Style Guidelines

This document collects the emerging principles, conventions, abstractions, and best practices for writing Rust code.

Since Rust is evolving at a rapid pace, these guidelines are preliminary. The hope is that writing them down explicitly will help drive discussion, consensus and adoption.

Whenever feasible, guidelines provide specific examples from Rust's standard libraries.

Guideline statuses

Every guideline has a status:

- **[FIXME]**: Marks places where there is more work to be done. In some cases, that just means going through the RFC process.

- **[FIXME #NNNNN]**: Like **[FIXME]**, but links to the issue tracker.

- **[RFC #NNNN]**: Marks accepted guidelines, linking to the rust-lang RFC establishing them.

Guideline stabilization

One purpose of these guidelines is to reach decisions on a number of cross-cutting API and stylistic choices. Discussion and development of the guidelines will happen primarily on http://discuss.rust-lang.org/ (http://discuss.rust-lang.org/), using the Guidelines category. Discussion can also occur on the guidelines issue tracker (https://github.com /rust-lang/rust-guidelines).

Guidelines that are under development or discussion will be marked with the status **[FIXME]**, with a link to the issue tracker when appropriate.

Once a concrete guideline is ready to be proposed, it should be filed as an FIXME: needs RFC (https://github.com/rust-lang/rfcs). If the RFC is accepted, the official guidelines will be updated to match, and will include the tag **[RFC #NNNN]** linking to the RFC document.

What's in this document

This document is broken into four parts:

- **Style (style/README.html)** provides a set of rules governing naming conventions, whitespace, and other stylistic issues.

- **Guidelines by Rust feature (features/README.html)** places the focus on each of Rust's features, starting from expressions and working the way out toward crates, dispensing guidelines relevant to each.

- **Topical guidelines and patterns**. The rest of the document proceeds by cross-cutting topic, starting with Ownership and resources (ownership/README.html).

- **APIs for a changing Rust (changing/README.html)** discusses the forward-compatibility hazards, especially those that interact with the pre-1.0 library stabilization process.

> **[FIXME]** Add cross-references throughout this document to the tutorial, reference manual, and other guides.
>
> **[FIXME]** What are some *non*-goals, *non*-principles, or *anti*-patterns that we should document?