Changelog

[Unreleased]

Fixed

- Fixes issue where wrapped strings would be incorrectly indented in macro defs when `format_strings` was enabled #4036

[1.4.38] 2021-10-20

Changed

- Switched from `rustc-ap-*` crates to `rustc_private` for consumption model of rustc internals
- `annotate-snippets` updated to v0.8 PR #4762
- Greatly improved the performance of `cargo fmt` in large workspaces utilizing the `--all` flag by updating to a newer version of `cargo_metadata` that leverages updated `cargo` output from v1.51+ PR #4997
- Improved formatting of long slice patterns #4530
  - **Note you must have `version = Two` in your configuration to take advantage of the new formatting**
- Stabilized `match_block_trailing_comma` configuration option #3380 - https://rust-lang.github.io/rustfmt/?version=v1.4.38&search=#match_block_trailing_comma
- Stabilized `disable_all_formatting` configuration option #5026 - https://rust-lang.github.io/rustfmt/?version=v1.4.38&search=#disable_all_formatting
- Various improvements to the configuration documentation website https://rust-lang.github.io/rustfmt/?version=v1.4.38
- Addressed various clippy and rustc warnings

Fixed

- Resolved issue where specious whitespace would be inserted when a block style comment was terminated within string literal processing #4312
- Nested out-of-line mods are again parsed and formatted #4874
- Accepts `2021` for edition value from rustfmt command line PR #4847
- Unstable command line options are no longer displayed in `--help` text on stable PR #4798
- Stopped panicking on patterns in match arms which start with non-ascii characters #4868
- Stopped stripping defaults on const params #4816
- Fixed issue with dropped content with GAT aliases with self bounds in impls #4911
- Stopped removing generic args on associated type constraints #4943
- Stopped dropping visibility on certain trait and impl items #4960

- Fixed dropping of qualified paths in struct patterns #4908 and #5005
- Fixed bug in line width calculation that was causing specious formatting of certain patterns #4031
  - **Note that this bug fix may cause observable formatting changes in cases where code had been formatted with prior versions of rustfmt that contained the bug**
- Fixed bug where rustfmt would drop parameter attributes if they were too long in certain cases #4579
- Resolved idempotency issue with extern body elements #4963
- rustfmt will now handle doc-style comments on function parameters, since they could appear with certain macro usage patterns even though it's generally invalid syntax #4936
- Fixed bug in `match_block_trailing_comma` where commas were not added to the blocks of bodies whose arm had a guard that did not fit on the same line as the pattern #4998
- Fixed bug in cases where derive attributes started with a block style comment #4984
- Fixed issue where the struct rest could be lost when `struct_field_align_threshold` was enabled #4926
- Handles cases where certain control flow type expressions have comments between patterns/keywords and the pattern ident contains the keyword #5009
- Handles tuple structs that have explicit visibilities and start with a block style comment #5011
- Handles leading line-style comments in certain types of macro calls #4615

Added

- Granular width heuristic options made available for user control PR #4782. This includes the following:
  - `array_width`
  - `attr_fn_like_width`
  - `chain_width`
  - `fn_call_width`
  - `single_line_if_else_max_width`
  - `struct_lit_width`
  - `struct_variant_width`

Note this hit the rustup distributions prior to the v1.4.38 release as part of an out-of-cycle updates, but is listed in this version because the feature was not in the other v1.4.37 releases. See also the `use_small_heuristics` section on the configuration site for more information https://rust-lang.github.io/rustfmt/?version=v1.4.38&search=#use_small_heuristics

- New `One` variant added to `imports_granularity` configuration option which can be used to reformat all imports into a single use statement #4669

- rustfmt will now skip files that are annotated with `@generated` at the top of the file #3958 if `format_generated_files` option is set to `false` (by default `@generated` files are formatted)
- New configuration option `hex_literal_case` that allows user to control the casing utilized for hex literals PR #4903

See the section on the configuration site for more information https://rust-lang.github.io/rustfmt/?version=v1.4.38&search=#hex_literal_case

- `cargo fmt` now directly supports the `--check` flag, which means it's now possible to run `cargo fmt --check` instead of the more verbose `cargo fmt -- --check` #3888

Install/Download Options

- **rustup (nightly)** - *pending*
- **GitHub Release Binaries** - Release v1.4.38
- **Build from source** - Tag v1.4.38, see instructions for how to install rustfmt from source

[1.4.37] 2021-04-03

Changed

- `rustc-ap-*` crates updated to v712.0.0

Fixed

- Resolve idempotence issue related to indentation of macro defs that contain or-patterns with inner comments (#4603)
- Addressed various clippy and rustc warnings

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - *pending*
- **GitHub Release Binaries** - Release v1.4.37
- **Build from source** - Tag v1.4.37, see instructions for how to install rustfmt from source

