes the flag (and potential value) as an argument and optionally binds the toggle function to a key. If `default` is non nil the flag is used by default.

The function defined by this macro will be named as the flag name stripped with leading dashes and prefixed with `rg-custom-toggle-flag-`.

```
(rg-define-toggle "--uu" "I" t)
```

Creates a function named `rg-custom-toggle-flag-uu` that is on by default and bound to `I` in `rg` result buffer.

```
(rg-define-toggle "--context 3" (kbd "C-c c"))
```

Creates a function named `rg-custom-toggle-flag-context` that is off by default and bound to `C-c c` in `rg` result buffer.

(rg-define-search name, &rest, args)

This macro can be used to define custom search functions in a declarative style. Default implementations for common behavior is available and custom forms can also be used.

It optionally starts with a string that is used as the docstring for the defined function. The rest of the arguments contain key value pairs according to the specification below. All keys are optional with specified default if left out.

- **:query** - Method for retrieving the search string. Allowed values are `point` which means extract thing at point and `ask` which means prompt the user for a string. Any form that evaluates to a string is allowed. Default is `ask`.
- **:format** - Specifies if `:query` is interpreted literally (`literal`) or as a regexp (`regexp`). If it is a form, eg. `(not current-prefix-arg)`, and is non-nil the `:query` is interpreted literally, otherwise as a regexp. Default is `regexp`.
- **:files** - Form that evaluates to a file alias or custom file glob. `current` means extract alias from current buffer file name, `ask` will prompt the user. Default is `ask`.
- **:dir** - Root search directory. Allowed values are `ask` for user prompt, `current` for current dir and `project` for project root. Any form that evaluates to a directory string is also allowed. Default is `ask`.
- **:confirm** - `never`, `always`, or `prefix` are allowed values. Specifies if the the final search command line string can be modified and confirmed the user. Default is `never`.
- **:flags** - `ask` or a list of command line flags that will be used when invoking the search.
- **:menu** - Bind the command into `rg-menu`. Must be a list with three items in it. The first item is the description of the group in witch the new command will appear. If the group does not exist a new will be created. The second item is the key binding for this new command (ether a key vector or a key description string) and the third item is the description of the command that will appear in the menu.

Examples:

```
(rg-define-search search-everything-at-home
  "Search files including hidden in home directory"
  :query ask
  :format literal
  :files "everything"
  :flags ("--hidden")
  :dir (getenv "HOME")
  :menu ("Search" "h" "Home"))

(rg-define-search rg-emacs
  "Search the emacs lisp source code."
  :dir "/usr/share/emacs/25.2/lisp/"
  :flags '("-z")
  :files "*. {el,el.gz}"
  :menu ("Custom" "L" "lisp"))
```

2.5 Customizing the menu

Please refer to the [transient](#) documentation for customizing the menu.

Contributions are very welcome. Development is done in the [GitHub repository](#). If you find a bug, please report it in the [issue tracker](#).

3.1 Pull requests

If you want to submit a patch, please submit a [GitHub pull request](#). If you want to submit any larger code changes, please create an issue first for discussion. Some features does not fit well into this package and there is also good to agree on the general design before doing any major work.

The minimum requirements for a pull request to be accepted is that all existing tests pass and test coverage should not decrease. Often a patch also needs additional tests, new/changed documentation etc.

Don't strive to submit a perfect pull request directly. It's often better to submit something simple that shows the main direction of the new code in order to discuss the best way to proceed and what additions are needed.

3.2 Tests

[Cask](#) is used for testing. The tests are written using the Emacs built in ERT framework and executed with [ert runner](#). There are also compilation tests, style check, package verification etc.

3.2.1 Setup

- [Install cask](#)
- Install all developer dependencies:

```
make deps
```

3.2.2 Running

- Run the whole test suite:

```
make test
```

- Run only the unit/integration tests:

```
make ert-test
```

- Manually test the package with Emacs:

```
cask emacs -Q -L . --eval="(progn (require 'rg) (enable-default-bindings))"
```

3.3 Documentation

The documentation is written in org mode. The export target is `restructured text` suitable for the `Sphinx` documentation generator. Sphinx is used to export the output from org mode to info and HTML documentation. The resulting `.rst` files are used for the online documentation on <https://readthedocs.io>.

The ReadTheDocs documentation is generated after committing to the main repository but the info manual needs to be created locally and submitted as part of a documentation patch. It's advisable to build both locally and verify the output to make sure the changes looks OK.

