

Changelog

All notable changes to this project will be documented in this file. See [Changelog Update](#) if you want to update this document.

Unreleased / In Rust Nightly

[0eff589...master](#)

Rust 1.59 (beta)

Current beta, release 2022-02-24

[e181011...0eff589](#)

New Lints

- [index_refutable_slice](#) [#7643](#)
- [needless_splitn](#) [#7896](#)
- [unnecessary_to_owned](#) [#7978](#)
- [needless_late_init](#) [#7995](#)
- [octal_escapes](#) [#8007](#)
- [return_self_not_must_use](#) [#8071](#)
- [init_numbered_fields](#) [#8170](#)

Moves and Deprecations

- Move `if_then_panic` to `pedantic` and rename to [manual_assert](#) (now allow-by-default) [#7810](#)
- Rename `disallow_type` to [disallowed_types](#) and `disallowed_method` to [disallowed_methods](#) [#7984](#)
- Move [map_flatten](#) to `complexity` (now warn-by-default) [#8054](#)

Enhancements

- [match_overlapping_arm](#) : Fix false negative where after included ranges, overlapping ranges weren't linted anymore [#7909](#)
- [deprecated_cfg_attr](#) : Now takes the specified MSRV into account [#7944](#)
- [cast_lossless](#) : Now also lints for `bool` to integer casts [#7948](#)
- [let_underscore_lock](#) : Also emit lints for the `parking_lot` crate [#7957](#)
- [needless_borrow](#) [#7977](#)
 - Lint when a borrow is auto-dereffed more than once
 - Lint in the trailing expression of a block for a match arm
- [strlen_on_c_strings](#) [8001](#)
 - Lint when used without a fully-qualified path
 - Suggest removing the surrounding unsafe block when possible
- [non_ascii_literal](#) : Now also lints on `char s`, not just `string s` [#8034](#)
- [single_char_pattern](#) : Now also lints on `split_inclusive`, `split_once`, `rsplit_once`, `replace`, and `replacen` [#8077](#)
- [unwrap_or_else_default](#) : Now also lints on `std` constructors like `Vec::new`, `HashSet::new`, and `HashMap::new` [#8163](#)

- [shadow_reuse](#) : Now also lints on shadowed `if let` bindings, instead of [shadow_unrelated](#) [#8165](#)

False Positive Fixes

- [or_fun_call](#) , [unnecessary_lazy_evaluations](#) : Improve heuristics, so that cheap functions (e.g. calling `.len()` on a `Vec`) won't get linted anymore [#7639](#)
- [manual_split_once](#) : No longer suggests code changing the original behavior [#7896](#)
- Don't show [no_effect](#) or [unnecessary_operation](#) warning for unit struct implementing `FnOnce` [#7898](#)
- [semicolon_if_nothing_returned](#) : Fixed a bug, where the lint wrongly triggered on `let-else` statements [#7955](#)
- [if_then_some_else_none](#) : No longer lints if there is an early return [#7980](#)
- [needless_collect](#) : No longer suggests removal of `collect` when removal would create code requiring mutably borrowing a value multiple times [#7982](#)
- [shadow_same](#) : Fix false positive for `async` function's params [#7997](#)
- [suboptimal_flops](#) : No longer triggers in constant functions [#8009](#)
- [type_complexity](#) : No longer lints on associated types in traits [#8030](#)
- [question_mark](#) : No longer lints if returned object is not local [#8080](#)
- [option_if_let_else](#) : No longer lint on complex sub-patterns [#8086](#)
- [blocks_in_if_conditions](#) : No longer lints on empty closures [#8100](#)
- [enum_variant_names](#) : No longer lint when first prefix is only a substring of a camel-case word [#8127](#)
- [identity_op](#) : Only lint on integral operands [#8183](#)

Suggestion Fixes/Improvements

- [search_is_some](#) : Fix suggestion for `any()` not taking item by reference [#7463](#)
- [almost_swapped](#) : Now detects if there is a `no_std` or `no_core` attribute and adapts the suggestion accordingly [#7877](#)
- [redundant_pattern_matching](#) : Fix suggestion for deref expressions [#7949](#)
- [explicit_counter_loop](#) : Now also produces a suggestion for non-`usize` types [#7950](#)
- [manual_map](#) : Fix suggestion when used with unsafe functions and blocks [#7968](#)
- [option_map_or_none](#) : Suggest `map` over `and_then` when possible [#7971](#)
- [option_if_let_else](#) : No longer expands macros in the suggestion [#7974](#)
- [iter_cloned_collect](#) : Suggest `copied` over `cloned` when possible [#8006](#)
- [doc_markdown](#) : No longer uses inline hints to improve readability of suggestion [#8011](#)
- [needless_question_mark](#) : Now better explains the suggestion [#8028](#)
- [single_char_pattern](#) : Escape backslash `\` in suggestion [#8067](#)
- [needless_bool](#) : Suggest `a != b` over `!(a == b)` [#8117](#)
- [iter_skip_next](#) : Suggest to add a `mut` if it is necessary in order to apply this lints suggestion [#8133](#)
- [neg_multiply](#) : Now produces a suggestion [#8144](#)
- [needless_return](#) : Now suggests the unit type `()` over an empty block `{}` in match arms [#8185](#)
- [suboptimal_flops](#) : Now gives a syntactically correct suggestion for `to_radians` and `to_degrees` [#8187](#)

ICE Fixes

- [undocumented_unsafe_blocks](#) [#7945](#) [#7988](#)
- [unnecessary_cast](#) [#8167](#)

Documentation Improvements

- `print_stdout`, `print_stderr`, `dbg_macro`: Document how the lint level can be changed crate-wide [#8040](#)
- Added a note to the `README` that config changes don't apply to already compiled code [#8175](#)

Others

- [Clippy's lint list](#) now displays the version a lint was added. :tada: [#7813](#)
- New and improved issue templates [#8032](#)
- Dev: Add `cargo dev lint` command, to run your modified Clippy version on a file [#7917](#)

Rust 1.58

Current stable, released 2022-01-13

[00e31fa...e181011](#)

Rust 1.58.1

- Move `non_send_fields_in_send_ty` to nursery (now allow-by-default) [#8075](#)
- `useless_format`: Handle implicit named arguments [#8295](#)

New lints

- `transmute_num_to_bytes` [#7805](#)
- `match_str_case_mismatch` [#7806](#)
- `format_in_format_args`, `to_string_in_format_args` [#7743](#)
- `uninit_vec` [#7682](#)
- `fn_to_numeric_cast_any` [#7705](#)
- `undocumented_unsafe_blocks` [#7748](#)
- `trailing_empty_array` [#7838](#)
- `string_slice` [#7878](#)

Moves or deprecations of lints

- Move `non_send_fields_in_send_ty` to suspicious [#7874](#)
- Move `non_ascii_literal` to restriction [#7907](#)

Changes that expand what code existing lints cover

- `question_mark` now covers `Result` [#7840](#)
- Make `useless_format` recognize bare `format!("{}",)` [#7801](#)
- Lint on underscored variables with no side effects in `no_effect` [#7775](#)
- Expand `match_ref_pats` to check for multiple reference patterns [#7800](#)

False positive fixes

- Fix false positive of `implicit_saturating_sub` with `else` clause [#7832](#)
- Fix `question_mark` when there is call in conditional predicate [#7860](#)
- `mut_mut` no longer lints when type is defined in external macros [#7795](#)
- Avoid `eq_op` in test functions [#7811](#)
- `cast_possible_truncation` no longer lints when cast is coming from `signum` method call [#7850](#)
- `match_str_case_mismatch` no longer lints on uncased characters [#7865](#)
- `ptr_arg` no longer lints references to type aliases [#7890](#)

- `missing_safety_doc` now also accepts "implementation safety" headers [#7856](#)
- `missing_safety_doc` no longer lints if any parent has `#[doc(hidden)]` attribute [#7849](#)
- `if_not_else` now ignores else-if statements [#7895](#)
- Avoid linting `cast_possible_truncation` on bit-reducing operations [#7819](#)
- Avoid linting `field_reassign_with_default` when `Drop` and `Copy` are involved [#7794](#)
- `unnecessary_sort_by` now checks if argument implements `Ord` trait [#7824](#)
- Fix false positive in `match_overlapping_arm` [#7847](#)
- Prevent `needless_lifetimes` false positive in `async` function definition [#7901](#)

Suggestion fixes/improvements

- Keep an initial `::` when `doc_markdown` suggests to use ticks [#7916](#)
- Add a machine applicable suggestion for the `doc_markdown` missing backticks lint [#7904](#)
- `equatable_if_let` no longer expands macros in the suggestion [#7788](#)
- Make `shadow_reuse` suggestion less verbose [#7782](#)

ICE fixes

- Fix ICE in `enum_variant_names` [#7873](#)
- Fix ICE in `undocumented_unsafe_blocks` [#7891](#)

Documentation improvements

- Fixed naive doc formatting for `#[must_use]` lints (`must_use_unit` , `double_must_use` , `must_use_candidate` , `let_underscore_must_use`) [#7827](#)
- Fix typo in example for `match_result_ok` [#7815](#)

Others

- Allow giving reasons for `disallowed_types` [#7791](#)
- Fix `manual_assert` and `match_wild_err_arm` for `#![no_std]` and Rust 2021. [#7851](#)
- Fix regression in `semicolon_if_nothing_returned` on macros containing while loops [#7789](#)
- Added a new configuration `literal-suffix-style` to enforce a certain style writing `unseparated_literal_suffix` [#7726](#)

Rust 1.57

Released 2021-12-02

[7bfc26e...00e31fa](#)

New Lints

- `negative_feature_names` [#7539](#)
- `redundant_feature_names` [#7539](#)
- `mod_module_files` [#7543](#)
- `self_named_module_files` [#7543](#)
- `manual_split_once` [#7565](#)
- `derivable_impls` [#7570](#)
- `needless_option_as_deref` [#7596](#)
- `iter_not_returning_iterator` [#7610](#)
- `same_name_method` [#7653](#)

- [manual_assert](#) [#7669](#)
- [non_send_fields_in_send_ty](#) [#7709](#)
- [equatable_if_let](#) [#7762](#)

Moves and Deprecations

- Move [shadow_unrelated](#) to restriction [#7338](#)
- Move [option_if_let_else](#) to nursery [#7568](#)
- Move [branches_sharing_code](#) to nursery [#7595](#)
- Rename `if_let_some_result` to [match_result_ok](#) which now also handles `while let` cases [#7608](#)
- Move [many_single_char_names](#) to pedantic [#7671](#)
- Move [float_cmp](#) to pedantic [#7692](#)
- Rename `box_vec` to [box_collection](#) and lint on more general cases [#7693](#)
- Uplift `invalid_atomic_ordering` to rustc [rust-lang/rust#84039](#)

Enhancements

- Rewrite the `shadow*` lints, so that they find a lot more shadows and are not limited to certain patterns [#7338](#)
- The `avoid-breaking-exported-api` configuration now also works for [box_collection](#), [redundant_allocation](#), [rc_buffer](#), [vec_box](#), [option_option](#), [linkedlist](#), [rc_mutex](#) [#7560](#)
- [unnecessary_unwrap](#): Now also checks for `expect s` [#7584](#)
- [disallowed_methods](#): Allow adding a reason that will be displayed with the lint message [#7621](#)
- [approx_constant](#): Now checks the MSRV for `LOG10_2` and `LOG2_10` [#7629](#)
- [approx_constant](#): Add `TAU` [#7642](#)
- [needless_borrow](#): Now also lints on needless mutable borrows [#7657](#)
- [missing_safety_doc](#): Now also lints on unsafe traits [#7734](#)

False Positive Fixes

- [manual_map](#): No longer lints when the option is borrowed in the match and also consumed in the arm [#7531](#)
- [filter_next](#): No longer lints if `filter` method is not the `Iterator::filter` method [#7562](#)
- [manual_flatten](#): No longer lints if expression is used after `if let` [#7566](#)
- [option_if_let_else](#): Multiple fixes [#7573](#)
 - `break` and `continue` statements local to the would-be closure are allowed
 - Don't lint in const contexts
 - Don't lint when yield expressions are used
 - Don't lint when the captures made by the would-be closure conflict with the other branch
 - Don't lint when a field of a local is used when the type could be potentially moved from
 - In some cases, don't lint when scrutinee expression conflicts with the captures of the would-be closure
- [redundant_allocation](#): No longer lints on `Box<Box<dyn T>>` which replaces wide pointers with thin pointers [#7592](#)
- [bool_assert_comparison](#): No longer lints on types that do not implement the `Not` trait with `Output = bool` [#7605](#)
- [mut_range_bound](#): No longer lints on range bound mutations, that are immediately followed by a `break;` [#7607](#)

- [mutable_key_type](#) : Improve accuracy and document remaining false positives and false negatives [#7640](#)
- [redundant_closure](#) : Rewrite the lint to fix various false positives and false negatives [#7661](#)
- [large_enum_variant](#) : No longer wrongly identifies the second largest variant [#7677](#)
- [needless_return](#) : No longer lints on let-else expressions [#7685](#)
- [suspicious_else_formatting](#) : No longer lints in proc-macros [#7707](#)
- [excessive_precision](#) : No longer lints when in some cases the float was already written in the shortest form [#7722](#)
- [doc_markdown](#) : No longer lints on intra-doc links [#7772](#)

Suggestion Fixes/Improvements

- [unnecessary_operation](#) : Recommend using an `assert!` instead of using a function call in an indexing operation [#7453](#)
- [manual_split_once](#) : Produce semantically equivalent suggestion when `rsplitn` is used [#7663](#)
- [while_let_on_iterator](#) : Produce correct suggestion when using `&mut` [#7690](#)
- [manual_assert](#) : No better handles complex conditions [#7741](#)
- Correctly handle signs in exponents in numeric literals lints [#7747](#)
- [suspicious_map](#) : Now also suggests to use `inspect` as an alternative [#7770](#)
- Drop exponent from suggestion if it is 0 in numeric literals lints [#7774](#)

ICE Fixes

- [implicit_hasher](#) [#7761](#)