[1.4.36] 2021-02-07

Changed

- `rustc-ap-*` crates updated to v705.0.0

Install/Download Options

- **crates.io package** - *pending*

- **rustup (nightly)** - *pending*
- **GitHub Release Binaries** - Release v1.4.36
- **Build from source** - Tag v1.4.36, see instructions for how to install rustfmt from source

[1.4.35] 2021-02-03

Changed

- `rustc-ap-*` crates updated to v702.0.0

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - *n/a (superseded by v1.4.36)
- **GitHub Release Binaries** - Release v1.4.35
- **Build from source** - Tag v1.4.35, see instructions for how to install rustfmt from source

[1.4.34] 2021-01-28

Fixed

- Don't insert trailing comma on (base-less) rest in struct literals within macros (#4675)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2021-01-31`
- **GitHub Release Binaries** - Release v1.4.34
- **Build from source** - Tag v1.4.34, see instructions for how to install rustfmt from source

[1.4.33] 2021-01-27

Changed

- `merge_imports` configuration has been deprecated in favor of the new `imports_granularity` option. Any existing usage of `merge_imports` will be automatically mapped to the corresponding value on `imports_granularity` with a warning message printed to encourage users to update their config files.

Added

- New `imports_granularity` option has been added which succeeds `merge_imports`. This new option supports several additional variants which allow users to merge imports at different levels (crate or module), and even flatten imports to have a single use statement per item. (PR #4634, PR #4639)

See the section on the configuration site for more information https://rust-lang.github.io/rustfmt/?version=v1.4.33&search=#imports_granularity

Fixed

- Fix erroneous removal of `const` keyword on const trait impl (#4084)
- Fix incorrect span usage wit const generics in supertraits (#4204)
- Use correct span for const generic params (#4263)
- Correct span on const generics to include type bounds (#4310)
- Idempotence issue on blocks containing only empty statements (#4627 and #3868)
- Fix issue with semicolon placement on required functions that have a trailing comment that ends in a line-style comment before the semicolon (#4646)
- Avoid shared interned cfg_if symbol since rustfmt can re-initialize the rustc_ast globals on multiple inputs (#4656)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - n/a (superseded by v1.4.34)
- **GitHub Release Binaries** - Release v1.4.33
- **Build from source** - Tag v1.4.33, see instructions for how to install rustfmt from source

[1.4.32] 2021-01-16

Fixed

- Indentation now correct on first bound in cases where the generic bounds are multiline formatted and the first bound itself is multiline formatted (#4636)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2021-01-18`
- **GitHub Release Binaries** - Release v1.4.32
- **Build from source** - Tag v1.4.32, see instructions for how to install rustfmt from source

[1.4.31] 2021-01-09

Changed

- `rustc-ap-*` crates updated to v697.0.0

Added

- Support for 2021 Edition #4618)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2021-01-16`
- **GitHub Release Binaries** - Release v1.4.31
- **Build from source** - Tag v1.4.31, see instructions for how to install rustfmt from source

[1.4.30] 2020-12-20

Fixed

- Last character in derive no longer erroneously stripped when `indent_style` is overridden to `Visual`. (#4584)
- Brace wrapping of closure bodies maintained in cases where the closure has an explicit return type and the body consists of a single expression statement. (#4577)
- No more panics on invalid code with `err` and `typeof` types (#4357, #4586)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2020-12-25`
- **GitHub Release Binaries** - Release v1.4.30
- **Build from source** - Tag v1.4.30, see instructions for how to install rustfmt from source

[1.4.29] 2020-12-04

Fixed

- Negative polarity on non-trait impl now preserved. (#4566)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2020-12-07`
- **GitHub Release Binaries** - Release v1.4.29

- **Build from source** - Tag v1.4.29, see instructions for how to install rustfmt from source

[1.4.28] 2020-11-29

Changed

- `rustc-ap-*` crates updated to v691.0.0
- In the event of an invalid inner attribute on a `cfg_if` condition, rustfmt will now attempt to continue and format the imported modules. Previously rustfmt would emit the parser error about an inner attribute being invalid in this position, but for rustfmt's purposes the invalid attribute doesn't prevent nor impact module formatting.

Added

- `group_imports` - a new configuration option that allows users to control the strategy used for grouping imports (#4107)

Fixed

- Formatting of malformed derived attributes is no longer butchered. (#3898, #4029, #4115, #4545)
- Correct indentation used in macro branches when `hard_tabs` is enabled. (#4152)
- Comments between the visibility modifier and item name are no longer dropped. (#2781)
- Comments preceding the assignment operator in type aliases are no longer dropped. (#4244)
- Comments between {`&` operator, lifetime, `mut` kw, type} are no longer dropped. (#4245)
- Comments between type bounds are no longer dropped. (#4243)
- Function headers are no longer dropped on foreign function items. (#4288)
- Foreign function blocks are no longer dropped. (#4313)
- `where_single_line` is no longer incorrectly applied to multiline function signatures that have no `where` clause. (#4547)
- `matches!` expressions with multiple patterns and a destructure pattern are now able to be formatted. (#4512)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - n/a (superseded by v1.4.29)
- **GitHub Release Binaries** - Release v1.4.28
- **Build from source** - Tag v1.4.28, see instructions for how to install rustfmt from source