3.3.1 Setup

- Install Sphinx

```
apt-get install python3-sphinx
```

3.3.2 Building

- HTML documentation

```
make html
```

Open `docs/rst/_build/html/index.html` in a browser.

- Info documentation

```
make info
```

To view in emacs:

```
C-u M-x info [RET]
```

Then select the `docs/rst/_build/info/rgel.info` file.

3.3.3 Create a patch

- Edit the org files in `docs` directory.
- When everything looks good in both HTML and info.

```
make info
```

- Commit both the changed org files and info file.
- Push and create PR.

rg is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

rg is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *GNU General Public License* for more details.

4.1 GNU General Public License

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you

(continues on next page)

(continued from previous page)

have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this

(continues on next page)

(continued from previous page)

License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's

(continues on next page)

(continued from previous page)

System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

(continues on next page)

(continued from previous page)

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

(continues on next page)

(continued from previous page)

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

(continues on next page)

(continued from previous page)

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the

(continues on next page)

(continued from previous page)

terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is

(continues on next page)

(continued from previous page)

reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted

(continues on next page)

(continued from previous page)

by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting

(continues on next page)

(continued from previous page)

any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY

(continues on next page)

(continued from previous page)

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

(continues on next page)

(continued from previous page)

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

C

compilation-next-error (*command*), 7
 compilation-next-file (*command*), 7
 compilation-previous-error (*command*), 7
 compilation-previous-file (*command*), 7
 compile-goto-error (*command*), 7

N

next-error-no-select (*command*), 7

P

previous-error-no-select (*command*), 7

R

rg (*command*), 5
 rg-align-column-number-field-length (*customize option*), 13
 rg-align-line-column-separator (*customize option*), 13
 rg-align-line-number-field-length (*customize option*), 13
 rg-align-position-content-separator (*customize option*), 13
 rg-align-position-numbers (*customize option*), 13
 rg-back-history (*command*), 8
 rg-buffer-name (*customize option*), 12
 rg-column-number-face (*customize option*), 14
 rg-command-line-flags (*customize option*), 11
 rg-context-face (*customize option*), 13
 rg-custom-type-aliases (*customize option*), 11
 rg-default-alias-fallback (*customize option*), 11
 rg-define-search (*function*), 15
 rg-define-toggle (*function*), 14
 rg-dwim (*command*), 6
 rg-dwim-current-dir (*command*), 6
 rg-dwim-current-file (*command*), 6
 rg-dwim-project-dir (*command*), 6
 rg-enable-default-bindings (*function*), 14
 rg-enable-menu (*function*), 14
 rg-error-face (*customize option*), 13
 rg-executable (*customize option*), 11
 rg-file-tag-face (*customize option*), 14
 rg-filename-face (*customize option*), 14
 rg-forward-history (*command*), 8
 rg-group-result (*customize option*), 12
 rg-hide-command (*customize option*), 12
 rg-ignore-case (*customize option*), 12
 rg-ignore-ripgrep (*customize option*), 12
 rg-info-face (*customize option*), 13
 rg-keymap-prefix (*customize option*), 12
 rg-kill-saved-searches (*command*), 9
 rg-line-number-face (*customize option*), 14
 rg-list-searches (*command*), 9
 rg-literal (*command*), 5
 rg-literal-face (*customize option*), 14
 rg-match-face (*customize option*), 13
 rg-match-position-face (*customize option*), 14
 rg-menu (*command*), 8
 rg-next-file (*command*), 7
 rg-prev-file (*command*), 7
 rg-project (*command*), 5
 rg-recompile (*command*), 8
 rg-regexp-face (*customize option*), 14
 rg-rerun-change-dir (*command*), 7
 rg-rerun-change-files (*command*), 8
 rg-rerun-change-literal (*command*), 8
 rg-rerun-change-regexp (*command*), 8
 rg-rerun-toggle-case (*command*), 8
 rg-rerun-toggle-ignore (*command*), 8
 rg-save-search (*command*), 9
 rg-save-search-as-name (*command*), 9
 rg-show-columns (*customize option*), 12
 rg-show-header (*customize option*), 12
 rg-toggle-off-face (*customize option*), 14
 rg-toggle-on-face (*customize option*), 14
 rg-use-old-defaults (*function*), 14
 rg-use-transient-menu (*customize option*), 12

rg-warning-face (*customize option*), 13

W

wgrep-change-to-wgrep-mode (*command*), 9