Others

- Clippy now uses the 2021 [Edition!](#) [#7664](#)

Rust 1.56

Released 2021-10-21

[74d1561...7bfc26e](#)

New Lints

- [unwrap_or_else_default](#) [#7516](#)

Enhancements

- [needless_continue](#) : Now also lints in `loop { continue; } case` [#7477](#)
- [disallowed_types](#) : Now also primitive types can be disallowed [#7488](#)
- [manual_swap](#) : Now also lints on xor swaps [#7506](#)
- [map_flatten](#) : Now also lints on the `Result` type [#7522](#)
- [no_effect](#) : Now also lints on inclusive ranges [#7556](#)

False Positive Fixes

- [nonstandard_macro_braces](#) : No longer lints on similar named nested macros [#7478](#)
- [too_many_lines](#) : No longer lints in closures to avoid duplicated diagnostics [#7534](#)
- [similar_names](#) : No longer complains about `iter` and `item` being too similar [#7546](#)

Suggestion Fixes/Improvements

- [similar_names](#) : No longer suggests to insert or add an underscore as a fix [#7221](#)

- [new_without_default](#) : No longer shows the full qualified type path when suggesting adding a `Default` implementation [#7493](#)
- [while_let_on_iterator](#) : Now suggests re-borrowing mutable references [#7520](#)
- [extend_with_drain](#) : Improve code suggestion for mutable and immutable references [#7533](#)
- [trivially_copy_pass_by_ref](#) : Now properly handles `Self` type [#7535](#)
- [never_loop](#) : Now suggests using `if let` instead of a `for` loop when applicable [#7541](#)

Documentation Improvements

- Clippy now uses a lint to generate its lint documentation. [Lints all the way down.](#) [#7502](#)
- Reworked Clippy's website: [#7172](#) [#7279](#)
 - Added applicability information about lints
 - Added a link to jump into the implementation
 - Improved loading times
 - Adapted some styling
- `cargo clippy --help` now also explains the `--fix` and `--no-deps` flag [#7492](#)
- [unnested_or_patterns](#) : Removed `or_patterns` feature gate in the code example [#7507](#)

Rust 1.55

Released 2021-09-09

[3ae8faf...74d1561](#)

Important Changes

- Stabilized `cargo clippy --fix` :tada: [#7405](#)

New Lints

- [rc_mutex](#) [#7316](#)
- [nonstandard_macro_braces](#) [#7299](#)
- [strlen_on_c_strings](#) [#7243](#)
- [self_named_constructors](#) [#7403](#)
- [disallowed_script_idsents](#) [#7400](#)
- [disallowed_types](#) [#7315](#)
- [missing_enforced_import_renames](#) [#7300](#)
- [extend_with_drain](#) [#7270](#)

Moves and Deprecations

- Moved [from_iter_instead_of_collect](#) to `pedantic` [#7375](#)
- Added `suspicious` as a new lint group for *code that is most likely wrong or useless* [#7350](#)
 - Moved [blanket_clippy_restriction_lints](#) to `suspicious`
 - Moved [empty_loop](#) to `suspicious`
 - Moved [eval_order_dependence](#) to `suspicious`
 - Moved [float_equality_without_abs](#) to `suspicious`
 - Moved [for_loops_over_fallibles](#) to `suspicious`
 - Moved [misrefactored_assign_op](#) to `suspicious`
 - Moved [mut_range_bound](#) to `suspicious`
 - Moved [mutable_key_type](#) to `suspicious`
 - Moved [suspicious_arithmetic_impl](#) to `suspicious`

- Moved [suspicious_assignment_formatting](#) to `suspicious`
- Moved [suspicious_else_formatting](#) to `suspicious`
- Moved [suspicious_map](#) to `suspicious`
- Moved [suspicious_op_assign_impl](#) to `suspicious`
- Moved [suspicious_unary_op_formatting](#) to `suspicious`

Enhancements

- [while_let_on_iterator](#) : Now suggests `&mut iter` inside closures [#7262](#)
- [doc_markdown](#) :
 - Now detects unbalanced ticks [#7357](#)
 - Add `FreeBSD` to the default configuration as an allowed identifier [#7334](#)
- [wildcard_enum_match_arm](#) , [match_wildcard_for_single_variants](#) : Now allows wildcards for enums with unstable or hidden variants [#7407](#)
- [redundant_allocation](#) : Now additionally supports the `Arc<>` type [#7308](#)
- [blacklisted_name](#) : Now allows blacklisted names in test code [#7379](#)
- [redundant_closure](#) : Suggests `&mut` for `FnMut` [#7437](#)
- [disallowed_methods](#) , [disallowed_types](#) : The configuration values `disallowed-method` and `disallowed-type` no longer require fully qualified paths [#7345](#)
- [zst_offset](#) : Fixed lint invocation after it was accidentally suppressed [#7396](#)

False Positive Fixes

- [default_numeric_fallback](#) : No longer lints on float literals as function arguments [#7446](#)
- [use_self](#) : No longer lints on type parameters [#7288](#)
- [unimplemented](#) : Now ignores the `assert` and `debug_assert` macros [#7439](#)
- [branches_sharing_code](#) : Now always checks for block expressions [#7462](#)
- [field_reassign_with_default](#) : No longer triggers in macros [#7160](#)
- [redundant_clone](#) : No longer lints on required clones for borrowed data [#7346](#)
- [default_numeric_fallback](#) : No longer triggers in external macros [#7325](#)
- [needless_bool](#) : No longer lints in macros [#7442](#)
- [useless_format](#) : No longer triggers when additional text is being appended [#7442](#)
- [assertions_on_constants](#) : `cfg!(...)` is no longer considered to be a constant [#7319](#)

Suggestion Fixes/Improvements

- [needless_collect](#) : Now show correct lint messages for shadowed values [#7289](#)
- [wrong_pub_self_convention](#) : The deprecated message now suggest the correct configuration value [#7382](#)
- [semicolon_if_nothing_returned](#) : Allow missing semicolon in blocks with only one expression [#7326](#)

ICE Fixes

- [zero_sized_map_values](#) [#7470](#)
- [redundant_pattern_matching](#) [#7471](#)
- [modulo_one](#) [#7473](#)
- [use_self](#) [#7428](#)

Rust 1.54

Released 2021-07-29

[7c7683c...3ae8faf](#)

New Lints

- [ref_binding_to_reference](#) [#7105](#)
- [needless_bitwise_bool](#) [#7133](#)
- [unused_async](#) [#7225](#)
- [manual_str_repeat](#) [#7265](#)
- [suspicious_splitn](#) [#7292](#)

Moves and Deprecations

- Deprecate `pub_enum_variant_names` and `wrong_pub_self_convention` in favor of the new `avoid-breaking-exported-api` config option (see [Enhancements](#) [#7187](#))
- Move [inconsistent_struct_constructor](#) to `pedantic` [#7193](#)
- Move [needless_borrow](#) to `style` (now warn-by-default) [#7254](#)
- Move [suspicious_operation_groupings](#) to `nursery` [#7266](#)
- Move [semicolon_if_nothing_returned](#) to `pedantic` [#7268](#)

Enhancements

- [while_let_on_iterator](#) : Now also lints in nested loops [#6966](#)
- [single_char_pattern](#) : Now also lints on `strip_prefix` and `strip_suffix` [#7156](#)
- [needless_collect](#) : Now also lints on assignments with type annotations [#7163](#)
- [if_then_some_else_none](#) : Now works with the MSRV config [#7177](#)
- Add `avoid-breaking-exported-api` config option for the lints [enum_variant_names](#) , [large_types_passed_by_value](#) , [trivially_copy_pass_by_ref](#) , [unnecessary_wraps](#) , [upper_case_acronyms](#) , and [wrong_self_convention](#) . We recommend to set this configuration option to `false` before a major release (1.0/2.0/...) to clean up the API [#7187](#)
- [needless_collect](#) : Now lints on even more data structures [#7188](#)
- [missing_docs_in_private_items](#) : No longer sees `#[<name> = "<value>"]` like attributes as sufficient documentation [#7281](#)
- [needless_collect](#) , [short_circuit_statement](#) , [unnecessary_operation](#) : Now work as expected when used with `allow` [#7282](#)

False Positive Fixes

- [implicit_return](#) : Now takes all diverging functions in account to avoid false positives [#6951](#)
- [while_let_on_iterator](#) : No longer lints when the iterator is a struct field and the struct is used in the loop [#6966](#)
- [multiple_inherent_impl](#) : No longer lints with generic arguments [#7089](#)
- [comparison_chain](#) : No longer lints in a `const` context [#7118](#)
- [while_immutable_condition](#) : Fix false positive where mutation in the loop variable wasn't picked up [#7144](#)
- [default_trait_access](#) : No longer lints in macros [#7150](#)
- [needless_question_mark](#) : No longer lints when the inner value is implicitly dereferenced [#7165](#)
- [unused_unit](#) : No longer lints when multiple macro contexts are involved [#7167](#)
- [eval_order_dependence](#) : Fix false positive in `async` context [#7174](#)
- [unnecessary_filter_map](#) : No longer lints if the `filter_map` changes the type [#7175](#)
- [wrong_self_convention](#) : No longer lints in trait implementations of non-`Copy` types [#7182](#)
- [suboptimal_flops](#) : No longer lints on `powi(2)` [#7201](#)

- [wrong_self_convention](#) : No longer lints if there is no implicit `self` [#7215](#)
- [option_if_let_else](#) : No longer lints on `else if let` pattern [#7216](#)
- [use_self](#) , [useless_conversion](#) : Fix false positives when generic arguments are involved [#7223](#)
- [manual_unwrap_or](#) : Fix false positive with deref coercion [#7233](#)
- [similar_names](#) : No longer lints on `wparam / lparam` [#7255](#)
- [redundant_closure](#) : No longer lints on using the `vec![]` macro in a closure [#7263](#)

Suggestion Fixes/Improvements

- [implicit_return](#) [#6951](#)
 - Fix suggestion for async functions
 - Improve suggestion with macros
 - Suggest to change `break` to `return` when appropriate
- [while_let_on_iterator](#) : Now suggests `&mut iter` when necessary [#6966](#)
- [match_single_binding](#) : Improve suggestion when match scrutinee has side effects [#7095](#)
- [needless_borrow](#) : Now suggests to also change usage sites as needed [#7105](#)
- [write_with_newline](#) : Improve suggestion when only `\n` is written to the buffer [#7183](#)
- [from_iter_instead_of_collect](#) : The suggestion is now auto applicable also when a `<_ as Trait>::_` is involved [#7264](#)
- [not_unsafe_ptr_arg_deref](#) : Improved error message [#7294](#)

ICE Fixes

- Fix ICE when running Clippy on `libstd` [#7140](#)
- [implicit_return](#) [#7242](#)

Rust 1.53

Released 2021-06-17

[6ed6f1e...7c7683c](#)

New Lints

- [option_filter_map](#) [#6342](#)
- [branches_sharing_code](#) [#6463](#)
- [needless_for_each](#) [#6706](#)
- [if_then_some_else_none](#) [#6859](#)
- [non_octal_unix_permissions](#) [#7001](#)
- [unnecessary_self_imports](#) [#7072](#)
- [bool_assert_comparison](#) [#7083](#)
- [cloned_instead_of_copied](#) [#7098](#)
- [flat_map_option](#) [#7101](#)

Moves and Deprecations

- Deprecate [filter_map](#) lint [#7059](#)
- Move [transmute_ptr_to_ptr](#) to `pedantic` [#7102](#)

Enhancements

- [mem_replace_with_default](#) : Also lint on common std constructors [#6820](#)
- [wrong_self_convention](#) : Also lint on `to_*_mut` methods [#6828](#)

- [wildcard_enum_match_arm](#) , [match_wildcard_for_single_variants](#) : #6863
 - Attempt to find a common path prefix in suggestion
 - Don't lint on `Option` and `Result`
 - Consider `Self` prefix
- [explicit_deref_methods](#) : Also lint on chained `deref` calls #6865
- [or_fun_call](#) : Also lint on `unsafe` blocks #6928
- [vec_box](#) , [linkedlist](#) , [option_option](#) : Also lint in `const` and `static` items #6938
- [search_is_some](#) : Also check for `is_none` #6942
- [string_lit_as_bytes](#) : Also lint on `into_bytes` #6959
- [len_without_is_empty](#) : Also lint if function signatures of `len` and `is_empty` don't match #6980
- [redundant_pattern_matching](#) : Also lint if the pattern is a `&` pattern #6991
- [clone_on_copy](#) : Also lint on chained method calls taking `self` by value #7000
- [missing_panics_doc](#) : Also lint on `assert_eq!` and `assert_ne!` #7029
- [needless_return](#) : Also lint in `async` functions #7067
- [unused_io_amount](#) : Also lint on expressions like `_.read().ok()?` #7100
- [iter_cloned_collect](#) : Also lint on large arrays, since `const-generics` are now stable #7138

False Positive Fixes

- [upper_case_acronyms](#) : No longer lints on public items #6805
- [suspicious_map](#) : No longer lints when side effects may occur inside the `map` call #6831
- [manual_map](#) , [manual_unwrap_or](#) : No longer lints in `const` functions #6917
- [wrong_self_convention](#) : Now respects `Copy` types #6924
- [needless_question_mark](#) : No longer lints if the `?` and the `Some(..)` come from different macro contexts #6935
- [map_entry](#) : Better detect if the entry API can be used #6937
- [or_fun_call](#) : No longer lints on some `len` function calls #6950
- [new_ret_no_self](#) : No longer lints when `Self` is returned with different generic arguments #6952
- [upper_case_acronyms](#) : No longer lints on public items #6981
- [explicit_into_iter_loop](#) : Only lint when `into_iter` is an implementation of `IntoIterator` #6982
- [expl_impl_clone_on_copy](#) : Take generic constraints into account before suggesting to use `derive` instead #6993
- [missing_panics_doc](#) : No longer lints when only debug-assertions are used #6996
- [clone_on_copy](#) : Only lint when using the `Clone` trait #7000
- [wrong_self_convention](#) : No longer lints inside a trait implementation #7002
- [redundant_clone](#) : No longer lints when the cloned value is modified while the clone is in use #7011
- [same_item_push](#) : No longer lints if the `Vec` is used in the loop body #7018
- [cargo_common_metadata](#) : Remove author requirement #7026
- [panic_in_result_fn](#) : No longer lints on `debug_assert` family #7060
- [panic](#) : No longer wrongfully lints on `debug_assert` with message #7063
- [wrong_self_convention](#) : No longer lints in trait implementations where no `self` is involved #7064
- [missing_const_for_fn](#) : No longer lints when unstable `const` function is involved #7076
- [suspicious_else_formatting](#) : Allow Allman style braces #7087
- [inconsistent_struct_constructor](#) : No longer lints in macros #7097
- [single_component_path_imports](#) : No longer lints on macro re-exports #7120