[1.4.27] 2020-11-16

Fixed

- Leading comments in an extern block are no longer dropped (a bug that exists in v1.4.26). (#4528)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2020-11-18`
- **GitHub Release Binaries** - Release v1.4.27
- **Build from source** - Tag v1.4.27, see instructions for how to install rustfmt from source

[1.4.26] 2020-11-14

Changed

- Original comment indentation for trailing comments within an `if` is now taken into account when determining the indentation level to use for the trailing comment in formatted code. This does not modify any existing code formatted with rustfmt; it simply gives the programmer discretion to specify whether the comment is associated to the `else` block, or if the trailing comment is just a member of the `if` block. (#1575, #4120, #4506)

In this example the `// else comment` refers to the `else`:

```rust
// if comment
if cond {
    "if"
// else comment
} else {
    "else"
}
```

Whereas in this case the `// continue` comments are members of their respective blocks and do not refer to the `else` below.

```rust
if toks.eat_token(Token::Word("modify"))? && toks.eat_token(Token::Word("labels"))? {
    if toks.eat_token(Token::Colon)? {
        // ate the token
    } else if toks.eat_token(Token::Word("to"))? {
        // optionally eat the colon after to, e.g.:
        // @rustbot modify labels to: -S-waiting-on-author, +S-waiting-on-review
        toks.eat_token(Token::Colon)?;
    } else {
        // It's okay if there's no to or colon, we can just eat labels
        // afterwards.
    }
    1 + 2;
    // continue
} else if toks.eat_token(Token::Word("label"))? {
    // continue
} else {
    return Ok(None);
}
```

Fixed

- Formatting of empty blocks with attributes which only contained comments is no longer butchered.(#4475, #4467, #4452, #4522)
- Indentation of trailing comments in non-empty extern blocks is now correct. (#4120)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2020-11-16`
- **GitHub Release Binaries** - Release v1.4.26
- **Build from source** - Tag v1.4.26, see instructions for how to install rustfmt from source

[1.4.25] 2020-11-10

Changed

- Semicolons are no longer automatically inserted on trailing expressions in macro definition arms (#4507). This gives the programmer control and discretion over whether there should be semicolons in these scenarios so that potential expansion issues can be avoided.

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2020-11-14`
- **GitHub Release Binaries** - Release v1.4.25
- **Build from source** - Tag v1.4.25, see instructions for how to install rustfmt from source

[1.4.24] 2020-11-05

Changed

- Block wrapped match arm bodies containing a single macro call expression are no longer flattened (#4496). This allows programmer discretion so that the block wrapping can be preserved in cases where needed to prevent issues in expansion, such as with trailing semicolons, and aligns with updated Style Guide guidance for such scenarios.

Fixed

- Remove useless `deprecated` attribute on a trait impl block in the rustfmt lib, as these now trigger errors (rust-lang/rust/#78626)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - Starting in `2020-11-09`
- **GitHub Release Binaries** - Release v1.4.24
- **Build from source** - Tag v1.4.24, see instructions for how to install rustfmt from source

[1.4.23] 2020-10-30

Changed

- Update `rustc-ap-*` crates to v686.0.0

Added

- Initial support for formatting new ConstBlock syntax (#4478)

Fixed

- Handling of unclosed delimiter-only parsing errors in input files (#4466)
- Misc. minor parser bugs (#4418 and #4431)
- Panic on nested tuple access (#4355)
- Unable to disable license template path via cli override (#4487)
- Preserve comments in empty statements #4018)
- Indentation on skipped code #4398)

Install/Download Options

- **crates.io package** - *pending*
- **rustup (nightly)** - n/a (superseded by v1.4.24)
- **GitHub Release Binaries** - Release v1.4.23
- **Build from source** - Tag v1.4.23, see instructions for how to install rustfmt from source

[1.4.22] 2020-10-04

Changed

- Update `rustc-ap-*` crates to v679.0.0

- Add config option to allow control of leading match arm pipes
- Support `RUSTFMT` environment variable in `cargo fmt` to run specified `rustfmt` instance

Fixed

- Fix preservation of type aliases within extern blocks

[1.4.9] 2019-10-07

Changed

- Update `rustc-ap-*` crates to 606.0.0.

Fixed

- Fix aligning comments of different group
- Fix flattening imports with a single `self`.
- Fix removing attributes on function parameters.
- Fix removing `impl` keyword from opaque type.

[1.4.8] 2019-09-08

Changed

- Update `rustc-ap-*` crates to 583.0.0.

[1.4.7] 2019-09-06

Added

- Add `--config` command line option.

Changed

- Update `rustc-ap-*` crates to 581.0.0.
- rustfmt now do not warn against trailing whitespaces inside macro calls.

