

What Rust 1.80 Changes for Our Codebase

Lazy statics move into the standard library, Cargo starts checking cfg names, and match patterns gain exclusive ranges. Upgrade with `rustup update stable`.

2

NEW LAZY TYPES

1

NEW WARN-BY-DEFAULT LINT

50+

STABILIZED APIS

STABLE

```

standard input:18641: warning [p 269, 7.51]
I can't break line
-rivar-visibility[public|protecte
d|private|package]
-freplace-objc-classes
-fzero-link-gen-decls
-massign-intercept-asm-protocol
-mselector
-mstrict-selector-match
-mundeclared-selector

Language Independent Options
-fmessage-lengthn

333276800 -0700 A P O , n H S V I A ,
\ x r y f p d - w j , >
r n / serials250/tty x b p 4 [ B L Y U Y
s b + T T [ O V C [ i
ks: 0 10 V j ) t i \ A q ,
< i b v v c l 2 L u i
ld: ( 0 / r o b p T A M D < O q n L S
S - f / ) m e A B * > S
+ ( ) N A b q v c
m i v s p > n \ M A B I
> d p A d p A ( n ^ f r m b
V O v - v = B H 2 1 ' y A >
n b / 6 " ( C U I O < n

line "cmatrix -b"
ERROR: apport (pid 18449) Thu Mar 16 11:44:50
2017: debug: session gdbus call: (true,)
ERROR: apport (pid 18449) Thu Mar 16 11:44:50
2017: report /usr/crash/_usr_bin_cm
atrix.1000.crash already exists and unsem, do
ing nothing to avoid disk usage but
ERROR: apport (pid 18449) Thu Mar 16 11:44:50
2017: called for pid 18449, signal 0, core
limit 0
ERROR: apport (pid 18449) Thu Mar 16 11:44:50
2017: executable: /usr/bin/cmatrix (command
line "cmatrix -b")

20 20 20 20 151
mmmmmmmm 20 45 78 45 43 m
20 20 20 20 1
mmmmmmmm 20 20 20 20 20
7a 31 20 20 1
mmmmmmmm 63 64 66 71 20
20 65 63 66 64 66 71 20
mmmmmmmm 67 20 24 37 20
20 20 20 7c 1a 11 mm
mmmmmmmm for 20 20 20 20
20 20 20 20
mmmmmmmm 20 45 78 45
20 20 20 20 1 mm

Screen must be at least

AI: 184.7 Vi: 209.9 A-Vi: 25.123 cti: 14.229 154

CPU: [|||||] 100.0% Tasks: 150
Mem: [|||||] 633M/973M Load: 0.00
Swp: [|||||] 432M/1024M

CPU: 114% 49% 41% 91% 65% 50% 40%
55705 netcons 20 17 23952 7576 2244 6 11.5 5
47651 netcons 20 0 65812 11712 2784 6 11.0
4826 netcons 20 0 655M 25200 11909 5 11.0

```

LazyLock

Thread-safe lazy init, works in statics.

LazyCell

No sync, fits `thread_local` statics.

unexpected_cfgs

Catches typos like `feature = "crayon"`.

a..b patterns

Exclusive ranges in match arms.



Made with MagicSlides

What we can delete or simplify

■ Primary ■ Responsive ■ Acceleration

1.80 completes the move of `lazy_static` and `once_cell` functionality into `std` and tightens build configuration checks.

AREA	BEFORE 1.80	IN 1.80	ACTION
Lazy statics	<code>lazy_static</code> or <code>once_cell</code> crates	<code>std::sync::LazyLock</code>	Replace crate usage
One-time init	<code>OnceLock::get_or_init</code> at call site	Init function lives in the cell	Move init into <code>LazyLock::new</code>
cfg typos	Silently compiled out	<code>unexpected_cfgs</code> warning	Fix or declare in <code>[lints]</code>
Range patterns	Needed <code>K - 1</code> constants	<code>..K</code> , <code>K..M</code> , <code>M..G</code>	Use shared constants

Custom `cfg` names can be declared in the `[lints]` table of `Cargo.toml`.

MIGRATION

Made with MagicSlides



STABILIZED APIS

Small std additions worth knowing

Beyond the headline features, 1.80 stabilizes a long list of APIs.

These four groups are the ones most likely to show up in everyday code.

Checked splits

`split_at_checked` on slices and `str` returns `None` instead of panicking.

`Option::take_if`

Takes the value out only when a predicate matches.

ASCII trimming

`str::trim_ascii`, `trim_ascii_start` and `trim_ascii_end`, also on `[u8]`.