Suggestion Fixes/Improvements

- [redundant_pattern_matching](#) : Add a note when applying this lint would change the drop order [#6568](#)
- [write_literal](#) , [print_literal](#) : Add auto-applicable suggestion [#6821](#)
- [manual_map](#) : Fix suggestion for complex `if let ... else` chains [#6856](#)
- [inconsistent_struct_constructor](#) : Make lint description and message clearer [#6892](#)
- [map_entry](#) : Now suggests `or_insert` , `insert_with` or `match _ .entry(_)` as appropriate [#6937](#)
- [manual_flatten](#) : Suggest to insert `copied` if necessary [#6962](#)
- [redundant_slicing](#) : Fix suggestion when a re-borrow might be required or when the value is from a macro call [#6975](#)
- [match_wildcard_for_single_variants](#) : Fix suggestion for hidden variant [#6988](#)
- [clone_on_copy](#) : Correct suggestion when the cloned value is a macro call [#7000](#)
- [manual_map](#) : Fix suggestion at the end of an if chain [#7004](#)
- Fix needless parenthesis output in multiple lint suggestions [#7013](#)
- [needless_collect](#) : Better explanation in the lint message [#7020](#)
- [useless_vec](#) : Now considers mutability [#7036](#)
- [useless_format](#) : Wrap the content in braces if necessary [#7092](#)
- [single_match](#) : Don't suggest an equality check for types which don't implement `PartialEq` [#7093](#)
- [from_over_into](#) : Mention type in help message [#7099](#)
- [manual_unwrap_or](#) : Fix invalid code suggestion due to a macro call [#7136](#)

ICE Fixes

- [macro_use_imports](#) [#7022](#)
- [missing_panics_doc](#) [#7034](#)
- [tabs_in_doc_comments](#) [#7039](#)
- [missing_const_for_fn](#) [#7128](#)

Others

- [Clippy's lint list](#) now supports themes [#7030](#)
- Lints that were uplifted to `rustc` now mention the new `rustc` name in the deprecation warning [#7056](#)

Rust 1.52

Released 2021-05-06

[3e41797...6ed6f1e](#)

New Lints

- [from_str_radix_10](#) [#6717](#)
- [implicit_clone](#) [#6730](#)
- [semicolon_if_nothing_returned](#) [#6681](#)
- [manual_flatten](#) [#6646](#)
- [inconsistent_struct_constructor](#) [#6769](#)
- [iter_count](#) [#6791](#)
- [default_numeric_fallback](#) [#6662](#)
- [bytes_nth](#) [#6695](#)
- [filter_map_identity](#) [#6685](#)
- [manual_map](#) [#6573](#)

Moves and Deprecations

- Moved [upper_case_acronyms](#) to pedantic [#6775](#)
- Moved [manual_map](#) to nursery [#6796](#)
- Moved [unnecessary_wraps](#) to pedantic [#6765](#)
- Moved [trivial_regex](#) to nursery [#6696](#)
- Moved [naive_bytecount](#) to pedantic [#6825](#)
- Moved [upper_case_acronyms](#) to style [#6788](#)
- Moved [manual_map](#) to style [#6801](#)

Enhancements

- [disallowed_methods](#) : Now supports functions in addition to methods [#6674](#)
- [upper_case_acronyms](#) : Added a new configuration `upper-case-acronyms-aggressive` to trigger the lint if there is more than one uppercase character next to each other [#6788](#)
- [collapsible_match](#) : Now supports block comparison with different value names [#6754](#)
- [unnecessary_wraps](#) : Will now suggest removing unnecessary wrapped return unit type, like `Option<()>` [#6665](#)
- Improved value usage detection in closures [#6698](#)

False Positive Fixes

- [use_self](#) : No longer lints in macros [#6833](#)
- [use_self](#) : Fixed multiple false positives for: generics, associated types and derive implementations [#6179](#)
- [missing_inline_in_public_items](#) : No longer lints for procedural macros [#6814](#)
- [inherent_to_string](#) : No longer lints on functions with function generics [#6771](#)
- [doc_markdown](#) : Add `OpenDNS` to the default configuration as an allowed identifier [#6783](#)
- [missing_panics_doc](#) : No longer lints on `unreachable!` [#6700](#)
- [collapsible_if](#) : No longer lints on if statements with attributes [#6701](#)
- [match_same_arms](#) : Only considers empty blocks as equal if the tokens contained are the same [#6843](#)
- [redundant_closure](#) : Now ignores macros [#6871](#)
- [manual_map](#) : Fixed false positives when control flow statements like `return`, `break` etc. are used [#6801](#)
- [vec_init_then_push](#) : Fixed false positives for loops and if statements [#6697](#)
- [len_without_is_empty](#) : Will now consider multiple impl blocks and `#[allow]` on the `len` method as well as the type definition. [#6853](#)
- [let_underscore_drop](#) : Only lints on types which implement `Drop` [#6682](#)
- [unit_arg](#) : No longer lints on unit arguments when they come from a path expression. [#6601](#)
- [cargo_common_metadata](#) : No longer lints if `publish = false` is defined in the manifest [#6650](#)

Suggestion Fixes/Improvements

- [collapsible_match](#) : Fixed lint message capitalization [#6766](#)
- [or_fun_call](#) : Improved suggestions for `or_insert(vec![])` [#6790](#)
- [manual_map](#) : No longer expands macros in the suggestions [#6801](#)
- Aligned Clippy's lint messages with the rustc dev guide [#6787](#)

ICE Fixes

- [zero_sized_map_values](#) [#6866](#)

Documentation Improvements

- [useless_format](#) : Improved the documentation example [#6854](#)
- Clippy's [README.md](#) : Includes a new subsection on running Clippy as a rustc wrapper [#6782](#)

Others

- Running `cargo clippy` after `cargo check` now works as expected (`cargo clippy` and `cargo check` no longer shares the same build cache) [#6687](#)
- Cargo now re-runs Clippy if arguments after `--` provided to `cargo clippy` are changed. [#6834](#)
- Extracted Clippy's `utils` module into the new `clippy_utils` crate [#6756](#)
- Clippy lintcheck tool improvements [#6800](#) [#6735](#) [#6764](#) [#6708](#) [#6780](#) [#6686](#)

Rust 1.51

Released 2021-03-25

[4911ab1...3e41797](#)

New Lints

- [upper_case_acronyms](#) [#6475](#)
- [from_over_into](#) [#6476](#)
- [case_sensitive_file_extension_comparisons](#) [#6500](#)
- [needless_question_mark](#) [#6507](#)
- [missing_panics_doc](#) [#6523](#)
- [redundant_slicing](#) [#6528](#)
- [vec_init_then_push](#) [#6538](#)
- [ptr_as_ptr](#) [#6542](#)
- [collapsible_else_if](#) (split out from `collapsible_if`) [#6544](#)
- [inspect_for_each](#) [#6577](#)
- [manual_filter_map](#) [#6591](#)
- [exhaustive_enums](#) [#6617](#)
- [exhaustive_structs](#) [#6617](#)

Moves and Deprecations

- Replace `find_map` with `manual_find_map` [#6591](#)
- `unknown_clippy_lints` Now integrated in the `unknown_lints` rustc lint [#6653](#)

Enhancements

- [ptr_arg](#) Now also suggests to use `&Path` instead of `&PathBuf` [#6506](#)
- [cast_ptr_alignment](#) Also lint when the `pointer::cast` method is used [#6557](#)
- [collapsible_match](#) Now also deals with `&` and `*` operators in the `match` scrutinee [#6619](#)

False Positive Fixes

- [similar_names](#) Ignore underscore prefixed names [#6403](#)
- [print_literal](#) and [write_literal](#) No longer lint numeric literals [#6408](#)
- [large_enum_variant](#) No longer lints in external macros [#6485](#)
- [empty_enum](#) Only lint if `never_type` feature is enabled [#6513](#)
- [field_reassign_with_default](#) No longer lints in macros [#6553](#)

- [size_of_in_element_count](#) No longer lints when dividing by element size [#6578](#)
- [needless_return](#) No longer lints in macros [#6586](#)
- [match_overlapping_arm](#) No longer lint when first arm is completely included in second arm [#6603](#)
- [doc_markdown](#) Add `WebGL` to the default configuration as an allowed identifier [#6605](#)

Suggestion Fixes/Improvements

- [field_reassign_with_default](#) Don't expand macro in lint suggestion [#6531](#)
- [match_like_matches_macro](#) Strip references in suggestion [#6532](#)
- [single_match](#) Suggest `if` over `if let` when possible [#6574](#)
- [ref_in_deref](#) Use parentheses correctly in suggestion [#6609](#)
- [stable_sort_primitive](#) Clarify error message [#6611](#)

ICE Fixes

- [zero_sized_map_values](#) [#6582](#)

Documentation Improvements

- Improve search performance on the Clippy website and make it possible to directly search for lints on the GitHub issue tracker [#6483](#)
- Clean up `README.md` by removing outdated paragraph [#6488](#)
- [await_holding_refcell_ref](#) and [await_holding_lock](#) [#6585](#)
- [as_conversions](#) [#6608](#)

Others

- Clippy now has a [Roadmap](#) for 2021. If you like to get involved in a bigger project, take a look at the [Roadmap project page](#). All issues listed there are actively mentored [#6462](#)
- The Clippy version number now corresponds to the Rust version number [#6526](#)
- Fix oversight which caused Clippy to lint deps in some environments, where `CLIPPY_TESTS=true` was set somewhere [#6575](#)
- Add `cargo dev-lintcheck` tool to the Clippy Dev Tool [#6469](#)

Rust 1.50

Released 2021-02-11

[b20d4c1...4bd77a1](#)

New Lints

- [suspicious_operation_groupings](#) [#6086](#)
- [size_of_in_element_count](#) [#6394](#)
- [unnecessary_wraps](#) [#6070](#)
- [let_underscore_drop](#) [#6305](#)
- [collapsible_match](#) [#6402](#)
- [redundant_else](#) [#6330](#)
- [zero_sized_map_values](#) [#6218](#)
- [print_stderr](#) [#6367](#)
- [string_from_utf8_as_bytes](#) [#6134](#)

Moves and Deprecations

- Previously deprecated `str_to_string` and `string_to_string` have been un-deprecated as restriction lints [#6333](#)
- Deprecate `panic_params` lint. This is now available in rustc as `non_fmt_panics` [#6351](#)
- Move `map_err_ignore` to restriction [#6416](#)
- Move `await_holding_refcell_ref` to pedantic [#6354](#)
- Move `await_holding_lock` to pedantic [#6354](#)

Enhancements

- Add the `unreadable-literal-lint-fractions` configuration to disable the `unreadable_literal` lint for fractions [#6421](#)
- `clone_on_copy` : Now shows the type in the lint message [#6443](#)
- `redundant_pattern_matching` : Now also lints on `std::task::Poll` [#6339](#)
- `redundant_pattern_matching` : Additionally also lints on `std::net::IpAddr` [#6377](#)
- `search_is_some` : Now suggests `contains` instead of `find(foo).is_some()` [#6119](#)
- `clone_double_ref` : Now prints the reference type in the lint message [#6442](#)
- `modulo_one` : Now also lints on `-1`. [#6360](#)
- `empty_loop` : Now lints `no_std` crates, too [#6205](#)
- `or_fun_call` : Now also lints when indexing `HashMap` or `BTreeMap` [#6267](#)
- `wrong_self_convention` : Now also lints in trait definitions [#6316](#)
- `needless_borrow` : Print the type in the lint message [#6449](#)

False Positive Fixes

- `manual_range_contains` : No longer lints in `const fn` [#6382](#)
- `unnecessary_lazy_evaluations` : No longer lints if closure argument is used [#6370](#)
- `match_single_binding` : Now ignores cases with `#[cfg()]` macros [#6435](#)
- `match_like_matches_macro` : No longer lints on arms with attributes [#6290](#)
- `map_clone` : No longer lints with `deref` and `clone` [#6269](#)
- `map_clone` : No longer lints in the case of `&mut` [#6301](#)
- `needless_update` : Now ignores `non_exhaustive` structs [#6464](#)
- `needless_collect` : No longer lints when a `collect` is needed multiple times [#6313](#)
- `unnecessary_cast` : No longer lints `cfg`-dependent types [#6369](#)
- `declare_interior_mutable_const` and `borrow_interior_mutable_const` : Both now ignore enums with frozen variants [#6110](#)
- `field_reassign_with_default` : No longer lint for private fields [#6537](#)

Suggestion Fixes/Improvements

- `vec_box` : Provide correct type scope suggestion [#6271](#)
- `manual_range_contains` : Give correct suggestion when using floats [#6320](#)
- `unnecessary_lazy_evaluations` : Don't always mark suggestion as MachineApplicable [#6272](#)
- `manual_async_fn` : Improve suggestion formatting [#6294](#)
- `unnecessary_cast` : Fix incorrectly formatted float literal suggestion [#6362](#)

ICE Fixes

- Fix a crash in `from_iter_instead_of_collect` [#6304](#)
- Fix a silent crash when parsing doc comments in `needless_doctest_main` [#6458](#)

Documentation Improvements

- The lint website search has been improved ([#6477](#)):
 - Searching for lints with dashes and spaces is possible now. For example `missing-errors-doc` and `missing errors doc` are now valid aliases for lint names
 - Improved fuzzy search in lint descriptions
- Various README improvements [#6287](#)
- Add known problems to `comparison_chain` documentation [#6390](#)
- Fix example used in `cargo_common_metadata` [#6293](#)
- Improve `map_clone` documentation [#6340](#)

