ag.el Documentation

Release 0.45

Wilfred Hughes

Feb 25, 2018

Contents

Ag.el allows you to search using `ag` from inside Emacs. You can filter by file type, edit results inline, or find files.

Ag.el tries very hard to be Do-What-I-Mean, and will make intelligent suggestions about what to search and which directories to search in.

Source on GitHub.

Bug tracking on GitHub.

Contents:

CHAPTER 1

Installation

1.1 Operating System

ag.el currently works on Linux, Mac and Windows. Patches or bug reports for other platforms are welcome.

1.2 Emacs

We currently support Emacs 23.4 or above, patches for other emacsen are also welcome.

1.3 Ag

You will need the `ag` binary. See the installation instructions on ag's GitHub repo. A precompiled Windows binary is also available.

1.4 Ag.el

Afterwards, you can install ag.el from MELPA (the recommended approach).

Functions are autoloaded, so `(require 'ag)` is unnecessary.

If you want to install it manually, add the following to your .emacs.d:

```
(add-to-list 'load-path "/path/to/ag.el")
(require 'ag)
```

CHAPTER 2

Usage

2.1 Running a search

You will now have the following interactive commands available for performing searches:

- `ag`
- `ag-files`
- `ag-regexp`
- `ag-project`
- `ag-project-files`
- `ag-project-regexp`

`*-project` commands automatically choose the directory to search, automatically detecting git, Subversion and Mercurial project roots.

`*-regexp` commands allow you to specify a PCRE pattern for your search term.

`*-files` commands allow you to specify a PCRE pattern for file names to search in. By default, ag searches in all files. Note that in both cases, ag ignores files that are ignored by your VCS (e.g. things mentioned in .gitignore).

2.2 The results buffer

In the search results buffer, you can move between results by pressing `n` and `p`, and you can visit the file by pressing `<return>` or clicking.

You can run the search again by pressing `g`, close the buffer with `q`, or kill the buffer with `k`.

You can activate `next-error-follow-minor-mode` with `C-c C-f`. With this minor mode enabled, moving in the results buffer will make Emacs automatically display the search result at point.

If you've [configured wgrep](#editing-the-results-inline) you can use `C-c C-p` to make the buffer writable and edit the results inline.

Of course, `C-h m` inside a results buffer will show all the keybindings available to you.

2.3 Search for file names

`ag` supports an option `-g` that lets you to list file names matching PCRE patterns. It is analogical to `find`, but comes with all the nice features of `ag` such as automatically ignoring all the vcs files. You can search for files matching a pattern using functions

- `ag-dired`
- `ag-dired-regexp`
- `ag-project-dired`
- `ag-project-dired-regexp`

The results are presented as a `dired-mode` buffer. The analogical interface to `find` in emacs is `find-dired`.

CHAPTER 3

Configuration

3.1 Highlighting results

ag.el supports highlighting results for ag 0.14 or later. Previous versions of ag don't support the `--color-match` argument.

If your version of ag is recent enough, you can add highlighting by adding the following to your Emacs configuration:

```
(setq ag-highlight-search t)
```

3.2 Path to the ag executable

ag.el assumes that the ag executable is in one of the directories on `exec-path`. Generally, this is sufficient.

However, you may find that you can run ag in a terminal but ag.el isn't finding the ag executable. This is common on Mac OS X. You'll need to update `exec-path` to match your terminal. The best way to do this is to install exec-path-from-shell (available on MELPA).

Alternatively, you can do this yourself by putting the following code in your Emacs configuration:

```
(defun set-exec-path-from-shell-PATH ()
  "Set up Emacs' `exec-path' and PATH environment variable to match that used by the␣
↪user's shell.

This is particularly useful under Mac OSX, where GUI apps are not started from a␣
↪shell."
  (interactive)
  (let ((path-from-shell (replace-regexp-in-string "[ \t\n]*$" "" (shell-command-to-
↪string "$SHELL --login -i -c 'echo $PATH'"))))
    (setenv "PATH" path-from-shell)
    (setq exec-path (split-string path-from-shell path-separator))))

(set-exec-path-from-shell-PATH)
```

Finally, as a last resort, you can specify the path to ag explicitly. This might be the case if:

- you are are in an environment where, for whatever reason, you can't easily change the path to include ag
- you are on windows, where the executable name ends in `.exe`.
- you have multiple versions of ag or want to use some other executable that works the same as ag.

