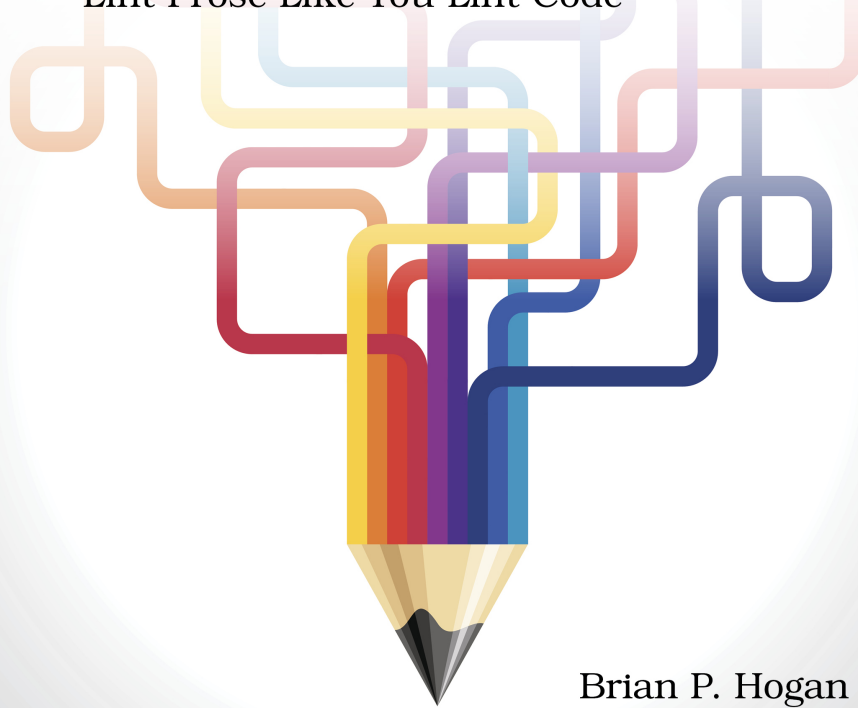


The
Pragmatic
Programmers

Pragmatic
exPress

Write Better with Vale

Automate Your Style Guides and
Lint Prose Like You Lint Code



Brian P. Hogan

edited by Susannah Davidson

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit <https://www.pragprog.com>.

Copyright © The Pragmatic Programmers, LLC.

Using Existing Styles

Building a style guide from scratch is no small task. That's why lots of content publishers adopt a well-known one and adapt it. Microsoft, Google, Grafana, RedHat, and GitLab have all published their style guides either on their public web sites or as part of their documentation repositories on GitHub. Best of all, these companies use Vale to enforce those styles, so you can start with these, peek at their rules, and mix and match to find a good foundation.

That's what you'll do in this chapter. You'll start with a single style. Then you'll bring in another style and manage conflicting rules. Then you'll look at other styles that make content more inclusive and readable. To do that, you'll need some content to improve.

In the book's companion files, you'll find a directory called `cli-tips`, which is a collection of tutorials for some command-line Linux tools. This content is in rough shape, but you'll use Vale to work out the inconsistencies. You'll spend this and the next chapter building a set of rules for your own content. In this chapter you'll use a handful of existing style rules.

Visit the book's web site ¹ and download and extract the companion files for the book. Open a new Terminal and navigate to the `cli-tips` folder. You'll see five files:

```
$ cd cli-tips
$ ls
awk.md  date.md  find.md  grep.md  sed.md
```

Look at each file and you'll see that there are some quality concerns. It's time to get to work fixing things up.

1. <https://thevalebook.com>

		capitalization.	
54:1	warning	Try to avoid using first-person plural like 'We'.	Google.We
54:35	warning	Try to avoid using first-person plural like 'Let's'.	Google.We
72:1	suggestion	Spell out 'AWK', if it's unfamiliar to the audience.	Google.Acronyms
72:70	error	Don't use exclamation points in text.	Google.Exclamation

✖ 2 errors, 5 warnings and 2 suggestions in 1 file.

Vale's output makes it clear that this content is not aligned with Google's style. Now you have a roadmap of what to clean up. But don't make any changes yet. First, take a look at some of the definitions.

Exploring the Style

The styles folder in your project now has a new Google folder. That folder contains files in YAML format that define the rules for the style.

When you scanned the `awk.md` file, the last suggestion you saw was this one:

```
72:70 error      Don't use exclamation points    Google.Exclamation
                        in text.
```

Look in `styles/Google` and you'll see a file called `Exclamation.yml` that contains the rule. Open the file in your editor to see the rule's definition:

```
use_existing_styles/single/styles/Google/Exclamation.yml
extends: existence
message: "Don't use exclamation points in text."
link: "https://developers.google.com/style/exclamation-points"
nonword: true
level: error
action:
  name: edit
  params:
    - trim_right
    - "!"
tokens:
  - '\w+! (?!\s|$)'
```

The rule includes the rule type, or “check”, a message, a link to the rule in the style guide, and most importantly, some kind of regular expression to match. In the next chapter, you'll become more familiar with these definitions as you create your own rules from scratch.

Some Vale rules include an action section that text editors and other tools can use to correct the errors Vale finds. In this case, the definition instructs the tool to remove the exclamation point. It's up to each editor or tool to handle the actual implementation. You'll explore using Vale with editors in [See Errors As You Work, on page ?](#).

Changing a Rule's Error Level

As you saw in [Controlling Error Levels, on page ?](#), you can change a rule's severity in the `.vale.ini` file. The Google package you added to your project has many rules split into individual YAML files. You can change the severity of each rule by specifying it using the rule name listed in the output.

For example, the rule about exclamation points has the name `Google.Exclamation`:

```
72:70 error          Don't use exclamation points    Google.Exclamation
                               in text.
```

This rule raise an error rather than a warning or suggestion. While you could directly edit the YAML file, doing so means you have to track changes and reapply them if you ever update the rules. Upstream packages do change as companies evolve their styles.

Instead, use the `.vale.ini` file to raise warnings instead of errors:

```
use_existing_styles/single/.vale.ini
```

```
[*.{md}]
```

```
BasedOnStyles = Vale, Google
```

```
➤ Google.Exclamation = warning
```

With the change in place, run Vale against the `awk.md` file again:

```
$ vale awk.md
```

The output shows warnings instead of errors this time:

```
...
72:70 warning       Don't use exclamation points    Google.Exclamation
                               in text.
```

You can control the severity level for errors, or even turn individual rules on and off. That'll come in handy next, as you incorporate another style guide into your project.