Others

- You can now tell Clippy about the MSRV your project supports. Please refer to the specific README section to learn more about MSRV support [here #6201](#)
- Add `--no-deps` option to avoid running on path dependencies in workspaces [#6188](#)

Rust 1.49

Released 2020-12-31

[e636b88...b20d4c1](#)

New Lints

- `field_reassign_with_default` [#5911](#)
- `await_holding_refcell_ref` [#6029](#)
- `disallowed_methods` [#6081](#)
- `inline_asm_x86_att_syntax` [#6092](#)
- `inline_asm_x86_intel_syntax` [#6092](#)
- `from_iter_instead_of_collect` [#6101](#)
- `mut_mutex_lock` [#6103](#)
- `single_element_loop` [#6109](#)
- `manual_unwrap_or` [#6123](#)
- `large_types_passed_by_value` [#6135](#)
- `result_unit_err` [#6157](#)
- `ref_option_ref` [#6165](#)
- `manual_range_contains` [#6177](#)
- `unusual_byte_groupings` [#6183](#)
- `comparison_to_empty` [#6226](#)
- `map_collect_result_unit` [#6227](#)
- `manual_ok_or` [#6233](#)

Moves and Deprecations

- Rename `single_char_push_str` to `single_char_add_str` [#6037](#)
- Rename `zero_width_space` to `invisible_characters` [#6105](#)
- Deprecate `drop_bounds` (uplifted) [#6111](#)
- Move `string_lit_as_bytes` to nursery [#6117](#)
- Move `rc_buffer` to `restriction` [#6128](#)

Enhancements

- `manual_memcpy` : Also lint when there are loop counters (and produce a reliable suggestion) [#5727](#)

- [single_char_add_str](#) : Also lint on `String::insert_str` [#6037](#)
- [invisible_characters](#) : Also lint the characters `\u{AD}` and `\u{2060}` [#6105](#)
- [eq_op](#) : Also lint on the `assert_*` macro family [#6167](#)
- [items_after_statements](#) : Also lint in local macro expansions [#6176](#)
- [unnecessary_cast](#) : Also lint casts on integer and float literals [#6187](#)
- [manual_unwrap_or](#) : Also lint `Result::unwrap_or` [#6190](#)
- [match_like_matches_macro](#) : Also lint when `match` has more than two arms [#6216](#)
- [integer_arithmetic](#) : Better handle `/` `an` `%` operators [#6229](#)

False Positive Fixes

- [needless_lifetimes](#) : Bail out if the function has a `where` clause with the lifetime [#5978](#)
- [explicit_counter_loop](#) : No longer lints, when loop counter is used after it is incremented [#6076](#)
- [or_fun_call](#) : Revert changes addressing the handling of `const fn` [#6077](#)
- [needless_range_loop](#) : No longer lints, when the iterable is used in the range [#6102](#)
- [inconsistent_digit_grouping](#) : Fix bug when using floating point exponent [#6104](#)
- [mistyped_literal_suffixes](#) : No longer lints on the fractional part of a float (e.g. `713.32_64`) [#6114](#)
- [invalid_regex](#) : No longer lint on unicode characters within `bytes::Regex` [#6132](#)
- [boxed_local](#) : No longer lints on `extern fn` arguments [#6133](#)
- [needless_lifetimes](#) : Fix regression, where lifetime is used in `where` clause [#6198](#)

Suggestion Fixes/Improvements

- [unnecessary_sort_by](#) : Avoid dereferencing the suggested closure parameter [#6078](#)
- [needless_arbitrary_self_type](#) : Correctly handle expanded code [#6093](#)
- [useless_format](#) : Preserve raw strings in suggestion [#6151](#)
- [empty_loop](#) : Suggest alternatives [#6162](#)
- [borrowed_box](#) : Correctly add parentheses in suggestion [#6200](#)
- [unused_unit](#) : Improve suggestion formatting [#6247](#)

Documentation Improvements

- Some doc improvements:
 - [rc_buffer](#) [#6090](#)
 - [empty_loop](#) [#6162](#)
- [doc_markdown](#) : Document problematic link text style [#6107](#)

Rust 1.48

Released 2020-11-19

[09bd400...e636b88](#)

New lints

- [self_assignment](#) [#5894](#)
- [unnecessary_lazy_evaluations](#) [#5720](#)
- [manual_strip](#) [#6038](#)
- [map_err_ignore](#) [#5998](#)
- [rc_buffer](#) [#6044](#)
- [to_string_in_display](#) [#5831](#)

- `single_char_push_str` [#5881](#)

Moves and Deprecations

- Downgrade `verbose_bit_mask` to pedantic [#6036](#)

Enhancements

- Extend `precedence` to handle chains of methods combined with unary negation [#5928](#)
- `useless_vec` : add a configuration value for the maximum allowed size on the stack [#5907](#)
- `suspicious_arithmetic_impl` : extend to implementations of `BitAnd` , `BitOr` , `BitXor` , `Rem` , `Shl` , and `Shr` [#5884](#)
- `invalid_atomic_ordering` : detect misuse of `compare_exchange` , `compare_exchange_weak` , and `fetch_update` [#6025](#)
- Avoid `redundant_pattern_matching` triggering in macros [#6069](#)
- `option_if_let_else` : distinguish pure from impure `else` expressions [#5937](#)
- `needless_doctest_main` : parse doctests instead of using textual search [#5912](#)
- `wildcard_imports` : allow `prelude` to appear in any segment of an import [#5929](#)
- Re-enable `len_zero` for ranges now that `range_is_empty` is stable [#5961](#)
- `option_as_ref_deref` : catch fully-qualified calls to `Deref::deref` and `DerefMut::deref_mut` [#5933](#)

False Positive Fixes

- `useless_attribute` : permit allowing `wildcard_imports` and `enum_glob_use` [#5994](#)
- `transmute_ptr_to_ptr` : avoid suggesting dereferencing raw pointers in const contexts [#5999](#)
- `redundant_closure_call` : take into account usages of the closure in nested functions and closures [#5920](#)
- Fix false positive in `borrow_interior_mutable_const` when referencing a field behind a pointer [#5949](#)
- `doc_markdown` : allow using "GraphQL" without backticks [#5996](#)
- `to_string_in_display` : avoid linting when calling `to_string()` on anything that is not `self` [#5971](#)
- `indexing_slicing` and `out_of_bounds_indexing` treat references to arrays as arrays [#6034](#)
- `should_implement_trait` : ignore methods with lifetime parameters [#5725](#)
- `needless_return` : avoid linting if a temporary borrows a local variable [#5903](#)
- Restrict `unnecessary_sort_by` to non-reference, Copy types [#6006](#)
- Avoid suggesting `from_bits` / `to_bits` in const contexts in `transmute_int_to_float` [#5919](#)
- `declare_interior_mutable_const` and `borrow_interior_mutable_const` : improve detection of interior mutable types [#6046](#)

Suggestion Fixes/Improvements

- `let_and_return` : add a cast to the suggestion when the return expression has adjustments [#5946](#)
- `useless_conversion` : show the type in the error message [#6035](#)
- `unnecessary_mut_passed` : discriminate between functions and methods in the error message [#5892](#)
- `float_cmp` and `float_cmp_const` : change wording to make margin of error less ambiguous [#6043](#)
- `default_trait_access` : do not use unnecessary type parameters in the suggestion [#5993](#)
- `collapsible_if` : don't use expanded code in the suggestion [#5992](#)
- Do not suggest empty format strings in `print_with_newline` and `write_with_newline` [#6042](#)
- `unit_arg` : improve the readability of the suggestion [#5931](#)

- `stable_sort_primitive` : print the type that is being sorted in the lint message [#5935](#)
- Show line count and max lines in `too_many_lines` lint message [#6009](#)
- Keep parentheses in the suggestion of `useless_conversion` where applicable [#5900](#)
- `option_map_unit_fn` and `result_map_unit_fn` : print the unit type `()` explicitly [#6024](#)
- `redundant_allocation` : suggest replacing `Rc<Box<T>>` with `Rc<T>` [#5899](#)
- Make lint messages adhere to rustc dev guide conventions [#5893](#)

ICE Fixes

- Fix ICE in `repeat_once` [#5948](#)

Documentation Improvements

- `mutable_key_type` : explain potential for false positives when the interior mutable type is not accessed in the `Hash` implementation [#6019](#)
- `unnecessary_mut_passed` : fix typo [#5913](#)
- Add example of false positive to `ptr_arg` docs. [#5885](#)
- `box_vec` , `vec_box` and `borrowed_box` : add link to the documentation of `Box` [#6023](#)

Rust 1.47

Released 2020-10-08

[c2c07fa...09bd400](#)

New lints

- `derive_ord_xor_partial_ord` [#5848](#)
- `trait_duplication_in_bounds` [#5852](#)
- `map_identity` [#5694](#)
- `unit_return_expect_ord` [#5737](#)
- `pattern_type_mismatch` [#4841](#)
- `repeat_once` [#5773](#)
- `same_item_push` [#5825](#)
- `needless_arbitrary_self_type` [#5869](#)
- `match_like_matches_macro` [#5769](#)
- `stable_sort_primitive` [#5809](#)
- `blanket_clippy_restriction_lints` [#5750](#)
- `option_if_let_else` [#5301](#)

Moves and Deprecations

- Deprecate `regex_macro` lint [#5760](#)
- Move `range_minus_one` to pedantic [#5752](#)

Enhancements

- Improve `needless_collect` by catching `collect` calls followed by `iter` or `into_iter` calls [#5837](#)
- `panic` , `todo` , `unimplemented` and `unreachable` now detect calls with formatting [#5811](#)
- Detect more cases of `suboptimal_flops` and `imprecise_flops` [#5443](#)
- Handle asymmetrical implementations of `PartialEq` in `cmp_owned` [#5701](#)
- Make it possible to allow `unsafe_derive_deserialize` [#5870](#)

- Catch `ord.min(a).max(b)` where `a < b` in [min_max](#) #5871
- Make [clone_on_copy](#) suggestion machine applicable #5745
- Enable [len_zero](#) on ranges now that `is_empty` is stable on them #5961

False Positive Fixes

- Avoid triggering [or_fun_call](#) with const fns that take no arguments #5889
- Fix [redundant_closure_call](#) false positive for closures that have multiple calls #5800
- Don't lint cases involving `ManuallyDrop` in [redundant_clone](#) #5824
- Treat a single expression the same as a single statement in the 2nd arm of a match in [single_match_else](#) #5771
- Don't trigger [unnested_or_patterns](#) if the feature `or_patterns` is not enabled #5758
- Avoid linting if key borrows in [unnecessary_sort_by](#) #5756
- Consider `Try` impl for `Poll` when generating suggestions in [try_err](#) #5857
- Take input lifetimes into account in [manual_async_fn](#) #5859
- Fix multiple false positives in [type_repetition_in_bounds](#) and add a configuration option #5761
- Limit the [suspicious_arithmetic_impl](#) lint to one binary operation #5820

Suggestion Fixes/Improvements

- Improve readability of [shadow_unrelated](#) suggestion by truncating the RHS snippet #5788
- Suggest `filter_map` instead of `flat_map` when mapping to `Option` in [map_flatten](#) #5846
- Ensure suggestion is shown correctly for long method call chains in [iter_nth_zero](#) #5793
- Drop borrow operator in suggestions of [redundant_pattern_matching](#) #5815
- Add suggestion for [iter_skip_next](#) #5843
- Improve [collapsible_if](#) fix suggestion #5732

ICE Fixes

- Fix ICE caused by [needless_collect](#) #5877
- Fix ICE caused by [unnested_or_patterns](#) #5784

Documentation Improvements

- Fix grammar of [await_holding_lock](#) documentation #5748

Others

- Make lints adhere to the rustc dev guide #5888

Rust 1.46

Released 2020-08-27

[7ea7cd1...c2c07fa](#)

New lints

- [unnested_or_patterns](#) #5378
- [iter_next_slice](#) #5597
- [unnecessary_sort_by](#) #5623
- [vec_resize_to_zero](#) #5637

Moves and Deprecations

- Move `cast_ptr_alignment` to pedantic [#5667](#)

Enhancements

- Improve `mem_replace_with_uninit` lint [#5695](#)

False Positive Fixes

- `len_zero` : Avoid linting ranges when the `range_is_empty` feature is not enabled [#5656](#)
- `let_and_return` : Don't lint if a temporary borrow is involved [#5680](#)
- `reversed_empty_ranges` : Avoid linting `N..N` in for loop arguments in [#5692](#)
- `if_same_then_else` : Don't assume multiplication is always commutative [#5702](#)
- `blacklisted_name` : Remove `bar` from the default configuration [#5712](#)
- `redundant_pattern_matching` : Avoid suggesting non-`const fn` calls in const contexts [#5724](#)

Suggestion Fixes/Improvements

- Fix suggestion of `unit_arg` lint, so that it suggest semantic equivalent code [#4455](#)
- Add auto applicable suggestion to `macro_use_imports` [#5279](#)

ICE Fixes

- Fix ICE in the `consts` module of Clippy [#5709](#)

Documentation Improvements

- Improve code examples across multiple lints [#5664](#)

Others

- Introduce a `--rustc` flag to `clippy-driver`, which turns `clippy-driver` into `rustc` and passes all the given arguments to `rustc`. This is especially useful for tools that need the `rustc` version Clippy was compiled with, instead of the Clippy version. E.g. `clippy-driver --rustc --version` will print the output of `rustc --version`. [#5178](#)
- New issue templates now make it easier to complain if Clippy is too annoying or not annoying enough! [#5735](#)

Rust 1.45

Released 2020-07-16

[891e1a8...7ea7cd1](#)

New lints

- `match_wildcard_for_single_variants` [#5582](#)
- `unsafe_derive_deserialize` [#5493](#)
- `if_let_mutex` [#5332](#)
- `mismatched_target_os` [#5506](#)
- `await_holding_lock` [#5439](#)
- `match_on_vec_items` [#5522](#)
- `manual_async_fn` [#5576](#)
- `reversed_empty_ranges` [#5583](#)
- `manual_non_exhaustive` [#5550](#)