Fixed

- Fix `merge_imports` generating invalid code.
- Fix removing discriminant values on enum variants.
- Fix modules defined inside `cfg_if!` not being formatted.
- Fix minor formatting issues.

[1.4.6] 2019-08-28

Added

- Add `--message-format` command line option to `cargo-fmt`.

- Add `-l,--files-with-diff` command line option to `rustfmt`.
- Add `json` emit mode.

Fixed

- Fix removing attributes on struct pattern's fields.
- Fix non-idempotent formatting of match arm.
- Fix `merge_imports` generating invalid code.
- Fix imports with `#![macro_use]` getting reordered with `reorder_imports`.
- Fix calculation of line numbers in checkstyle output.
- Fix poor formatting of complex fn type.

[1.4.5] 2019-08-13

Fixed

- Fix generating invalid code when formatting an impl block with const generics inside a where clause.
- Fix adding a trailing space after a `dyn` keyword which is used as a macro argument by itself.

[1.4.4] 2019-08-06

Fixed

- Fix `cargo fmt` incorrectly formatting crates that is not part of the workspace or the path dependencies.
- Fix removing a trailing comma from a tuple pattern.

[1.4.3] 2019-08-02

Changed

- Update `rustc-ap-*` crates to 546.0.0.

Fixed

- Fix an underscore pattern getting removed.

[1.4.2] 2019-07-31

Changed

- Explicitly require the version of `rustfmt-config_proc_macro` to be 0.1.2 or later.

[1.4.1] 2019-07-30

Changed

- Update `rustc-ap-*` crates to 542.0.0.

[1.4.0] 2019-07-29

Added

- Add new attribute `rustfmt::skip::attributes` to prevent rustfmt from formatting an attribute #3665

Changed

- Update `rustc-ap-*` crates to 541.0.0.
- Remove multiple semicolons.

[1.3.3] 2019-07-15

Added

- Add `--manifest-path` support to `cargo fmt` (#3683).

Fixed

- Fix `cargo fmt -- --help` printing nothing (#3620).
- Fix inserting an extra comma (#3677).
- Fix incorrect handling of CRLF with `file-lines` (#3684).
- Fix `print-config=minimal` option (#3687).

[1.3.2] 2019-07-06

Fixed

- Fix rustfmt crashing when `await!` macro call is used in a method chain.
- Fix rustfmt not recognizing a package whose name differs from its directory's name.

[1.3.1] 2019-06-30

Added

- Implement the `Display` trait on the types of `Config`.

Changed

- `ignore` configuration option now only supports paths separated by `/`. Windows-style paths are not supported.
- Running `cargo fmt` in a sub-directory of a project is now supported.

Fixed

- Fix bugs that may cause rustfmt to crash.

[1.3.0] 2019-06-09

Added

- Format modules defined inside `cfg_if` macro calls #3600

Changed

- Change option `format_doc_comment` to `format_code_in_doc_comment`.
- `use_small_heuristics` changed to be an enum and stabilised. Configuration options are now ready for 1.0.
- Stabilise `fn_args_density` configuration option and rename it to `fn_args_layout` #3581
- Update `rustc-ap-*` crates to 486.0.0
- Ignore sub-modules when skip-children is used #3607
- Removed bitrig support #3608

Fixed

- `wrap_comments` should not imply `format_doc_comments` #3535
- Incorrect handling of const generics #3555
- Add the handling for `vec!` with paren inside macro #3576
- Format trait aliases with where clauses #3586
- Catch panics from the parser while rewriting macro calls #3589
- Fix erasing inner attributes in struct #3593
- Inline the attribute with its item even with the `macro_use` attribute or when `reorder_imports` is disabled #3598
- Fix the bug add unwanted code to impl #3602

[1.2.2] 2019-04-24

Fixed

- Fix processing of `ignore` paths #3522
- Attempt to format attributes if only they exist #3523

[1.2.1] 2019-04-18

Added

- Add `--print-config current` CLI option b473e65
- Create GitHub page for Configuration.md #3485

Fixed

- Keep comment appearing between parameter's name and its type #3491
- Do not delete semicolon after macro call with square brackets #3500
- Fix `--version` CLI option #3506
- Fix duplication of attributes on a match arm's body #3510
- Avoid overflowing item with attributes #3511

[1.2.0] 2019-03-27

Added

- Add new attribute `rustfmt::skip::macros` to prevent rustfmt from formatting a macro #3454

Changed

- Discard error report in silent_emitter #3466

Fixed

- Fix bad performance on deeply nested binary expressions #3467
- Use BTreeMap to guarantee consistent ordering b4d4b57

[1.1.1] 2019-03-21

Fixed

- Avoid panic on macro inside deeply nested block c9479de
- Fix line numbering in missed spans and handle file_lines in edge cases cdd08da
- Fix formatting of async blocks 1fa06ec
- Avoid duplication on the presence of spaces between macro name and `!` #3464

[1.1.0] 2019-03-17

Added

- Add `inline_attribute_width` configuration option to write an item and its attribute on the same line if their combined width is below a threshold #3409
- Support `const` generics f0c861b
- Support path clarity module #3448