IP bit conversions

`Ipv4Addr` and `Ipv6Addr` gain `BITS`, `to_bits` and `from_bits`.

Made with MagicSlides

From OnceCell to LazyLock

Each step below is taken from the Rust 1.80.0 announcement and the releases it references.



RUST 1.70

OnceCell and OnceLock

One-time initialization types reach stable.



RUST 1.79

-check-cfg in rustc

The compiler stabilizes cfg checking.



RUST 1.80

LazyCell and LazyLock

Lazy init with the function inside the cell.



CARGO 1.80

cfg checks on by default

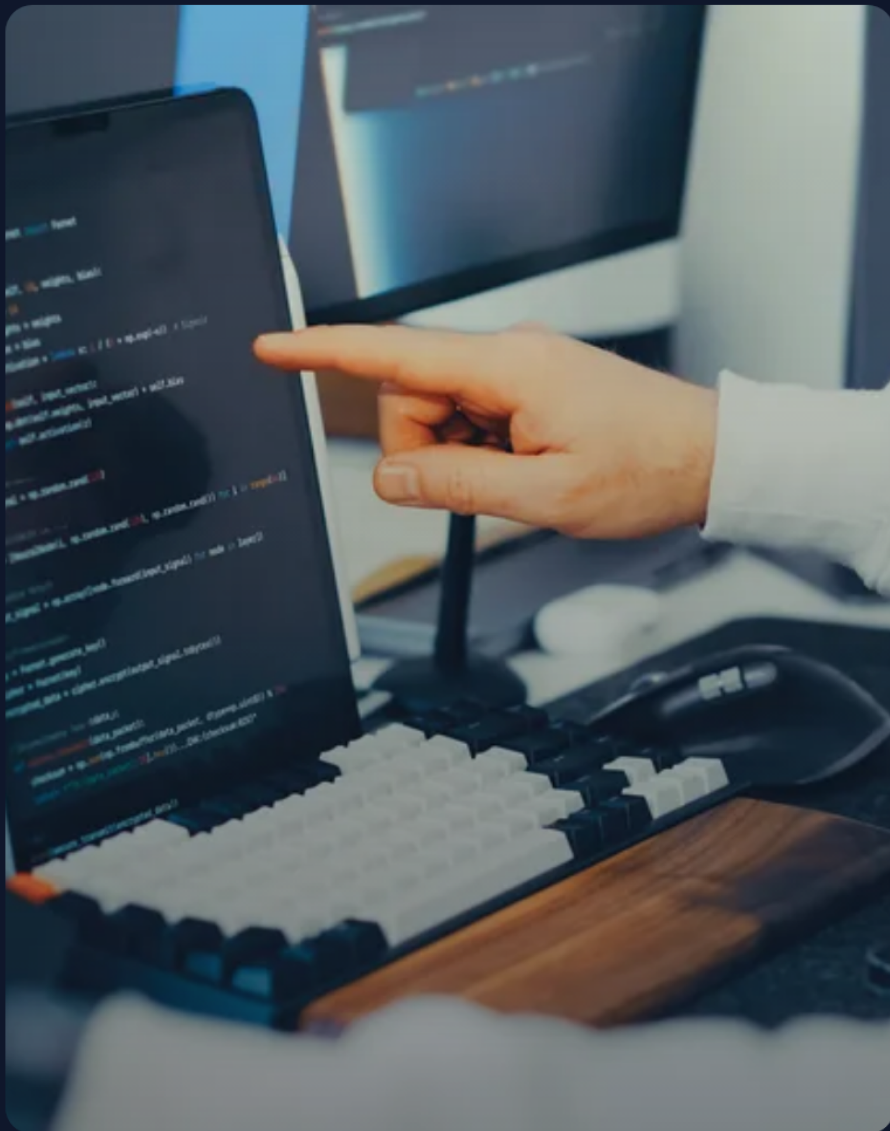
Feature names and build script cfgs are checked.



RUST 1.80

Exclusive range patterns

New lints flag gaps and overlapping endpoints.



UPGRADE CHECKLIST

Four steps for this repo

A short plan to adopt 1.80 without surprises, based only on what the release changes.

Update toolchain

Run `rustup update stable` and bump the version in CI.

Clear cfg warnings

Fix typos or list custom cfgs under `[lints.rust]`.

Swap lazy crates

Move `lazy_static` blocks to `std::LazyLock`.

Review range arms

Drop `K - 1` constants and heed the new range lints.

Made with MagicSlides