Moves and Deprecations

- Downgrade `match_bool` to pedantic [#5408](#)
- Downgrade `match_wild_err_arm` to pedantic and update help messages. [#5622](#)
- Downgrade `useless_let_if_seq` to nursery. [#5599](#)
- Generalize `option_and_then_some` and rename to `bind_instead_of_map`. [#5529](#)
- Rename `identity_conversion` to `useless_conversion`. [#5568](#)
- Merge `block_in_if_condition_expr` and `block_in_if_condition_stmt` into `blocks_in_if_conditions`. [#5563](#)
- Merge `option_map_unwrap_or`, `option_map_unwrap_or_else` and `result_map_unwrap_or_else` into `map_unwrap_or`. [#5563](#)
- Merge `option_unwrap_used` and `result_unwrap_used` into `unwrap_used`. [#5563](#)
- Merge `option_expect_used` and `result_expect_used` into `expect_used`. [#5563](#)
- Merge `for_loop_over_option` and `for_loop_over_result` into `for_loops_over_fallibles`. [#5563](#)

Enhancements

- Avoid running cargo lints when not enabled to improve performance. [#5505](#)
- Extend `useless_conversion` with `TryFrom` and `TryInto`. [#5631](#)
- Lint also in type parameters and where clauses in `unused_unit`. [#5592](#)
- Do not suggest deriving `Default` in `new_without_default`. [#5616](#)

False Positive Fixes

- `while_let_on_iterator` [#5525](#)
- `empty_line_after_outer_attr` [#5609](#)
- `unnecessary_unwrap` [#5558](#)
- `comparison_chain` [#5596](#)
- Don't trigger `used_underscore_binding` in await desugaring. [#5535](#)
- Don't trigger `borrowed_box` on mutable references. [#5491](#)
- Allow `1 << 0` in `identity_op`. [#5602](#)
- Allow `use super::*`; glob imports in `wildcard_imports`. [#5564](#)
- Whitelist more words in `doc_markdown`. [#5611](#)
- Skip dev and build deps in `multiple_crate_versions`. [#5636](#)
- Honor `allow` attribute on arguments in `ptr_arg`. [#5647](#)
- Honor lint level attributes for `redundant_field_names`, `just_underscores_and_digits`, `many_single_char_names` and `similar_names`. [#5651](#)
- Ignore calls to `len` in `or_fun_call`. [#4429](#)

Suggestion Improvements

- Simplify suggestions in `manual_memcpy`. [#5536](#)
- Fix suggestion in `redundant_pattern_matching` for macros. [#5511](#)
- Avoid suggesting `copied()` for mutable references in `map_clone`. [#5530](#)
- Improve help message for `clone_double_ref`. [#5547](#)

ICE Fixes

- Fix ICE caused in unwrap module. [#5590](#)
- Fix ICE on rustc test issue-69020-assoc-const-arith-overflow.rs [#5499](#)

Documentation

- Clarify the documentation of [unnecessary_mut_passed](#) . #5639
- Extend example for [unneeded_field_pattern](#) . #5541

Rust 1.44

Released 2020-06-04

[204bb9b...891e1a8](#)

New lints

- [explicit_deref_methods](#) #5226
- [implicit_saturating_sub](#) #5427
- [macro_use_imports](#) #5230
- [verbose_file_reads](#) #5272
- [future_not_send](#) #5423
- [redundant_pub_crate](#) #5319
- [large_const_arrays](#) #5248
- [result_map_or_into_option](#) #5415
- [redundant_allocation](#) #5349
- [fn_address_comparisons](#) #5294
- [vtable_address_comparisons](#) #5294

Moves and Deprecations

- Deprecate [replace_consts](#) lint #5380
- Move [cognitive_complexity](#) to nursery #5428
- Move [useless_transmute](#) to nursery #5364
- Downgrade [inefficient_to_string](#) to pedantic #5412
- Downgrade [option_option](#) to pedantic #5401
- Downgrade [unreadable_literal](#) to pedantic #5419
- Downgrade [let_unit_value](#) to pedantic #5409
- Downgrade [trivially_copy_pass_by_ref](#) to pedantic #5410
- Downgrade [implicit_hasher](#) to pedantic #5411

Enhancements

- On *nightly* you can now use `cargo clippy --fix -Z unstable-options` to auto-fix lints that support this #5363
- Make [redundant_clone](#) also trigger on cases where the cloned value is not consumed. #5304
- Expand [integer_arithmetic](#) to also disallow bit-shifting #5430
- [option_as_ref_deref](#) now detects more deref cases #5425
- [large_enum_variant](#) now report the sizes of the largest and second-largest variants #5466
- [bool_comparison](#) now also checks for inequality comparisons that can be written more concisely #5365
- Expand [clone_on_copy](#) to work in method call arguments as well #5441
- [redundant_pattern_matching](#) now also handles `while let` #5483
- [integer_arithmetic](#) now also lints references of integers #5329
- Expand [float_cmp_const](#) to also work on arrays #5345
- Trigger [map_flatten](#) when map is called on an `Option` #5473

False Positive Fixes

- [many_single_char_names](#) #5468
- [should_implement_trait](#) #5437
- [unused_self](#) #5387
- [redundant_clone](#) #5453
- [precedence](#) #5445
- [suspicious_op_assign_impl](#) #5424
- [needless_lifetimes](#) #5293
- [redundant_pattern](#) #5287
- [inconsistent_digit_grouping](#) #5451

Suggestion Improvements

- Improved [question_mark](#) lint suggestion so that it doesn't add redundant `as_ref()` #5481
- Improve the suggested placeholder in [option_map_unit_fn](#) #5292
- Improve suggestion for [match_single_binding](#) when triggered inside a closure #5350

ICE Fixes

- Handle the unstable `trivial_bounds` feature #5296
- `shadow_*` lints #5297

Documentation

- Fix documentation generation for configurable lints #5353
- Update documentation for [new_ret_no_self](#) #5448
- The documentation for [option_option](#) now suggest using a tri-state enum #5403
- Fix bit mask example in [verbose_bit_mask](#) documentation #5454
- [wildcard_imports](#) documentation now mentions that `use ...::prelude::*` is not linted #5312

Rust 1.43

Released 2020-04-23

[4ee1206...204bb9b](#)

New lints

- [imprecise_flops](#) #4897
- [suboptimal_flops](#) #4897
- [wildcard_imports](#) #5029
- [single_component_path_imports](#) #5058
- [match_single_binding](#) #5061
- [let_underscore_lock](#) #5101
- [struct_excessive_bools](#) #5125
- [fn_params_excessive_bools](#) #5125
- [option_env_unwrap](#) #5148
- [lossy_float_literal](#) #5202
- [rest_pat_in_fully_bound_structs](#) #5258

Moves and Deprecations

- Move [unneeded_field_pattern](#) to pedantic group #5200

Enhancements

- Make [missing_errors_doc](#) lint also trigger on `async` functions [#5181](#)
- Add more constants to [approx_constant](#) [#5193](#)
- Extend [question_mark](#) lint [#5266](#)

False Positive Fixes

- [use_debug](#) [#5047](#)
- [unnecessary_unwrap](#) [#5132](#)
- [zero_prefixed_literal](#) [#5170](#)
- [missing_const_for_fn](#) [#5216](#)

Suggestion Improvements

- Improve suggestion when blocks of code are suggested [#5134](#)

ICE Fixes

- `misc_early` lints [#5129](#)
- [missing_errors_doc](#) [#5213](#)
- Fix ICE when evaluating `usize` s [#5256](#)

Documentation

- Improve documentation of [iter_nth_zero](#)
- Add documentation pages for stable releases [#5171](#)

Others

- Clippy now completely runs on GitHub Actions [#5190](#)

Rust 1.42

Released 2020-03-12

[69f99e7...4ee1206](#)

New lints

- [filetype_is_file](#) [#4543](#)
- [let_underscore_must_use](#) [#4823](#)
- [modulo_arithmetic](#) [#4867](#)
- [mem_replace_with_default](#) [#4881](#)
- [mutable_key_type](#) [#4885](#)
- [option_as_ref_deref](#) [#4945](#)
- [wildcard_in_or_patterns](#) [#4960](#)
- [iter_nth_zero](#) [#4966](#)
- `invalid_atomic_ordering` [#4999](#)
- [skip_while_next](#) [#5067](#)

Moves and Deprecations

- Move [transmute_float_to_int](#) from nursery to complexity group [#5015](#)
- Move [range_plus_one](#) to pedantic group [#5057](#)
- Move [debug_assert_with_mut_call](#) to nursery group [#5106](#)

- Deprecate `unused_label` [#4930](#)

Enhancements

- Lint vectored IO in `unused_io_amount` [#5027](#)
- Make `vec_box` configurable by adding a size threshold [#5081](#)
- Also lint constants in `cmp_nan` [#4910](#)
- Fix false negative in `expect_fun_call` [#4915](#)
- Fix false negative in `redundant_clone` [#5017](#)

False Positive Fixes

- `map_clone` [#4937](#)
- `replace_consts` [#4977](#)
- `let_and_return` [#5008](#)
- `eq_op` [#5079](#)
- `possible_missing_comma` [#5083](#)
- `debug_assert_with_mut_call` [#5106](#)
- Don't trigger `let_underscore_must_use` in external macros [#5082](#)
- Don't trigger `empty_loop` in `no_std` crates [#5086](#)

Suggestion Improvements

- `option_map_unwrap_or` [#4634](#)
- `wildcard_enum_match_arm` [#4934](#)
- `cognitive_complexity` [#4935](#)
- `decimal_literal_representation` [#4956](#)
- `unknown_clippy_lints` [#4963](#)
- `explicit_into_iter_loop` [#4978](#)
- `useless_attribute` [#5022](#)
- `if_let_some_result` [#5032](#)

ICE fixes

- `unsound_collection_transmute` [#4975](#)

Documentation

- Improve documentation of `empty_enum`, `replace_consts`, `redundant_clone`, and `iterator_step_by_zero`

Rust 1.41

Released 2020-01-30

[c8e3cfb...69f99e7](#)

- New Lints:
 - `exit` [#4697](#)
 - `to_digit_is_some` [#4801](#)
 - `tabs_in_doc_comments` [#4806](#)
 - `large_stack_arrays` [#4807](#)
 - `same_functions_in_if_condition` [#4814](#)
 - `zst_offset` [#4816](#)

- [as_conversions](#) [#4821](#)
- [missing_errors_doc](#) [#4884](#)
- [transmute_float_to_int](#) [#4889](#)
- Remove plugin interface, see [Inside Rust Blog](#) for details [#4714](#)
- Move [use_self](#) to nursery group [#4863](#)
- Deprecate [into_iter_on_array](#) [#4788](#)
- Expand [string_lit_as_bytes](#) to also trigger when literal has escapes [#4808](#)
- Fix false positive in [comparison_chain](#) [#4842](#)
- Fix false positive in [while_immutable_condition](#) [#4730](#)
- Fix false positive in [explicit_counter_loop](#) [#4803](#)
- Fix false positive in [must_use_candidate](#) [#4794](#)
- Fix false positive in [print_with_newline](#) and [write_with_newline](#) [#4769](#)
- Fix false positive in [derive_hash_xor_eq](#) [#4766](#)
- Fix false positive in [missing_inline_in_public_items](#) [#4870](#)
- Fix false positive in [string_add](#) [#4880](#)
- Fix false positive in [float_arithmetic](#) [#4851](#)
- Fix false positive in [cast_sign_loss](#) [#4883](#)
- Fix false positive in [manual_swap](#) [#4877](#)
- Fix ICEs occurring while checking some block expressions [#4772](#)
- Fix ICE in [use_self](#) [#4776](#)
- Fix ICEs related to [const_generics](#) [#4780](#)
- Display help when running [clippy-driver](#) without arguments, instead of ICEing [#4810](#)
- Clippy has its own ICE message now [#4588](#)
- Show deprecated lints in the documentation again [#4757](#)
- Improve Documentation by adding positive examples to some lints [#4832](#)

Rust 1.40

Released 2019-12-19

[4e7e71b...c8e3cfb](#)

- New Lints:
 - [unnneeded_wildcard_pattern](#) [#4537](#)
 - [needless_doctest_main](#) [#4603](#)
 - [suspicious_unary_op_formatting](#) [#4615](#)
 - [debug_assert_with_mut_call](#) [#4680](#)
 - [unused_self](#) [#4619](#)
 - [inefficient_to_string](#) [#4683](#)
 - [must_use_unit](#) [#4560](#)
 - [must_use_candidate](#) [#4560](#)
 - [double_must_use](#) [#4560](#)
 - [comparison_chain](#) [#4569](#)
 - [unsound_collection_transmute](#) [#4592](#)
 - [panic](#) [#4657](#)
 - [unreachable](#) [#4657](#)
 - [todo](#) [#4657](#)
 - [option_expect_used](#) [#4657](#)

- `result_expect_used` [#4657](#)
- Move `redundant_clone` to perf group [#4509](#)
- Move `manual_mul_add` to nursery group [#4736](#)
- Expand `unit_cmp` to also work with `assert_eq!`, `debug_assert_eq!`, `assert_ne!` and `debug_assert_ne!` [#4613](#)
- Expand `integer_arithmetic` to also detect mutating arithmetic like `+=` [#4585](#)
- Fix false positive in `nonminimal_bool` [#4568](#)
- Fix false positive in `missing_safety_doc` [#4611](#)
- Fix false positive in `cast_sign_loss` [#4614](#)
- Fix false positive in `redundant_clone` [#4509](#)
- Fix false positive in `try_err` [#4721](#)
- Fix false positive in `toplevel_ref_arg` [#4570](#)
- Fix false positive in `multiple_inherent_impl` [#4593](#)
- Improve more suggestions and tests in preparation for the unstable `cargo fix --clippy` [#4575](#)
- Improve suggestion for `zero_ptr` [#4599](#)
- Improve suggestion for `explicit_counter_loop` [#4691](#)
- Improve suggestion for `mul_add` [#4602](#)
- Improve suggestion for `assertions_on_constants` [#4635](#)
- Fix ICE in `use_self` [#4671](#)
- Fix ICE when encountering const casts [#4590](#)

