

[Skip to main content](#)

Stack Overflow

1. [About](#)
2. Products
3. [For Teams](#)
4. [Stack Internal Implement a knowledge platform layer to power your enterprise and AI tools.](#)
5. [Stack Data Licensing Get access to top-class technical expertise with trusted & attributed content.](#)
6. [Stack Ads Connect your brand to the world's most trusted technologist communities.](#)
7. [Releases Keep up-to-date on features we add to Stack Overflow and Stack Internal.](#)
8. [About the company Visit the blog](#)



Loading...

[current community](#)

- [Stack Overflow](#)
- [help chat](#)
- [Meta Stack Overflow](#)

your communities

[Sign up](#) or [log in](#) to customize your list.

[more stack exchange communities](#)

- [company blog](#)
 - 3. [Log in](#)
 - 4. [Sign up](#)
- [AI Assist is now on Stack Overflow.](#) Start a chat to get instant answers from across the network. Sign up to save and share your chats.
- [Home](#)
 - [Questions](#)
 - [AI Assist](#)
 - [Tags](#)
 - [Challenges](#)
 - 7. [Chat](#)
 - [Articles](#)
 - [Users](#)
 - [Companies](#)
 - [Collectives](#)

Communities for your favorite technologies. [Explore all Collectives](#)

Stack Internal

Stack Overflow for Teams is now called **Stack Internal**. Bring the best of human thought and AI automation together at your work.

[Try for free](#) [Learn more](#)

[Stack Internal](#)

Bring the best of human thought and AI automation together at your work. [Learn more](#)

Collectives™ on Stack Overflow

Find centralized, trusted content and collaborate around the technologies you use most.

[Learn more about Collectives](#)

[Stack Internal](#)

Knowledge at work

Bring the best of human thought and AI automation together at your work.

[Explore Stack Internal](#)

[Cloning and moving traits](#)

[Ask Question](#)

Asked 10 years, 3 months ago

Modified [10 years, 3 months ago](#)

Viewed 79 times

0

I'd like to be able to move a cloned version of an object that implements trait `Foo` into various threads. I cannot seem to figure out how this is achieved. I've tried cloning and moving a `Box<Foo>` and specifying the parameter as a generic, but cannot seem make the compiler happy.

My first attempt:

```
use std::thread;

pub trait Foo {
    fn bar();
}

pub struct Thing;

impl Thing {
    pub fn something(handler: Box<Foo>) {
        let handler_1 = handler.clone();
        thread::spawn(move || {
            Thing::another_thread(handler_1)
        });
    }

    fn another_thread(handler: Box<Foo>) { }
}
```

Resulted in the following errors:

```
error: no method named clone found for type Box<Foo + 'static> in the current scope
error: the trait core::marker::Send is not implemented for the type Foo [E0277]
```

Next I tried writing it as a generic parameter...

```
use std::thread;

pub trait Foo {
    fn bar();
}

pub struct Thing;

impl Thing {
    pub fn something<T: Foo + Clone + Send>(handler: T) {
        let handler_1 = handler.clone();
        thread::spawn(move || {
            Thing::another_thread(handler_1)
        });
    }

    fn another_thread<T: Foo>(handler: T) { }
}
```

Received the following error:

```
error: the parameter type T may not live long enough [E0310]
help: consider adding an explicit lifetime bound T: 'static...
```

Now we've reached the point where I'm lost and I start poking `<'a>` into every crevice hoping it solves the problem. But, unfortunately, I have no idea what the syntax is for what I'm trying to achieve. Thanks in advance for any help!

- [multithreading](#)
- [rust](#)
- [clone](#)
- [traits](#)
- [lifetime](#)

[Share](#)

[Improve this question](#)

Follow

asked Aug 27, 2015 at 19:50

[nathansizemore](#)

3,22888 gold badges 4444 silver badges 6767 bronze badges

[Add a comment](#) |

1 Answer

Sorted by: [Reset to default](#)

[Highest score \(default\)](#) [Trending \(recent votes count more\)](#) [Date modified \(newest first\)](#) [Date created \(oldest first\)](#)

1

consider adding an explicit lifetime bound T: 'static...

That looks like this:

```
pub fn something<T: 'static + Foo + Clone + Send>(handler: T)
```

which compiles.

In case you are unfamiliar with this syntax, it means that whatever concrete type of T is picked, it must outlive the 'static lifetime. This includes any interior references / lifetimes the concrete type may have.