To change the ag executable used:

```
(setq ag-executable "C:/Wherever/I/Installed/Ag/ag.exe")
```

3.3 Visiting the results

By default, ag.el will open results in a different window in the frame, so the results buffer is still visible. You can override this so the results buffer is hidden and the selected result is shown in its place:

```
(setq ag-reuse-window 't)
```

3.4 Overriding the arguments passed to ag

ag.el provides a customisable variable `ag-arguments`.

For temporary changes to arguments, the search commands will prompt you for arguments if you call them with a prefix argument. For example, `C-u M-x ag`.

3.5 Hooks

ag.el provides `ag-mode-hook` which is run when you start a search, and `ag-search-finished-hook` which is run when the search completes.

3.6 Multiple search buffers

Ag.el provides the interactive commands for closing old search buffers:

- `ag-kill-buffers`
- `ag-kill-other-buffers`

Alternatively, you can make ag.el reuse the same `*ag*` buffer for all your searches:

```
(setq ag-reuse-buffers 't)
```

CHAPTER 4

Extensions

4.1 Using with Projectile

Projectile supports ag.el. If you have Projectile installed, `C-c p s s` runs `ag` on your project or `ag-regexp` if a prefix argument is provided.

4.2 Customising the project root

By default, `ag-project` and `ag-project-regexp` use the root of the VCS repo as the directory to search in. You can override this by setting or customising `ag-project-root-function`.

4.3 Editing the results inline

wgrep has support for ag.el. If you install wgrep-ag (available on MELPA), you can simply run `wgrep-change-to-wgrep-mode` and edit the `*ag*` buffer. Press `C-x C-s` when you're done to make the changes to buffers.

4.4 Focusing and filtering the results

winnow can be used to focus or filter the results from ag. With the minor mode enabled, `x` excludes unwanted results, and `m` selects only the lines that match.

4.5 Writing Your Own

You can use `ag`, `ag-project` and so on from an elisp function. `ag/FOO` functions are private and are more likely to change. Please file a bug if you find a use for internal functions that you can't do otherwise.

CHAPTER 5

Developing ag.el

Contributing to ag.el is just a matter of editing the ag.el file and sending a pull request on GitHub.

5.1 Using flycheck (optional)

If you're using flycheck, you can configure it to check ag.el.

First, you will need to install Cask. You can then install the ag.el dependencies:

```
$ cask install
```

Flycheck will now check ag.el, including arguments of functions in included libraries.

CHAPTER 6

Changelog

6.1 Previous Versions

6.2 master

Ag search commands now show N lines of context when called with numeric prefix argument N.

Ag search commands can now group output. See `ag-group-matches`.

Docstring improvements.

6.2.1 0.47

Added a workaround for an ag bug on windows where results were shown without filenames (see issue #97). Note this only applies when `ag-highlight-search` is nil.

Fixed an issue where `ag-projects-regexp` escaped its input, resulting in literal searches rather than regexp searches (see #94).

Detection of project roots now supports bzr.

6.2.2 0.46

Replaced calls to `ido-completing-read` with `completing-read`. This allows helm users to use helm completion. To continue using ido for completion, please install ido-ubiquitous-mode. This only affects `ag-files` and `ag-project-files`.

Fixed an issue where pressing `k` would kill the search results buffer, even if `evil-mode` was active. `k` now only kills the results buffer if you're not using evil.

Search results buffers now include a summary of the total number of results and the number of files matched.

When choosing a search term, the prompt is now empty. Ag.el will use the default if no search term is given, but saves a keystroke if you don't want the default. You can edit the default value by pressing `M-n` in the minibuffer.

Added `ag-search-finished-hook`, a hook that's run when the search is completed.

6.2.3 0.45

Fixed another case where `ag-dired*` commands ignored `ag-executable`.

Fixed an issue with `ag-dired` where inputs would be quoted twice.

Added `ag-ignore-list` which allows you specify patterns to ignore.

Fixed a crash with `ag-files` when using a built-in file type.

6.2.4 0.44

Fixed a bug with `ag-dired*` commands where it ignored `ag-executable`.

Improved alignment of the output from `ag-dired*` commands.

`ag-dired*` commands now call `dired-before-readin-hook` and `dired-after-readin-hook` where appropriate.

Fixed a bug with path name escaping in `ag-dired*` commands.

Fixed a bug with calling ag on Windows such that you can't jump to results from the results buffer (you just get 'No Error Here').

Minor docs improvements.

6.2.5 0.43

When calling ag with a prefix argument, we now place the point after the last argument in the minibuffer. See #48.

Minor docstring improvements.

6.2.6 0.42

When passing a prefix argument, ag.el now presents you with the whole command so you can edit any part, as a string. See #38.

Documentation and docstring improvements, mostly around clarifying what regular expression syntax ag.el expects.