Rust 1.39

Released 2019-11-07

[3aea860..4e7e71b](#)

- New Lints:
 - `uninit_assumed_init` [#4479](#)
 - `flat_map_identity` [#4231](#)
 - `missing_safety_doc` [#4535](#)
 - `mem_replace_with_uninit` [#4511](#)
 - `suspicious_map` [#4394](#)
 - `option_and_then_some` [#4386](#)
 - `manual_saturating_arithmetic` [#4498](#)
- Deprecate `unused_collect` lint. This is fully covered by rustc's `#[must_use]` on `collect` [#4348](#)
- Move `type_repetition_in_bounds` to pedantic group [#4403](#)
- Move `cast_lossless` to pedantic group [#4539](#)
- `temporary_cstring_as_ptr` now catches more cases [#4425](#)
- `use_self` now works in constructors, too [#4525](#)
- `cargo_common_metadata` now checks for license files [#4518](#)
- `cognitive_complexity` now includes the measured complexity in the warning message [#4469](#)
- Fix false positives in `block_in_if_*` lints [#4458](#)
- Fix false positive in `cast_lossless` [#4473](#)
- Fix false positive in `clone_on_copy` [#4411](#)
- Fix false positive in `deref_addrof` [#4487](#)
- Fix false positive in `too_many_lines` [#4490](#)

- Fix false positive in `new_ret_no_self` [#4365](#)
- Fix false positive in `manual_swap` [#4478](#)
- Fix false positive in `missing_const_for_fn` [#4450](#)
- Fix false positive in `extra_unused_lifetimes` [#4477](#)
- Fix false positive in `inherent_to_string` [#4460](#)
- Fix false positive in `map_entry` [#4495](#)
- Fix false positive in `unused_unit` [#4445](#)
- Fix false positive in `redundant_pattern` [#4489](#)
- Fix false positive in `wrong_self_convention` [#4369](#)
- Improve various suggestions and tests in preparation for the unstable `cargo fix --clippy` [#4558](#)
- Improve suggestions for `redundant_pattern_matching` [#4352](#)
- Improve suggestions for `explicit_write` [#4544](#)
- Improve suggestion for `or_fun_call` [#4522](#)
- Improve suggestion for `match_as_ref` [#4446](#)
- Improve suggestion for `unnecessary_fold_span` [#4382](#)
- Add suggestions for `unseparated_literal_suffix` [#4401](#)
- Add suggestions for `char_lit_as_u8` [#4418](#)

Rust 1.38

Released 2019-09-26

[e3cb40e...3aea860](#)

- New Lints:
 - `main_recursion` [#4203](#)
 - `inherent_to_string` [#4259](#)
 - `inherent_to_string_shadow_display` [#4259](#)
 - `type_repetition_in_bounds` [#3766](#)
 - `try_err` [#4222](#)
- Move `{unnecessary,panicking}_unwrap` out of nursery [#4307](#)
- Extend the `use_self` lint to suggest uses of `Self::Variant` [#4308](#)
- Improve suggestion for needless return [#4262](#)
- Add auto-fixable suggestion for `let_unit` [#4337](#)
- Fix false positive in `pub_enum_variant_names` and `enum_variant_names` [#4345](#)
- Fix false positive in `cast_ptr_alignment` [#4257](#)
- Fix false positive in `string_lit_as_bytes` [#4233](#)
- Fix false positive in `needless_lifetimes` [#4266](#)
- Fix false positive in `float_cmp` [#4275](#)
- Fix false positives in `needless_return` [#4274](#)
- Fix false negative in `match_same_arms` [#4246](#)
- Fix incorrect suggestion for `needless_bool` [#4335](#)
- Improve suggestion for `cast_ptr_alignment` [#4257](#)
- Improve suggestion for `single_char_literal` [#4361](#)
- Improve suggestion for `len_zero` [#4314](#)
- Fix ICE in `implicit_hasher` [#4268](#)
- Fix allow bug in `trivially_copy_pass_by_ref` [#4250](#)

Rust 1.37

Released 2019-08-15

[082cfa7...e3cb40e](#)

- New Lints:
 - [checked_conversions](#) #4088
 - [get_last_with_len](#) #3832
 - [integer_division](#) #4195
- Renamed Lint: `const_static_lifetime` is now called [redundant_static_lifetimes](#). The lint now covers statics in addition to consts #4162
- [match_same_arms](#) now warns for all identical arms, instead of only the first one #4102
- [needless_return](#) now works with void functions #4220
- Fix false positive in [redundant_closure](#) #4190
- Fix false positive in [useless_attribute](#) #4107
- Fix incorrect suggestion for [float_cmp](#) #4214
- Add suggestions for [print_with_newline](#) and [write_with_newline](#) #4136
- Improve suggestions for `option_map_unwrap_or_else` and `result_map_unwrap_or_else` #4164
- Improve suggestions for [non_ascii_literal](#) #4119
- Improve diagnostics for [let_and_return](#) #4137
- Improve diagnostics for [trivially_copy_pass_by_ref](#) #4071
- Add macro check for [unreadable_literal](#) #4099

Rust 1.36

Released 2019-07-04

[eb9f9b1...082cfa7](#)

- New lints: [find_map](#), [filter_map_next](#) #4039
- New lint: [path_buf_push_overwrite](#) #3954
- Move `path_buf_push_overwrite` to the nursery #4013
- Split [redundant_closure](#) into [redundant_closure](#) and [redundant_closure_for_method_calls](#) #4110
- Allow allowing of [toplevel_ref_arg](#) lint #4007
- Fix false negative in [or_fun_call](#) pertaining to nested constructors #4084
- Fix false positive in [or_fun_call](#) pertaining to enum variant constructors #4018
- Fix false positive in [useless_let_if_seq](#) pertaining to interior mutability #4035
- Fix false positive in [redundant_closure](#) pertaining to non-function types #4008
- Fix false positive in [let_and_return](#) pertaining to attributes on `let s` #4024
- Fix false positive in [module_name_repetitions](#) lint pertaining to attributes #4006
- Fix false positive on [assertions_on_constants](#) pertaining to `debug_assert!` #3989
- Improve suggestion in [map_clone](#) to suggest `.copied()` where applicable #3970 #4043
- Improve suggestion for [search_is_some](#) #4049
- Improve suggestion applicability for [naive_bytecount](#) #3984
- Improve suggestion applicability for [while_let_loop](#) #3975
- Improve diagnostics for [too_many_arguments](#) #4053
- Improve diagnostics for [cast_lossless](#) #4021

- Deal with macro checks in desugarings better [#4082](#)
- Add macro check for `unnecessary_cast` [#4026](#)
- Remove `approx_constant` 's documentation's "Known problems" section. [#4027](#)
- Fix ICE in `suspicious_else_formatting` [#3960](#)
- Fix ICE in `decimal_literal_representation` [#3931](#)

Rust 1.35

Released 2019-05-20

[1fac380..37f5c1e](#)

- New lint: `drop_bounds` to detect `T: Drop` bounds
- Split `redundant_closure` into `redundant_closure` and `redundant_closure_for_method_calls` [#4110](#)
- Rename `cyclomatic_complexity` to `cognitive_complexity`, start work on making lint more practical for Rust code
- Move `get_unwrap` to the restriction category
- Improve suggestions for `iter_cloned_collect`
- Improve suggestions for `cast_lossless` to suggest suffixed literals
- Fix false positives in `print_with_newline` and `write_with_newline` pertaining to raw strings
- Fix false positive in `needless_range_loop` pertaining to structs without a `.iter()`
- Fix false positive in `bool_comparison` pertaining to non-bool types
- Fix false positive in `redundant_closure` pertaining to differences in borrows
- Fix false positive in `option_map_unwrap_or` on non-copy types
- Fix false positives in `missing_const_for_fn` pertaining to macros and trait method impls
- Fix false positive in `needless_pass_by_value` pertaining to procedural macros
- Fix false positive in `needless_continue` pertaining to loop labels
- Fix false positive for `boxed_local` pertaining to arguments moved into closures
- Fix false positive for `use_self` in nested functions
- Fix suggestion for `expect_fun_call` (<https://github.com/rust-lang/rust-clippy/pull/3846>)
- Fix suggestion for `explicit_counter_loop` to deal with parenthesizing range variables
- Fix suggestion for `single_char_pattern` to correctly escape single quotes
- Avoid triggering `redundant_closure` in macros
- ICE fixes: [#3805](#), [#3772](#), [#3741](#)

Rust 1.34

Released 2019-04-10

[1b89724...1fac380](#)

- New lint: `assertions_on_constants` to detect for example `assert!(true)`
- New lint: `dbg_macro` to detect uses of the `dbg!` macro
- New lint: `missing_const_for_fn` that can suggest functions to be made `const`
- New lint: `too_many_lines` to detect functions with excessive LOC. It can be configured using the `too-many-lines-threshold` configuration.
- New lint: `wildcard_enum_match_arm` to check for wildcard enum matches using `_`
- Expand `redundant_closure` to also work for methods (not only functions)
- Fix ICEs in `vec_box`, `needless_pass_by_value` and `implicit_hasher`

- Fix false positive in `cast_sign_loss`
- Fix false positive in `integer_arithmetic`
- Fix false positive in `unit_arg`
- Fix false positives in `implicit_return`
- Add suggestion to `explicit_write`
- Improve suggestions for `question_mark` lint
- Fix incorrect suggestion for `cast_lossless`
- Fix incorrect suggestion for `expect_fun_call`
- Fix incorrect suggestion for `needless_bool`
- Fix incorrect suggestion for `needless_range_loop`
- Fix incorrect suggestion for `use_self`
- Fix incorrect suggestion for `while_let_on_iterator`
- Clippy is now slightly easier to invoke in non-cargo contexts. See [#3665](#) for more details.
- We now have [improved documentation](#) on how to add new lints

Rust 1.33

Released 2019-02-26

[b2601be...1b89724](#)

- New lints: `implicit_return`, `vec_box`, `cast_ref_to_mut`
- The `rust-clippy` repository is now part of the `rust-lang` org.
- Rename `stutter` to `module_name_repetitions`
- Merge `new_without_default_derive` into `new_without_default` lint
- Move `large_digit_groups` from `style` group to `pedantic`
- Expand `bool_comparison` to check for `<`, `<=`, `>`, `>=`, and `!=` comparisons against booleans
- Expand `no_effect` to detect writes to constants such as `A_CONST.field = 2`
- Expand `redundant_clone` to work on struct fields
- Expand `suspicious_else_formatting` to detect `if .. {...} {...}`
- Expand `use_self` to work on tuple structs and also in local macros
- Fix ICE in `result_map_unit_fn` and `option_map_unit_fn`
- Fix false positives in `implicit_return`
- Fix false positives in `use_self`
- Fix false negative in `clone_on_copy`
- Fix false positive in `doc_markdown`
- Fix false positive in `empty_loop`
- Fix false positive in `if_same_then_else`
- Fix false positive in `infinite_iter`
- Fix false positive in `question_mark`
- Fix false positive in `useless_asref`
- Fix false positive in `wildcard_dependencies`
- Fix false positive in `write_with_newline`
- Add suggestion to `explicit_write`
- Improve suggestions for `question_mark` lint
- Fix incorrect suggestion for `get_unwrap`

Rust 1.32

Released 2019-01-17

[2e26fdc2...b2601be](#)

- New lints: `slow_vector_initialization`, `mem_discriminant_non_enum`, `redundant_clone`, `wildcard_dependencies`, `into_iter_on_ref`, `into_iter_on_array`, `deprecated_cfg_attr`, `cargo_common_metadata`
- Add support for `u128` and `i128` to integer related lints
- Add float support to `mistyped_literal_suffixes`
- Fix false positives in `use_self`
- Fix false positives in `missing_comma`
- Fix false positives in `new_ret_no_self`
- Fix false positives in `possible_missing_comma`
- Fix false positive in `integer_arithmetic` in constant items
- Fix false positive in `needless_borrow`
- Fix false positive in `out_of_bounds_indexing`
- Fix false positive in `new_without_default_derive`
- Fix false positive in `string_lit_as_bytes`
- Fix false negative in `out_of_bounds_indexing`
- Fix false negative in `use_self`. It will now also check existential types
- Fix incorrect suggestion for `redundant_closure_call`
- Fix various suggestions that contained expanded macros
- Fix `bool_comparison` triggering 3 times on on on the same code
- Expand `trivially_copy_pass_by_ref` to work on trait methods
- Improve suggestion for `needless_range_loop`
- Move `needless_pass_by_value` from `pedantic` group to `style`

Rust 1.31

Released 2018-12-06

[125907ad..2e26fdc2](#)

- Clippy has been relicensed under a dual MIT / Apache license. See [#3093](#) for more information.
- With Rust 1.31, Clippy is no longer available via crates.io. The recommended installation method is via `rustup component add clippy`.
- New lints: `redundant_pattern_matching`, `unnecessary_filter_map`, `unused_unit`, `map_flatten`, `mem_replace_option_with_none`
- Fix ICE in `if_let_redundant_pattern_matching`
- Fix ICE in `needless_pass_by_value` when encountering a generic function argument with a lifetime parameter
- Fix ICE in `needless_range_loop`
- Fix ICE in `single_char_pattern` when encountering a constant value
- Fix false positive in `assign_op_pattern`
- Fix false positive in `boxed_local` on trait implementations
- Fix false positive in `cmp_owned`
- Fix false positive in `collapsible_if` when conditionals have comments
- Fix false positive in `double_parens`
- Fix false positive in `excessive_precision`