[Share](#)

[Improve this answer](#)

Follow

[edited Aug 27, 2015 at 20:37](#)

answered Aug 27, 2015 at 20:17

[Shepmaster](#)

439k116116 gold badges 1.3k1.3k silver badges 1.5k1.5k bronze badges

Sign up to request clarification or add additional context in comments.

Comments

[Add a comment](#)

Your Answer

Thanks for contributing an answer to Stack Overflow!

- Please be sure to *answer the question*. Provide details and share your research!

But *avoid* ...

- Asking for help, clarification, or responding to other answers.
- Making statements based on opinion; back them up with references or personal experience.

To learn more, see our [tips on writing great answers](#).

[Sign up](#) or [log in](#)

Sign up using Google

Sign up using Email and Password

Submit

Post as a guest

Name

Email

Required, but never shown

Post as a guest

Name

Email

Required, but never shown

Post Your Answer Discard

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

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

- [multithreading](#)
- [rust](#)
- [clone](#)
- [traits](#)
- [lifetime](#)

See similar questions with these tags.

- The Overflow Blog
[Introducing Stack Overflow AI Assist—a tool for the modern developer](#)
[Treating your agents like microservices](#)
- Featured on Meta
[Chat room owners can now establish room guidelines](#)
[AI Assist is now available on Stack Overflow](#)
[Policy: Generative AI \(e.g., ChatGPT\) is banned](#)

Related

[0](#)

[Lifetimes in traits](#)

[1](#)

[Trait type and lifetime issues](#)

[1](#)

[Arc lifetime does not live long enough from clone\(\)](#)

[2](#)

[Lifetime issues in rust: borrowed value does not live long enough](#)

[1](#)

[Problem with lifetimes and implementing Clone](#)

[1](#)

[Rust lifetime confusion with custom clone function](#)

[1](#)

[conflicting lifetime requirements on implementing Clone for simple struct](#)

2

[life time of a trait object reference](#)

2

[Can Clone be implemented for a trait object with finite lifetime \(without using unsafe code\)?](#)

0

[Struggling with Clone on Trait](#)

Hot Network Questions

- [Correct website for Thailand ETA application form](#)
- [Ping sweep of IP subnets](#)
- [Looking for a proper formal substitute for "quick fix" for formal letter](#)
- [Unity HDRP custom pass stencil buffer not working in build mode](#)
- [The first Pontryagin class of Milnor's example of exotic 7-sphere](#)
- [Can I Automatically List Supported Kernel Parameters in GRUB?](#)
- [How does supporting "One China" work without also supporting Taiwan-China reunification?](#)
- [EEPROM Decoupling Capacitor](#)
- [Can four points in a unit square have mutual distances all larger than 1?](#)
- [Generating Irregular, Separated Polygonal Cells along a Curve in Geometry Nodes](#)
- [Curve has weird bumps when triangulated](#)
- [Finding real variable where a 2x2 complex matrix is singular in Mathematica](#)
- [How to use Bézier curves in 'TikZ' to draw a quotation mark?](#)
- [Is Brian Ripley's Pattern Recognition and NN book still an insightful read?](#)
- [How to handle Expedited Retreat with the optional chase rules](#)
- [Is there any way to restore hearts without consuming anything, without hot springs, and without shrines?](#)
- [The state of icy worlds during differentiation and accretion, and how they survive with their water](#)
- [When did "doubt" meaning question or point of uncertainty fall out of use outside of Indian English?](#)
- [variable name change causes segfault](#)
- [Would it be plausible to have a flat eye with a moving pupil?](#)
- [Who is Patrick and why is he referred to at anti-AfD demonstrations?](#)
- [How to protect 100-year-old wooden floors for the next 100 years?](#)
- ["Creatures your opponents control" after the ability is activated](#)
- [Storm Sphere and Obstruction](#)

[more hot questions](#)

[Question feed](#)

Subscribe to RSS

[Question feed](#)

To subscribe to this RSS feed, copy and paste this URL into your RSS reader.

Stack Overflow

- [Questions](#)
- [Help](#)
- [Chat](#)

Business

- [Stack Internal](#)
- [Stack Data Licensing](#)
- [Stack Ads](#)

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](#)

Site design / logo © 2025 Stack Exchange Inc; user contributions licensed under [CC BY-SA](#). rev 2025.12.4.37651