



Products

Why 'static' lifetime needed for owned variable by cloning

[Login](#) [Sign up](#)

[Ask Question](#)

Asked 2 years, 7 months ago

Active 1 year, 9 months ago

Viewed 503 times



2



I struggle a little with rust lifetimes: why is 'static' lifetime needed for `run_trait_test` ?.

I want it to behave like `run_concrete_test`

I read the documentation, searched stackoverflow and the rust-book but I'm obviously missing something.

```
use std::thread;

#[derive(Debug, Clone)]
struct Test {
    test_string: String,
}

trait Example {
    fn tst(&self) -> ();
}

impl Example for Test {
    fn tst(&self) {
        println!("{}", self);
    }
}

// compiles, no 'static here
fn run_concrete_test(tester: &Test) {
    let t = tester.clone();
    thread::spawn(move || {
        t.tst();
    });
}

// compiles with 'static
// but F shouldn't be only static
fn run_trait_test<F>(tester: &'static F)
where
    F: Example + Sync + Send + 'static,
{
    let t = tester.clone();
    let store_t = thread::spawn(move || {
        t.tst();
    });
}
```

rust

[Share](#)

[Improve this question](#)

[Follow](#)

edited Mar 18 '20 at 9:21



[Ömer Erden](#)

5,861 ● 5 ● 25 ● 39

asked May 28 '19 at 6:53



[murphy](#)

499 ● 4 ● 12

doc.rust-lang.org/std/thread/fn.spawn.html that's why

– [hellow](#)

May 28 '19 at 7:11

@hellow so what's the difference between the trait ref parameter and the Test ref parameter? Is the Test ref implicitly 'static?

– [murphy](#)

May 28 '19 at 7:29

You're cheating ;) You use `tester.clone()` and that's why it works without the 'static . If you remove that and move `tester` directly into the closure you will get the same error: play.rust-lang.org/

– [hellow](#)

May 28 '19 at 7:39

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).
@hellow sort of...:) Currently my effort goes mainly in the effort to get the code running and to understand how things work. I just tried to get the concrete implementation to a more general one.
just hoped that I could move from `run_concrete_test` to `run_trait_test` without changes to the function body.

Accept all cookies [murphy](#) [Customize settings](#)

May 28 '19 at 7:46

Add a comment

1 Answer

[Active](#) [Oldest](#) [Votes](#)



3



[From reference](#) :

The following traits are implemented for all `&T`, regardless of the type of its referent:

- ...
- `Clone` (Note that this will not defer to `T`'s `Clone` implementation if it exists!)
- ...

Since `F` has 'static' lifetime boundary, `&F` is a type of shared variable:

- Without `Clone` boundary compiler will use `clone` from borrowed version of `F` (Simply it would just clone the reference)
- With `Clone` boundary, compiler will use the implementation of `F`'s `Clone`

With all of these the code below will work as same with `run_concrete_test` :

```
trait Example {
    fn test(&self);
}

fn _run_trait_test<F>(tester: &F)
where
    F: Example + Sync + Send + Clone + 'static,
{
    let t = tester.clone();

    ::std::thread::spawn(move || {
        t.test();
    });
}
```

Note: Without 'static' lifetime boundary it is possible that `F` could be considered as some `T`'s borrowed type like `F = &T`

[Playground](#)

[Share](#)

[Improve this answer](#)

[Follow](#)

edited Mar 18 '20 at 9:18

answered May 28 '19 at 8:57



[Omer Erden](#)

5,861 ● 5 ● 25 ● 39

Add a comment

Your Answer

Post Your Answer



By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [rust](#) or [ask your own question](#).

The Overflow Blog

- [Sequencing your DNA with a USB dongle and open source code](#)
- [Don't push that button: Exploring the software that flies SpaceX rockets and...](#)

Featured on Meta

-  Providing a JavaScript API for userscripts
-  Congratulations to the 59 sites that just left Beta

Related

Why are explicit lifetimes needed in Rust?

"the type does not fulfill the required lifetime" when using a method in a thread

How to create a Quickcheck Arbitrary of a struct containing a reference?

5
Is there a CloneMut trait?






Lifetime constraints to model scoped garbage collection

Store data that implements a trait in a vector

Understanding lifetimes: max lifetime and 'static

Is there a way of spawning new threads with copies of existing data that have a non-static lifetime?

Hot Network Questions

-  EU ETS: if within-EU flight emissions are limited to climate goals, is there still a reason not to fly as long as you can afford it?
-  Could mass timber construction techniques have been used in the past?
-  Is Elon Musk really exploiting a loophole to avoid taxes?
-  If we can get people to the moon and back, why are we so adamant that it's impossible to service James Webb at 4x that with a one way robotic vehicle?
-  Can a Pyromancer Sorcerer deal fire damage to a creature under the effect of Invulnerability?

[more hot questions](#)

 [Question feed](#)

STACK OVERFLOW

[Questions](#)
[Jobs](#)
[Developer Jobs Directory](#)
[Salary Calculator](#)
[Help](#)
[Mobile](#)

PRODUCTS

[Teams](#)
[Talent](#)
[Advertising](#)
[Enterprise](#)

COMPANY

[About](#)
[Press](#)
[Work Here](#)
[Legal](#)
[Privacy Policy](#)
[Terms of Service](#)
[Contact Us](#)
[Cookie Settings](#)
[Cookie Policy](#)

STACK EXCHANGE NETWORK

[Technology](#)
[Culture & recreation](#)
[Life & arts](#)
[Science](#)
[Professional](#)
[Business](#)
[API](#)
[Data](#)

[Blog](#)
[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)