- Fix false positive in `explicit_counter_loop`
- Fix false positive in `fn_to_numeric_cast_with_truncation`
- Fix false positive in `map_clone`
- Fix false positive in `new_ret_no_self`
- Fix false positive in `new_without_default` when `new` is unsafe
- Fix false positive in `type_complexity` when using extern types
- Fix false positive in `useless_format`
- Fix false positive in `wrong_self_convention`
- Fix incorrect suggestion for `excessive_precision`
- Fix incorrect suggestion for `expect_fun_call`
- Fix incorrect suggestion for `get_unwrap`
- Fix incorrect suggestion for `useless_format`
- `fn_to_numeric_cast_with_truncation` lint can be disabled again
- Improve suggestions for `manual_memcpy`
- Improve help message for `needless_lifetimes`

Rust 1.30

Released 2018-10-25

[14207503...125907ad](#)

- Deprecate `assign_ops` lint
- New lints: `mistyped_literal_suffixes`, `ptr_offset_with_cast`, `needless_collect`, `copy_iterator`
- `cargo clippy -V` now includes the Clippy commit hash of the Rust Clippy component
- Fix ICE in `implicit_hasher`
- Fix ICE when encountering `println!("{}", a);`
- Fix ICE when encountering a macro call in match statements
- Fix false positive in `default_trait_access`
- Fix false positive in `trivially_copy_pass_by_ref`
- Fix false positive in `similar_names`
- Fix false positive in `redundant_field_name`
- Fix false positive in `expect_fun_call`
- Fix false negative in `identity_conversion`
- Fix false negative in `explicit_counter_loop`
- Fix `range_plus_one` suggestion and false negative
- `print_with_newline` / `write_with_newline`: don't warn about string with several `\n`s in them
- Fix `useless_attribute` to also whitelist `unused_extern_crates`
- Fix incorrect suggestion for `single_char_pattern`
- Improve suggestion for `identity_conversion` lint
- Move `explicit_iter_loop` and `explicit_into_iter_loop` from `style` group to `pedantic`
- Move `range_plus_one` and `range_minus_one` from `nursery` group to `complexity`
- Move `shadow_unrelated` from `restriction` group to `pedantic`
- Move `indexing_slicing` from `pedantic` group to `restriction`

Rust 1.29

Released 2018-09-13

[v0.0.212...14207503](#)

- :tada: :tada: **Rust 1.29 is the first stable Rust that includes a bundled Clippy** :tada: :tada: You can now run `rustup component add clippy-preview` and then `cargo clippy` to run Clippy. This should put an end to the continuous nightly upgrades for Clippy users.
- Clippy now follows the Rust versioning scheme instead of its own
- Fix ICE when encountering a `while let (..) = x.iter()` construct
- Fix false positives in `use_self`
- Fix false positive in `trivially_copy_pass_by_ref`
- Fix false positive in `useless_attribute` lint
- Fix false positive in `print_literal`
- Fix `use_self` regressions
- Improve lint message for `neg_cmp_op_on_partial_ord`
- Improve suggestion highlight for `single_char_pattern`
- Improve suggestions for various print/write macro lints
- Improve website header

0.0.212 (2018-07-10)

- Rustup to *rustc 1.29.0-nightly (e06c87544 2018-07-06)*

0.0.211

- Rustup to *rustc 1.28.0-nightly (e3bf634e0 2018-06-28)*

0.0.210

- Rustup to *rustc 1.28.0-nightly (01cc982e9 2018-06-24)*

0.0.209

- Rustup to *rustc 1.28.0-nightly (523097979 2018-06-18)*

0.0.208

- Rustup to *rustc 1.28.0-nightly (86a8f1a63 2018-06-17)*

0.0.207

- Rustup to *rustc 1.28.0-nightly (2a0062974 2018-06-09)*

0.0.206

- Rustup to *rustc 1.28.0-nightly (5bf68db6e 2018-05-28)*

0.0.205

- Rustup to *rustc 1.28.0-nightly (990d8aa74 2018-05-25)*
- Rename `unused_lifetimes` to `extra_unused_lifetimes` because of naming conflict with new `rustc` lint

0.0.204

- Rustup to *rustc 1.28.0-nightly (71e87be38 2018-05-22)*

0.0.203

- Rustup to *rustc 1.28.0-nightly* (a3085756e 2018-05-19)
- Clippy attributes are now of the form `clippy::cyclomatic_complexity` instead of `clippy(cyclomatic_complexity)`

0.0.202

- Rustup to *rustc 1.28.0-nightly* (952f344cd 2018-05-18)

0.0.201

- Rustup to *rustc 1.27.0-nightly* (2f2a11dfc 2018-05-16)

0.0.200

- Rustup to *rustc 1.27.0-nightly* (9fae15374 2018-05-13)

0.0.199

- Rustup to *rustc 1.27.0-nightly* (ff2ac35db 2018-05-12)

0.0.198

- Rustup to *rustc 1.27.0-nightly* (acd3871ba 2018-05-10)

0.0.197

- Rustup to *rustc 1.27.0-nightly* (428ea5f6b 2018-05-06)

0.0.196

- Rustup to *rustc 1.27.0-nightly* (e82261dfb 2018-05-03)

0.0.195

- Rustup to *rustc 1.27.0-nightly* (ac3c2288f 2018-04-18)

0.0.194

- Rustup to *rustc 1.27.0-nightly* (bd40cbbe1 2018-04-14)
- New lints: `cast_ptr_alignment` , `transmute_ptr_to_ptr` , `write_literal` , `write_with_newline` , `writeln_empty_string`

0.0.193

- Rustup to *rustc 1.27.0-nightly* (eeee94c11 2018-04-06)

0.0.192

- Rustup to *rustc 1.27.0-nightly* (fb44b4c0e 2018-04-04)
- New lint: `print_literal`

0.0.191

- Rustup to *rustc 1.26.0-nightly* (ae544ee1c 2018-03-29)

- Lint audit; categorize lints as style, correctness, complexity, pedantic, nursery, restriction.

0.0.190

- Fix a bunch of intermittent cargo bugs

0.0.189

- Rustup to *rustc 1.26.0-nightly* (5508b2714 2018-03-18)

0.0.188

- Rustup to *rustc 1.26.0-nightly* (392645394 2018-03-15)
- New lint: [while_immutable_condition](#)

0.0.187

- Rustup to *rustc 1.26.0-nightly* (322d7f7b9 2018-02-25)
- New lints: [redundant_field_names](#) , [suspicious_arithmetic_impl](#) , [suspicious_op_assign_impl](#)

0.0.186

- Rustup to *rustc 1.25.0-nightly* (0c6091fbd 2018-02-04)
- Various false positive fixes

0.0.185

- Rustup to *rustc 1.25.0-nightly* (56733bc9f 2018-02-01)
- New lint: [question_mark](#)

0.0.184

- Rustup to *rustc 1.25.0-nightly* (90eb44a58 2018-01-29)
- New lints: [double_comparisons](#) , [empty_line_after_outer_attr](#)

0.0.183

- Rustup to *rustc 1.25.0-nightly* (21882aad7 2018-01-28)
- New lint: [misaligned_transmute](#)

0.0.182

- Rustup to *rustc 1.25.0-nightly* (a0dcecf9 2018-01-24)
- New lint: [decimal_literal_representation](#)

0.0.181

- Rustup to *rustc 1.25.0-nightly* (97520ccb1 2018-01-21)
- New lints: [else_if_without_else](#) , [option_option](#) , [unit_arg](#) , [unnecessary_fold](#)
- Removed `unit_expr`
- Various false positive fixes for [needless_pass_by_value](#)

0.0.180

- Rustup to *rustc 1.25.0-nightly* (3f92e8d89 2018-01-14)

0.0.179

- Rustup to *rustc 1.25.0-nightly* (61452e506 2018-01-09)

0.0.178

- Rustup to *rustc 1.25.0-nightly* (ee220daca 2018-01-07)

0.0.177

- Rustup to *rustc 1.24.0-nightly* (250b49205 2017-12-21)
- New lint: [match_as_ref](#)

0.0.176

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

0.0.175

- Rustup to *rustc 1.24.0-nightly* (bb42071f6 2017-12-01)

0.0.174

- Rustup to *rustc 1.23.0-nightly* (63739ab7b 2017-11-21)

0.0.173

- Rustup to *rustc 1.23.0-nightly* (33374fa9d 2017-11-20)

0.0.172

- Rustup to *rustc 1.23.0-nightly* (d0f8e2913 2017-11-16)

0.0.171

- Rustup to *rustc 1.23.0-nightly* (ff0f5de3b 2017-11-14)

0.0.170

- Rustup to *rustc 1.23.0-nightly* (d6b06c63a 2017-11-09)

0.0.169

- Rustup to *rustc 1.23.0-nightly* (3b82e4c74 2017-11-05)
- New lints: [just_underscores_and_digits](#), `result_map_unwrap_or_else`, [transmute_bytes_to_str](#)

0.0.168

- Rustup to *rustc 1.23.0-nightly* (f0fe716db 2017-10-30)

0.0.167

- Rustup to *rustc 1.23.0-nightly* (90ef3372e 2017-10-29)

- New lints: `const_static_lifetime` , `erasing_op` , `fallible_impl_from` , `println_empty_string` , `useless_asref`

0.0.166

- Rustup to *rustc 1.22.0-nightly* (b7960878b 2017-10-18)
- New lints: `explicit_write` , `identity_conversion` , `implicit_hasher` , `invalid_ref` , `option_map_or_none` , `range_minus_one` , `range_plus_one` , `transmute_int_to_bool` , `transmute_int_to_char` , `transmute_int_to_float`

0.0.165

- Rust upgrade to *rustc 1.22.0-nightly* (0e6f4cf51 2017-09-27)
- New lint: `mut_range_bound`

0.0.164

- Update to *rustc 1.22.0-nightly* (6c476ce46 2017-09-25)
- New lint: `int_plus_one`

0.0.163

- Update to *rustc 1.22.0-nightly* (14039a42a 2017-09-22)

0.0.162

- Update to *rustc 1.22.0-nightly* (0701b37d9 2017-09-18)
- New lint: `chars_last_cmp`
- Improved suggestions for `needless_borrow` , `ptr_arg` ,

0.0.161

- Update to *rustc 1.22.0-nightly* (539f2083d 2017-09-13)

0.0.160

- Update to *rustc 1.22.0-nightly* (dd08c3070 2017-09-12)

0.0.159

- Update to *rustc 1.22.0-nightly* (eba374fb2 2017-09-11)
- New lint: `clone_on_ref_ptr`

0.0.158

- New lint: `manual_memcpy`
- `cast_lossless` no longer has redundant parentheses in its suggestions
- Update to *rustc 1.22.0-nightly* (dead08cb3 2017-09-08)

0.0.157 - 2017-09-04

- Update to *rustc 1.22.0-nightly* (981ce7d8d 2017-09-03)
- New lint: `unit_expr`

0.0.156 - 2017-09-03

- Update to *rustc 1.22.0-nightly* (744dd6c1d 2017-09-02)

0.0.155

- Update to *rustc 1.21.0-nightly* (c11f689d2 2017-08-29)
- New lint: [infinite_iter](#) , [maybe_infinite_iter](#) , [cast_lossless](#)

0.0.154

- Update to *rustc 1.21.0-nightly* (2c0558f63 2017-08-24)
- Fix [use_self](#) triggering inside derives
- Add support for linting an entire workspace with `cargo clippy --all`
- New lint: [naive_bytecount](#)

0.0.153

- Update to *rustc 1.21.0-nightly* (8c303ed87 2017-08-20)
- New lint: [use_self](#)

0.0.152

- Update to *rustc 1.21.0-nightly* (df511d554 2017-08-14)

0.0.151

- Update to *rustc 1.21.0-nightly* (13d94d5fa 2017-08-10)

0.0.150

- Update to *rustc 1.21.0-nightly* (215e0b10e 2017-08-08)

0.0.148

- Update to *rustc 1.21.0-nightly* (37c7d0ebb 2017-07-31)
- New lints: [unreadable_literal](#) , [inconsistent_digit_grouping](#) , [large_digit_groups](#)

0.0.147

- Update to *rustc 1.21.0-nightly* (aac223f4f 2017-07-30)

0.0.146

- Update to *rustc 1.21.0-nightly* (52a330969 2017-07-27)
- Fixes false positives in `inline_always`
- Fixes false negatives in `panic_params`

0.0.145

- Update to *rustc 1.20.0-nightly* (afe145d22 2017-07-23)

0.0.144

- Update to *rustc 1.20.0-nightly* (086eaa78e 2017-07-15)

0.0.143

- Update to *rustc 1.20.0-nightly* (d84693b93 2017-07-09)
- Fix `cargo clippy` crashing on `dylib` projects
- Fix false positives around `nested_while_let` and `never_loop`

0.0.142

- Update to *rustc 1.20.0-nightly* (067971139 2017-07-02)

0.0.141

- Rewrite of the `doc_markdown` lint.
- Deprecated [range_step_by_zero](#)
- New lint: [iterator_step_by_zero](#)
- New lint: [needless_borrowed_reference](#)
- Update to *rustc 1.20.0-nightly* (69c65d296 2017-06-28)

0.0.140 - 2017-06-16

- Update to *rustc 1.19.0-nightly* (258ae6dd9 2017-06-15)

0.0.139 — 2017-06-10

- Update to *rustc 1.19.0-nightly* (4bf5c99af 2017-06-10)
- Fix bugs with for loop desugaring
- Check for [AsRef](#) / [AsMut](#) arguments in [wrong_self_convention](#)

0.0.138 — 2017-06-05

- Update to *rustc 1.19.0-nightly* (0418fa9d3 2017-06-04)

0.0.137 — 2017-06-05

- Update to *rustc 1.19.0-nightly* (6684d176c 2017-06-03)

0.0.136 — 2017—05—26

- Update to *rustc 1.19.0-nightly* (557967766 2017-05-26)

0.0.135 — 2017—05—24

- Update to *rustc 1.19.0-nightly* (5b13bff52 2017-05-23)

0.0.134 — 2017—05—19

- Update to *rustc 1.19.0-nightly* (0ed1ec9f9 2017-05-18)

0.0.133 — 2017—05—14

- Update to *rustc 1.19.0-nightly* (826d8f385 2017-05-13)

0.0.132 — 2017—05—05

- Fix various bugs and some ices

0.0.131 — 2017—05—04

- Update to *rustc 1.19.0-nightly* (2d4ed8e0c 2017-05-03)