Changed

- Align loop and while formatting 7d9a2ef
- Support `EmitMode::ModifiedLines` with stdin input #3424

- Update `rustc-ap-*` crates to 407.0.0
- Remove trailing whitespaces in missing spans 2d5bc69

Fixed

- Do not remove comment in the case of no arg 8e3ef3e
- Fix `Ident of macro+ident gets duplicated` error 40ff078
- Format the if expression at the end of the block in a single line 5f3dfe6

[1.0.3] 2019-02-14

Added

- Point unstable options to tracking issues 412dcc7

Changed

- Update `rustc-ap-*` crates to 373.0.0

[1.0.2] 2019-02-12

Added

- Add a section to the Contributing.md file about version-gating formatting changes 36e2cb0
- Allow specifying package with `-p` CLI option a8d2591
- Support `rustfmt::skip` on imports #3289
- Support global `rustfmt.toml` to be written in user config directory #3280
- Format visibility on trait alias 96a3df3

Changed

- Do not modify original source code inside macro call #3260
- Recognize strings inside comments in order to avoid indenting them baa62c6
- Use Unicode-standard char width to wrap comments or strings a01990c
- Change new line point in the case of no args #3294
- Use the same formatting rule between functions and macros #3298
- Update rustc-ap-rustc_target to 366.0.0, rustc-ap-syntax to 366.0.0, and rustc-ap-syntax_pos to 366.0.0

Fixed

- rewrite_comment: fix block fallback when failing to rewrite an itemized block ab7f4e1
- Catch possible tokenizer panics #3240
- Fix macro indentation on Windows #3266
- Fix shape when formatting return or break expr on statement position #3259

- rewrite_comment: fix block fallback when failing to rewrite an itemized block
- Keep leading double-colon to respect the 2018 edition of rust's paths a2bfc02
- Fix glob and nested global imports 2125ad2
- Do not force trailing comma when using mixed layout #3306
- Prioritize `single_line_fn` and `empty_item_single_line` over `brace_style` #3308
- Fix `internal error: left behind trailing whitespace` with long lines c2534f5
- Fix attribute duplication #3325
- Fix formatting of strings within a macro 813aa79
- Handle a macro argument with a single keyword 9a7ea6a

[1.0.1] 2018-12-09

Added

- Add a `version` option 378994b

Changed

- End expressions like return/continue/break with a semicolon #3223
- Update rustc-ap-rustc_target to 306.0.0, rustc-ap-syntax to 306.0.0, and rustc-ap-syntax_pos to 306.0.0

Fixed

- Allow to run a rustfmt command from cargo-fmt even when there is no target a2da636
- Fix `un-closed delimiter` errors when formatting break labels 40174e9

[1.0.0] 2018-11-19

Changed

- Preserve possibly one whitespace for brace macros 1a3bc79
- Prefer to break arguments over putting output type on the next line 1dd54e6

[0.99.9] 2018-11-15

Changed

- Update rustc-ap-rustc_target to 297.0.0, rustc-ap-syntax to 297.0.0, to rustc-ap-syntax_pos to 297.0.0
- Don't align comments on `extern crate`s dd7add7

[0.99.8] 2018-11-14

Added

- Add `overflow_delimited_expr` config option to more aggressively allow overflow #3175

Fixed

- Fix the logic for retaining a comment before the arrow in a match #3181
- Do not wrap comments in doctest to avoid failing doctest runs #3183
- Fix comment rewriting that was wrapping code into a line comment #3188
- Fix formatting of unit-struct with `where`-clause #3200

[0.99.7] 2018-11-07

Changed

- Force a newline after the `if` condition if there is a different indentation level #3109
- Use correct width when formatting type on local statement #3126
- Treat crates non-alphabetically when ordering 799005f
- Fix formatting of code that is annotated with rustfmt::skip #3113
- Stabilize `edition` configuration option 9c3ae2d
- cargo-fmt: detect Rust edition in use #3129
- Trim the indentation on macros which heuristically appear to use block-style indentation #3178

Fixed

- Do not remove path disambiugator inside macro #3142
- Improve handling of Windows newlines #3141
- Fix alignment of a struct's fields (`struct_field_align_threshold` option) with the Visual `indent_style` #3165
- Fix a bug in formatting markdown lists within comments #3172

[0.99.6] 2018-10-18

Added

- Add `enum_discrim_align_threshold` option to vertically align enum discriminants cc22869
- Add `println!`-like heuristic to the `fail` attribute #3067
- Handle itemized items inside comments #3083
- Add `format_doc_comments` configuration option to control the formatting of code snippets inside comments #3089