6.2.7 0.41

Added a setting `ag-executable` which allows you to override the name or path of the ag executable.

Added support for Emacs 23.4.

Buffers created by ag.el are now always named `*ag: FOO*`.

`ag-dired` now respects the value of `ag-reuse-buffers`.

6.2.8 0.40

`ag-project-regexp` now defaults to the escaped value at point, as an escaped regular expression. For example, if point is at `foo-bar`, then the suggested search regexp is `foo\-bar`.

`ag-regexp-project-at-point` is now just an obsolete alias for `ag-project-regexp`.

6.2.9 0.39

The commands `ag`, `ag-files`, `ag-regexp`, `ag-project`, `ag-project-files` and `ag-project-regexp` can now take a prefix argument. For example, `C-u M-x ag`. If given a prefix argument, you are also prompted for the flags to pass ag itself.

6.2.10 0.38

`ag-dired` and `ag-project-dired` should now work on Mac OS X (previously we assumed xargs supported GNU extensions).

6.2.11 0.37

Added `ag-dired` and `ag-project-dired` to search for files matching a pattern.

6.2.12 0.36

Fixed a bug in `ag-regexp` and `ag-project-regexp` due to an internal API change (`ag/search` now uses keyword arguments).

6.2.13 0.35

Added the `ag-files` and `ag-project-files` commands.

Note that the *internal API changed* in this release: `ag/search` now takes `regexp` as a keyword argument instead of a positional argument. I'm not aware of any external packages depending on this, so I'm not incrementing the major version.

6.2.14 0.34

Specifying the path as an argument to ag, allowing ag.el to do searches on Windows.

6.2.15 0.33

Fixed a bug with ag.el not searching if `shell-command-switch` had been modified by the user.

6.2.16 0.32

Adding `ag-project-root-function` which allows users to override how ag.el finds the root of a project.

6.2.17 0.31

Ag.el faces (which are `ag-match-face` and `ag-hit-face``x) are defined with ``defface`, so you can use `customize-face` on them.

6.2.18 0.30

Improved quoting of arguments passed to ag.

6.2.19 0.29

Added customisable variable `ag-reuse-window`. If set to `t` (defaults to `nil`) then selecting a search result hides the results buffer and shows the match, rather than using a different window in the frame.

6.2.20 0.28

`-project` functions now handle the case of multiple nested VCS repositories. Ag.el now takes the most deepest subdirectory, so if `/foo/bar` is a subversion repo that contains a git repo `/foo/bar/baz`, ag.el will search `/foo/bar/baz`.

6.2.21 0.27

Ag.el autopopulates the minibuffer with the text at point, or the active selection. If this text was read-only, the minibuffer text would also be read-only. It's now always possible to edit the text in the minibuffer.

6.2.22 0.26

Fixed a crash when refreshing a search buffer by pressing `g`.

6.2.23 0.25

Added commands `ag-kill-buffers` and `ag-kill-other-buffers` to close old search result buffers. Also added a customisable variable `ag-reuse-buffers` so users can optionally stop ag.el creating multiple buffers.

6.2.24 0.24

Search results buffers now take the form **ag text:something dir:~/some/path**, so new searches will create new buffers.

6.2.25 0.23

ag.el now detects the project root for Mercurial repositories in the `ag-project*` commands.

6.2.26 0.22

The keys `n` and `p` now move between matches, similar to the behaviour of dired.

6.2.27 0.21

Added a new face `ag-hit-face` to distinguish from `ag-match-face`.

6.2.28 0.20

Fixed `next-error` and `previous-error` not working with ag.el (broken in v0.18).

6.2.29 0.19

`ag` now has a default search term of the symbol at point.

6.2.30 0.18

Search results are now highlighted as information, rather than errors. The ag output is now more consistent with grep.el.

6.2.31 0.17

The interactive functions provided by ag.el are now autoloaded.

6.2.32 0.16

Removed the unused variable `ag-last-buffer`

6.2.33 0.15

Fixed `ag-project` and `ag-project-regexp` not working in buffers that aren't associated with a specific file, such as dired and magit buffers.

6.2.34 0.14

The compilation mode regexp is now more accurate, so you should no longer get 'compilation-next-error: No error here' when trying to open a file in the results list.

6.2.35 0.13

Current stable ag (0.13.1) doesn't support `--color-match`, ag.el now only highlights when `ag-highlight-search` is non-nil (the default is nil).

If you're upgrading ag.el and your ag version is 0.14 or higher, you need to explicitly enable highlighting:

```
(setq ag-highlight-search t)
```