0.0.130 — 2017—05—03

- Update to *rustc 1.19.0-nightly* (6a5fc9eec 2017-05-02)

0.0.129 — 2017-05-01

- Update to *rustc 1.19.0-nightly* (06fb4d256 2017-04-30)

0.0.128 — 2017-04-28

- Update to *rustc 1.18.0-nightly* (94e884b63 2017-04-27)

0.0.127 — 2017-04-27

- Update to *rustc 1.18.0-nightly* (036983201 2017-04-26)
- New lint: [needless_continue](#)

0.0.126 — 2017-04-24

- Update to *rustc 1.18.0-nightly* (2bd4b5c6d 2017-04-23)

0.0.125 — 2017-04-19

- Update to *rustc 1.18.0-nightly* (9f2abadca 2017-04-18)

0.0.124 — 2017-04-16

- Update to *rustc 1.18.0-nightly* (d5cf1cb64 2017-04-15)

0.0.123 — 2017-04-07

- Fix various false positives

0.0.122 — 2017-04-07

- Rustup to *rustc 1.18.0-nightly* (91ae22a01 2017-04-05)
- New lint: [op_ref](#)

0.0.121 — 2017-03-21

- Rustup to *rustc 1.17.0-nightly* (134c4a0f0 2017-03-20)

0.0.120 — 2017-03-17

- Rustup to *rustc 1.17.0-nightly* (0aeb9c129 2017-03-15)

0.0.119 — 2017-03-13

- Rustup to *rustc 1.17.0-nightly* (824c9ebbd 2017-03-12)

0.0.118 — 2017-03-05

- Rustup to *rustc 1.17.0-nightly* (b1e31766d 2017-03-03)

0.0.117 — 2017-03-01

- Rustup to *rustc 1.17.0-nightly* (be760566c 2017-02-28)

0.0.116 — 2017-02-28

- Fix `cargo clippy` on 64 bit windows systems

0.0.115 — 2017-02-27

- Rustup to *rustc 1.17.0-nightly* (60a0edc6c 2017-02-26)
- New lints: [zero_ptr](#) , [never_loop](#) , [mut_from_ref](#)

0.0.114 — 2017-02-08

- Rustup to *rustc 1.17.0-nightly* (c49d10207 2017-02-07)
- Tests are now ui tests (testing the exact output of rustc)

0.0.113 — 2017-02-04

- Rustup to *rustc 1.16.0-nightly* (eedaa94e3 2017-02-02)
- New lint: [large_enum_variant](#)
- `explicit_into_iter_loop` provides suggestions

0.0.112 — 2017-01-27

- Rustup to *rustc 1.16.0-nightly* (df8debf6d 2017-01-25)

0.0.111 — 2017-01-21

- Rustup to *rustc 1.16.0-nightly* (a52da95ce 2017-01-20)

0.0.110 — 2017-01-20

- Add badges and categories to `Cargo.toml`

0.0.109 — 2017-01-19

- Update to *rustc 1.16.0-nightly* (c07a6ae77 2017-01-17)

0.0.108 — 2017-01-12

- Update to *rustc 1.16.0-nightly* (2782e8f8f 2017-01-12)

0.0.107 — 2017-01-11

- Update regex dependency
- Fix FP when matching `&&mut` by `&ref`
- Reintroduce `for (_, x) in &mut hash_map -> for x in hash_map.values_mut()`
- New lints: [unused_io_amount](#) , [forget_ref](#) , [short_circuit_statement](#)

0.0.106 — 2017-01-04

- Fix FP introduced by rustup in [wrong_self_convention](#)

0.0.105 — 2017-01-04

- Update to *rustc 1.16.0-nightly* (468227129 2017-01-03)
- New lints: [deref_addrof](#) , [double_parens](#) , [pub_enum_variant_names](#)
- Fix suggestion in [new_without_default](#)
- FP fix in [absurd_extreme_comparisons](#)

0.0.104 — 2016-12-15

- Update to *rustc 1.15.0-nightly* (8f02c429a 2016-12-15)

0.0.103 — 2016-11-25

- Update to *rustc 1.15.0-nightly* (d5814b03e 2016-11-23)

0.0.102 — 2016-11-24

- Update to *rustc 1.15.0-nightly* (3bf2be9ce 2016-11-22)

0.0.101 — 2016-11-23

- Update to *rustc 1.15.0-nightly* (7b3eeea22 2016-11-21)
- New lint: [string_extend_chars](#)

0.0.100 — 2016-11-20

- Update to *rustc 1.15.0-nightly* (ac635aa95 2016-11-18)

0.0.99 — 2016-11-18

- Update to *rustc 1.15.0-nightly* (0ed951993 2016-11-14)
- New lint: [get_unwrap](#)

0.0.98 — 2016-11-08

- Fixes an issue due to a change in how cargo handles `--sysroot` , which broke `cargo clippy`

0.0.97 — 2016-11-03

- For convenience, `cargo clippy` defines a `cargo-clippy` feature. This was previously added for a short time under the name `clippy` but removed for compatibility.
- `cargo clippy --help` is more helping (and less helpful :smile:)
- Rustup to *rustc 1.14.0-nightly* (5665bdf3e 2016-11-02)
- New lints: [if_let_redundant_pattern_matching](#) , [partial_eq_ne_impl](#)

0.0.96 — 2016-10-22

- Rustup to *rustc 1.14.0-nightly* (f09420685 2016-10-20)
- New lint: [iter_skip_next](#)

0.0.95 — 2016-10-06

- Rustup to *rustc 1.14.0-nightly* (3210fd5c2 2016-10-05)

0.0.94 — 2016-10-04

- Fixes bustage on Windows due to forbidden directory name

0.0.93 — 2016-10-03

- Rustup to *rustc 1.14.0-nightly* (144af3e97 2016-10-02)
- `option_map_unwrap_or` and `option_map_unwrap_or_else` are now allowed by default.
- New lint: [explicit_into_iter_loop](#)

0.0.92 — 2016-09-30

- Rustup to *rustc 1.14.0-nightly* (289f3a4ca 2016-09-29)

0.0.91 — 2016-09-28

- Rustup to *rustc 1.13.0-nightly* (d0623cf7b 2016-09-26)

0.0.90 — 2016-09-09

- Rustup to *rustc 1.13.0-nightly* (f1f40f850 2016-09-09)

0.0.89 — 2016-09-06

- Rustup to *rustc 1.13.0-nightly* (cbe4de78e 2016-09-05)

0.0.88 — 2016-09-04

- Rustup to *rustc 1.13.0-nightly* (70598e04f 2016-09-03)
- The following lints are not new but were only usable through the `clippy` lint groups: [filter_next](#), [for_loop_over_option](#), [for_loop_over_result](#) and [match_overlapping_arm](#). You should now be able to `#[allow/deny]` them individually and they are available directly through `cargo clippy`.

0.0.87 — 2016-08-31

- Rustup to *rustc 1.13.0-nightly* (eac41469d 2016-08-30)
- New lints: [builtin_type_shadow](#)
- Fix FP in [zero_prefixed_literal](#) and `0b / 0o`

0.0.86 — 2016-08-28

- Rustup to *rustc 1.13.0-nightly* (a23064af5 2016-08-27)
- New lints: [missing_docs_in_private_items](#), [zero_prefixed_literal](#)

0.0.85 — 2016-08-19

- Fix ICE with [useless_attribute](#)
- [useless_attribute](#) ignores `unused_imports` on `use` statements

0.0.84 — 2016-08-18

- Rustup to *rustc 1.13.0-nightly* (aef6971ca 2016-08-17)

0.0.83 — 2016-08-17

- Rustup to *rustc 1.12.0-nightly (1bf5fa326 2016-08-16)*
- New lints: [print_with_newline](#) , [useless_attribute](#)

0.0.82 — 2016-08-17

- Rustup to *rustc 1.12.0-nightly (197be89f3 2016-08-15)*
- New lint: [module_inception](#)

0.0.81 — 2016-08-14

- Rustup to *rustc 1.12.0-nightly (1deb02ea6 2016-08-12)*
- New lints: [eval_order_dependence](#) , [mixed_case_hex_literals](#) , [unseparated_literal_suffix](#)
- False positive fix in [too_many_arguments](#)
- Addition of functionality to [needless_borrow](#)
- Suggestions for [clone_on_copy](#)
- Bug fix in [wrong_self_convention](#)
- Doc improvements

0.0.80 — 2016-07-31

- Rustup to *rustc 1.12.0-nightly (1225e122f 2016-07-30)*
- New lints: [misrefactored_assign_op](#) , [serde_api_misuse](#)

0.0.79 — 2016-07-10

- Rustup to *rustc 1.12.0-nightly (f93aaf84c 2016-07-09)*
- Major suggestions refactoring

0.0.78 — 2016-07-02

- Rustup to *rustc 1.11.0-nightly (01411937f 2016-07-01)*
- New lints: [wrong_transmute](#) , [double_neg](#) , [filter_map](#)
- For compatibility, `cargo clippy` does not defines the `clippy` feature introduced in 0.0.76 anymore
- [collapsible_if](#) now considers `if let`

0.0.77 — 2016-06-21

- Rustup to *rustc 1.11.0-nightly (5522e678b 2016-06-20)*
- New lints: `stutter` and [iter_nth](#)

0.0.76 — 2016-06-10

- Rustup to *rustc 1.11.0-nightly (7d2f75a95 2016-06-09)*
- `cargo clippy` now automatically defines the `clippy` feature
- New lint: [not_unsafe_ptr_arg_deref](#)

0.0.75 — 2016-06-08

- Rustup to *rustc 1.11.0-nightly (763f9234b 2016-06-06)*

0.0.74 — 2016-06-07

- Fix bug with `cargo-clippy` JSON parsing
- Add the `CLIPPY_DISABLE_DOCS_LINKS` environment variable to deactivate the “for further information visit *lint-link*” message.

0.0.73 — 2016-06-05

- Fix false positives in `useless_let_if_seq`

0.0.72 — 2016-06-04

- Fix false positives in `useless_let_if_seq`

0.0.71 — 2016-05-31

- Rustup to *rustc 1.11.0-nightly* (a967611d8 2016-05-30)
- New lint: `useless_let_if_seq`

0.0.70 — 2016-05-28

- Rustup to *rustc 1.10.0-nightly* (7bddce693 2016-05-27)
- `invalid_regex` and `trivial_regex` can now warn on `RegexSet::new`, `RegexBuilder::new` and byte regexes

0.0.69 — 2016-05-20

- Rustup to *rustc 1.10.0-nightly* (476fe6eef 2016-05-21)
- `used_underscore_binding` has been made `Allow` temporarily

0.0.68 — 2016-05-17

- Rustup to *rustc 1.10.0-nightly* (cd6a40017 2016-05-16)
- New lint: `unnecessary_operation`

0.0.67 — 2016-05-12

- Rustup to *rustc 1.10.0-nightly* (22ac88f1a 2016-05-11)

0.0.66 — 2016-05-11

- New `cargo clippy` subcommand
- New lints: `assign_op_pattern`, `assign_ops`, `needless_borrow`

0.0.65 — 2016-05-08

- Rustup to *rustc 1.10.0-nightly* (62e2b2fb7 2016-05-06)
- New lints: `float_arithmetic`, `integer_arithmetic`

0.0.64 — 2016-04-26

- Rustup to *rustc 1.10.0-nightly* (645dd013a 2016-04-24)
- New lints: `temporary_cstring_as_ptr`, `unsafe_removed_from_name`, and `mem_forget`

0.0.63 — 2016-04-08

- Rustup to *rustc 1.9.0-nightly* (7979dd608 2016-04-07)

0.0.62 — 2016-04-07

- Rustup to *rustc 1.9.0-nightly* (bf5da36f1 2016-04-06)

0.0.61 — 2016-04-03

- Rustup to *rustc 1.9.0-nightly* (5ab11d72c 2016-04-02)
- New lint: [invalid_upcast_comparisons](#)

0.0.60 — 2016-04-01

- Rustup to *rustc 1.9.0-nightly* (e1195c24b 2016-03-31)

0.0.59 — 2016-03-31

- Rustup to *rustc 1.9.0-nightly* (30a3849f2 2016-03-30)
- New lints: [logic_bug](#) , [nonminimal_bool](#)
- Fixed: [match_same_arms](#) now ignores arms with guards
- Improved: [useless_vec](#) now warns on `for ... in vec![...]`

0.0.58 — 2016-03-27

- Rustup to *rustc 1.9.0-nightly* (d5a91e695 2016-03-26)
- New lint: [doc_markdown](#)

0.0.57 — 2016-03-27

- Update to *rustc 1.9.0-nightly* (a1e29daf1 2016-03-25)
- Deprecated lints: [str_to_string](#) , [string_to_string](#) , [unstable_as_slice](#) , [unstable_as_mut_slice](#)
- New lint: [crosspointer_transmute](#)

0.0.56 — 2016-03-23

- Update to *rustc 1.9.0-nightly* (0dcc413e4 2016-03-22)
- New lints: [many_single_char_names](#) and [similar_names](#)

0.0.55 — 2016-03-21

- Update to *rustc 1.9.0-nightly* (02310fd31 2016-03-19)

0.0.54 — 2016-03-16

- Update to *rustc 1.9.0-nightly* (c66d2380a 2016-03-15)

0.0.53 — 2016-03-15

- Add a [configuration file](#)

0.0.52

0.0.51 — 2016-03-13

- Add `str` to types considered by [len_zero](#)
- New lints: [indexing_slicing](#)

0.0.50 — 2016-03-11

- Update to *rustc 1.9.0-nightly (c9629d61c 2016-03-10)*

0.0.49 — 2016-03-09

- Update to *rustc 1.9.0-nightly (eabfc160f 2016-03-08)*
- New lints: [overflow_check_conditional](#) , `unused_label` , [new_without_default](#)

0.0.48 — 2016-03-07

- Fixed: ICE in [needless_range_loop](#) with globals

0.0.47 — 2016-03-07

- Update to *rustc 1.9.0-nightly (998a6720b 2016-03-07)*
- New lint: [redundant_closure_call](#)