Changed

- Makes brace behavior consistent with empty bodies for traits and impls 2727d41
- Consider a multi-lined array as a block-like expression #3969
- Improve formatting of strings #3073
- Get rid of extra commas in Visual struct literal formatting #3077
- Update rustc-ap-rustc_target to 274.0.0, rustc-ap-syntax to 274.0.0, and rustc-ap-syntax_pos to 274.0.0
- Format macro calls with item-like arguments #3080
- Avoid control flow expressions conditions to go multi line ef59b34
- Simplify multi-lining binop expressions #3101

Fixed

- Do not format a code block in documentation if it is annotated with ignore or text 2bcc3a9
- Fix inconsistent overflow behavior in Visual style #3078
- Fix corner cases of the string formatting implementation #3083
- Do not add parens around lifetimes 0ac68c9
- Catch parser panic in format_snippet 8c4e92a

[0.99.5] 2018-09-25

Added

- Handle leading module separator for 2018 Edition #2952
- Add configuration option `normalize_doc_attributes`: convert doc attributes to comments #3002

Changed

- Accept 2015 and 2018 instead of Edition2015 and Edition2018 for edition option eec7436
- Support platforms without a timer 46e2a2e
- Update rustc-ap-rustc_target to 263.0.0, rustc-ap-syntax to 263.0.0, and rustc-ap-syntax_pos to 263.0.0

Fixed

- Format of attributes with commas #2971
- Fix optional arg condensing #2972
- Improve formatting of long function parameters #2981
- Fix formatting of raw string literals #2983
- Handle chain with try operators with spaces #2986
- Use correct shape in Visual tuple rewriting #2987
- Impove formatting of arguments with `visual_style = "Visual"` option #2988

- Change `print_diff` to output the correct line number 992b179
- Propagate errors about failing to rewrite a macro 6f318e3
- Handle formatting of long function signature #3010
- Fix indent computation of a macro with braces c3edf6d
- Format generics on associated types #3035
- Incorrect indentation of multiline block match expression #3042
- Fix bug in import where two consecutive module separators were possible 98a0ef2
- Prevent right-shifting of block comments with bare lines 5fdb6db

[0.99.4] 2018-08-27

Added

- Handle formatting of underscore imports #2951
- Handle formatting of try blocks #2965

Changed

- Update rustc-ap-rustc_target to 237.0.0, rustc-ap-syntax to 237.0.0, and rustc-ap-syntax_pos to 237.0.0 ca19c9a
- Consider `dev` channel as nightly for unstable features #2948

Fixed

- Fix formatting of patterns with ellipsis # 2942

[0.99.3] 2018-08-23

Added

- Use path attribute when searching for modules #2901
- Expose FileLines JSON representation to allow external libraries to use the file_lines option #2915

Changed

- Replace '–conifig-help' with '–config=help' cb10e06
- Improve formatting of slice patterns #2912

Fixed

- Format chains with comment #2899
- Fix indentation of formatted macro body #2920
- Fix indentation of block comments f23e6aa

[0.99.2] 2018-08-07

Changed

- Update rustc-ap-rustc_target to 218.0.0, rustc-ap-syntax to 218.0.0, and rustc-ap-syntax_pos to 218.0.0 5c9a2b6
- Combine function-like attributes #2900

Fixed

- Explicitly handle semicolon after the item in statement position d96e3ca
- Fix parsing '#'-hiding of rustdoc 2eca09e

[0.99.1] 2018-08-04

Fixed

- fix use statements ordering when a number is present 1928ae7

[0.99.0] 2018-08-03

- 1.0 RC release

Changed

- Clarification in README.md 30fe66b

[0.9.0] 2018-08-01

Added

- Handle raw identifiers 3027c21
- Format async closure 60ce411
- Add max_width option for all heuristics c2ae39e
- Add config option `format_macro_matchers` to format the metavariable matching patterns in macros 79c5ee8
- Add config option `format_macro_bodies` to format the bodies of macros 79c5ee8
- Format exitential type fc307ff
- Support raw identifiers in struct expressions f121b1a
- Format Async block and async function 0b25f60

Changed

- Update rustc-ap-rustc_target to 211.0.0, rustc-ap-syntax to 211.0.0, and rustc-ap-syntax_pos to 211.0.0
- Put each nested import on its own line while putting non-nested imports on the same line as much as possible 42ab258

- Respect `empty_item_single_line` config option when formatting empty impls. Put the `where` on its own line to improve readability #2771
- Strip leading `|` in match arm patterns 1d4b988
- Apply short function call heuristic to attributes 3abebf9
- Indent a match guard if the pattern is multiline be4d37d
- Change default newline style to `Native` 9d8f381
- Improve formatting of series of binop expressions a4cdb68
- Trigger an internal error if we skip formatting due to a lost comment b085113
- Refactor chain formatting #2838

Fixed

- Do not insert spaces around braces with empty body or multiple lines 2f65852
- Allow using mixed layout with comments #2766
- Handle break labels #2726
- fix rewrite_string when a line feed is present 472a2ed
- Fix an anomaly with comments and array literals b28a0cd
- Check for comments after the `=>` in a match arm 6899471

