

Stratis Coding Style Guidelines

Last modified: 01/05/2017

Contents

1	Introduction	1
2	Style (Rust)	1
3	Style (Python)	2

Asking Questions and Making Changes to this Document

This document can be found in the stratis-docs repo, and is written using LyX 2.2.2. Please ask any questions by opening an issue, and propose changes as pull requests.

1 Introduction

A consistent style helps legibility and saves developers from needing to think about issues that are not really important.

In order to promote this, this document specifies some coding conventions that go beyond the standard Rust coding style, as enforced by *rustfmt*.

2 Style (Rust)

1. *rustfmt* (link) defines spacing, brace placement, and other formatting, and should be followed (or automatically applied) for new Stratis code.
2. All new files must start with the 3-line license as a comment.
3. “use” declarations within a module (such as a source file) should be grouped into three sections:
 - (a) declarations from “std”

- (b) declarations from other external dependencies
- (c) declarations from other modules in the same crate

as needed, separated by blank lines. Ordering within blocks is alphabetical.

4. Use “expect()” instead of “unwrap()” to unwrap things that should always succeed. “unwrap()” is not allowed in new code.
 - (a) Output from Linux kernel APIs is considered “super-stable” and should be “expect”ed instead of returning a parse error.
 - (b) “expect()” text should say what “should never fail” condition was not met to trigger it.
5. When importing multiple items from a module, whether to group these with “{ }” on a single line, or across multiple declarations, is not yet specified.
6. TBD: Recommendations on when it’s time to break up a single module (file) into submodules
7. TBD: Recommendations on when to define const variables versus when to use string literals

3 Style (Python)

TBD