[0.8.0,0.8.1,0.8.2] 2018-05-28

Added

- Use scoped attributes for skip attribute https://github.com/rust-lang/rustfmt/pull/2703

Changed

- Comment options `wrap_comments` and `normalize_comments` are reverted back to unstable 416bc4c
- Stabilise `reorder_imports` and `reorder_modules` options 7b6d2b4
- Remove `spaces_within_parens_and_brackets` option d726492
- Stabilise shorthand options: `use_try_shorthand`, `use_field_init_shorthand`, and `force_explicit_abi` 8afe367
- Stabilise `remove_nested_parens` and set default to true a70f716
- Unstabilise `unstable_features` dd9c15a
- Remove `remove_blank_lines_at_start_or_end_of_block` option 2ee8b0e
- Update rustc-ap-syntax to 146.0.0 and rustc-ap-rustc_target to 146.0.0 2c275a2
- Audit the public API #2639

Fixed

- Handle code block in doc comment without rust prefix f1974e2

[0.7.0] 2018-05-14

Added

- Add integration tests against crates in the rust-lang-nursery c79f39a

Changed

- Update rustc-ap-syntax to 128.0.0 and ustc-ap-rustc_target to 128.0.0 195395f
- Put operands on its own line when each fits in a single line f8439ce
- Improve CLI options 55ac062 1869888 798bffb 4d9de48 eca7796 8396da1 5d9f5aa

Fixed

- Use correct line width for list attribute 61a401a
- Avoid flip-flopping impl items when reordering them 37c216c
- Formatting breaks short lines when max_width is less than 100 9b36156
- Fix variant “Mixed” of imports_layout option 8c8676c
- Improve handling of long lines f885039
- Fix up lines exceeding max width 51c07f4
- Fix handling of modules in non_modrs_mods style cf573e8
- Do not duplicate attributes on use items e59ceaf
- Do not insert an extra brace in macros with native newlines 4c9ef93

[0.6.1] 2018-05-01

Changed

- Change the default value of imports_indent to IndentStyle::Block https://github.com/rust-lang/rustfmt/pull/2662

Fixed

- Handle formatting of auto traits 5b5a72c
- Use consistent formatting for empty enum and struct https://github.com/rust-lang/rustfmt/pull/2656

[0.6.0] 2018-04-20

Changed

- Improve public API 8669004

[0.5.0] 2018-04-20

Added

- Add `verbose-diff` CLI option 5194984

Changed

- Update rustc-ap-syntax to 103.0.0 dd807e2
- Refactor to make a sensible public API ca610d3

Fixed

- Add spaces between consecutive `..` `..=` 61d29eb

[0.4.2] 2018-04-12

Added

- Handle binary operators and lifetimes 0fd174d
- Add reorder_impl_items config option 94f5a05
- Add `--unstable-features` CLI option to list unstable options from the `--help` output 8208f8a
- Add merge_imports config option 5dd203e

Changed

- Format macro arguments with vertical layout ec71459
- Reorder imports by default 164cf7d
- Do not collapse block around expr with condition on match arm 5b9b7d5
- Use vertical layout for complex attributes c77708f
- Format array using heuristics for function calls 98c6f7b
- Implement stable ordering for impl items with the the following item priority: type, const, macro, then method fa80ddf
- Reorder imports by default 164cf7d
- Group `extern crate` by default 3a138a2
- Make `error_on_line_overflow` false by default f146711
- Merge imports with the same prefix into a single nested import 1954513
- Squash the various 'reorder imports' option into one 911395a

Fixed

- Print version is missing the channel ca6fc67
- Do not add the beginning vert to the match arm 1e1d9d4
- Follow indent style config when formatting attributes efd295a
- Do not insert newline when item is empty a8022f3
- Do not indent or unindent inside string literal ec1907b

[0.4.1] 2018-03-16

Added

- Add `ignore` configuration option.
- Add `license_template_path` configuration option.

- Format `lazy_static!`.

Fixed

- Fix formatting bugs.
- Fix setting `reorder_modules` removing inline modules.
- Format attributes on block expressions.
- Support `dyn trait` syntax.
- Support multiple patterns in `if let` and `while let`.
- Support a pattern with parentheses.

[0.4.0] 2018-03-02

Changed

- Do not print verbose outputs when formatting with stdin.
- Preserve trailing whitespaces in doc comments.
- Scale the values of width heuristics by `max_width`.

Fixed

- Do not reorder items with `#[macro_use]`.
- Fix formatting bugs.
- Support the beginning `|` on a match arm.

[0.3.8] 2018-02-04

Added

- Format (or at least try to format) `macro_rules!`.

[0.3.7] 2018-02-01

Added

- Add `use_field_init_shorthand` config option.
- Add `reorder_modules` configuration option.

[0.3.6] 2018-01-18

Fixed

- Fix panicking on formatting certain macros (#2371).

[0.3.5] 2018-01-15

Changed

- Format code block in comments when `wrap_comments` is set to `true`.

- Remove `same_line_attributes` configuration option.
- Rename `git-fmt` to `git-rustfmt`.

Fixed

- Rustup to `rustc 1.25.0-nightly (e6072a7b3 2018-01-13)`.
- Fix formatting bugs.

[0.3.4] 2017-12-23

Added

- Add `--version` flag to `cargo-fmt`, allow `cargo fmt --version`.

Fixed

- Rustup to `rustc 1.24.0-nightly (5165ee9e2 2017-12-22)`.

[0.3.3] 2017-12-22

Added

- Format trait aliases.

Changed

- `cargo fmt` will format every workspace member.

Fixed

- Rustup to `rustc 1.24.0-nightly (250b49205 2017-12-21)`
- Fix formatting bugs.

[0.3.2] 2017-12-15

Changed

- Warn when unknown configuration option is used.

Fixed

- Rustup to `rustc 1.24.0-nightly (0077d128d 2017-12-14)`.

[0.3.1] 2017-12-11

Added

- Add `error_on_unformatted` configuration option.
- Add `--error-on-unformatted` command line option.

Changed

- Do not report formatting errors on comments or strings by default.
- Rename `error_on_line_overflow_comments` to `error_on_unformatted`.

Fixed

- Fix formatting bugs.
- Fix adding a trailing whitespace inside code block when `wrap_comments = true`.

[0.3.0] 2017-12-11

Added

- Support nested imports.

Changed

- Do not report errors on skipped items.
- Do not format code block inside comments when `wrap_comments = true`.
- Keep vertical spaces between items within range.
- Format `format!` and its variants using compressed style.
- Format `write!` and its variants using compressed style.
- Format **simple** array using compressed style.

Fixed

- Fix `rustfmt --package package_name` not working properly.
- Fix formatting bugs.

[0.2.17] 2017-12-03

Added

- Add `blank_lines_lower_bound` and `blank_lines_upper_bound` configuration options.

Changed

- Combine configuration options related to width heuristic into `width_heuristic`.
- If the match arm's body is `if` expression, force to use block.

Fixed

- Fix `cargo fmt --all` being trapped in an infinite loop.
- Fix many formatting bugs.

Removed

- Remove legacy configuration options.

[0.2.16] 2017-11-21

Added

- Remove empty lines at the beginning of the file.
- Soft wrapping on doc comments.

Changed

- Break before `|` when using multiple lines for match arm patterns.
- Combine `control_style`, `where_style` and `*_indent` config options into `indent_style`.
- Combine `item_brace_style` and `fn_brace_style` config options into `brace_style`.
- Combine config options related spacing around colons into `space_before_colon` and `space_after_colon`.

Fixed

- Fix many bugs.

[0.2.15] 2017-11-08

Added

- Add git-fmt tool
- `where_single_line` configuration option.

Changed

- Rename `chain_one_line_max` to `chain_width`.
- Change the suffix of indent-related configuration options to `_indent`.

[0.2.14] 2017-11-06

Fixed

- Rustup to the latest nightly.

[0.2.13] 2017-10-30

Fixed

- Rustup to the latest nightly.

[0.2.12] 2017-10-29

Fixed

- Fix a bug that `cargo fmt` hangs forever.

[0.2.11] 2017-10-29

Fixed

- Fix a bug that `cargo fmt` crashes.

[0.2.10] 2017-10-28

[0.2.9] 2017-10-16

[0.2.8] 2017-09-28

[0.2.7] 2017-09-21

Added

- `binop_separator` configuration option (#1964).

Changed

- Use horizontal layout for function call with a single argument.

Fixed

- Fix panicking when calling `cargo fmt --all` (#1963).
- Refactorings & faster rustfmt.

[0.2.6] 2017-09-14

Fixed

- Fix a performance issue with nested block (#1940).
- Refactorings & faster rustfmt.

[0.2.5] 2017-08-31

Added

- Format and preserve attributes on statements (#1933).

Fixed

- Use getters to access `Span` fields (#1899).

[0.2.4] 2017-08-30

Added

- Add support for `Yield` (#1928).

[0.2.3] 2017-08-30

Added

- `multiline_closure_forces_block` configuration option (#1898).
- `multiline_match_arm_forces_block` configuration option (#1898).
- `merge_derives` configuration option (#1910).
- `struct_remove_empty_braces` configuration option (#1930).
- Various refactorings.

Changed

- Put single-lined block comments on the same line with list-like structure's item (#1923).
- Preserve blank line between doc comment and attribute (#1925).
- Put the opening and the closing braces of enum and struct on the same line, even when `item_brace_style = "AlwaysNextLine"` (#1930).

Fixed

- Format attributes on `ast::ForeignItem` and take max width into account (#1916).
- Ignore empty lines when calculating the shortest indent width inside macro with braces (#1918).
- Handle tabs properly inside macro with braces (#1918).
- Fix a typo in `compute_budgets_for_args()` (#1924).
- Recover comment between keyword (`impl` and `trait`) and `{` which used to get removed (#1925).