



Products

How to return a reference to something inside a moved box?

[Log in](#) [Sign up](#)

[Ask Question](#)

Asked 1 year, 4 months ago

Active 1 year, 4 months ago

Viewed 192 times



1



I want to store a Box in a map, and take the address of the contents of the Box and hand it out to C code. This is why I don't just store the value directly in the map, as map values move when a map grows.

I'd like a function that returns a reference to the contents of the Box, that I can use in Rust code and either use as-is in Rust code, or convert the returned reference to a raw pointer for C.

This code illustrates what I'm trying to do, but it doesn't work for obvious reasons. However, it's not clear to me how to fix it.

```
use std::collections::HashMap;

fn box_and_ref<A>(map: &mut HashMap<String, Box<Vec<u8>>>)-> &A Vec<u8> {
    let v = vec![b'h', b'e', b't', b'o'];
    let b = Box::new(v);
    let r = b.as_ref();
    map.insert("foo".to_string(), b);
    r
}

fn main() {
    let mut map: HashMap<String, Box<Vec<u8>>> = HashMap::new();

    let v = box_and_ref(&mut map);
    println!("{}", v);
}
```

[playground link](#)

[rust](#)

Share

[Improve this question](#)

Follow

asked Aug 7 '20 at 19:48



Eloff

19.4k • 16 • 74 • 100

Err, wouldn't you be able to fix your current compiling error with `map["foo"].as_ref()` ? After all, that's the new location of your `Box` .

– Zeta

Aug 7 '20 at 20:37

I think so, but now I have to do an unnecessary lookup of the value in the map. All I really want is the address (pointer) to the heap value that I know is in that Box. I could do this "safely" with `unsafe`. Is this just one of those times where you have to just shrug and do it the rust way?

– Eloff

Aug 7 '20 at 20:40

Also because the key is moved into the map, you need to copy the key (String in this case) as well. I'm going to try to do this with the `hashmap` entry API (I thought it was still unstable, but that's on `hashset` only).

– Eloff

Aug 7 '20 at 20:50

@Eloff most of the entry API has been stable since Rust 1.0.0, and what we have now is certainly enough for this use case.

– mcarton

Aug 7 '20 at 20:55

[Add a comment](#)

1 Answer

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



2



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](#).

[Accept all cookies](#)

[Customize settings](#)



For any "avoid double lookup" issues, use the [entry API](#).

```
fn box_and_ref<A>(map: &a mut HashMap<String, Box<Vec<u8>>>) -> &a Vec<u8> {
    let v = vec!{b'h', b'e', b'l', b'l', b'o'};
    let b = Box::new(v);
    map.entry("foo".to_string()).or_insert(b)
}
```

[Share](#)

[Improve this answer](#)

[Follow](#)

answered Aug 7 '20 at 20:54



[mcarton](#)

22.7k ● 5 ● 63 ● 75

This answers the question, but as soon as I tried to apply it to the real problem I was foiled because it seems you can't take a reference to anything in a map behind a Mutex, as the `MutexGuard` lives only as long as you hold the lock. I'm not sure what the fix is there, but I'm considering just swapping the `Box` for a raw pointer obtained from `Box::leak` and calling it a day.

– [Bloff](#)

Aug 7 '20 at 21:16

If your borrow of the box's contents were to outlive your borrow of the map, consider what happens if the map is dropped? The box (and its contents) would also be dropped, its memory deallocated and your reference would dangle. The statically verifiable safe solution is to borrow the map for the duration of your reference to the contents; a runtime verifiable safe solution would be to share ownership, e.g. with `Rc`. Otherwise you are introducing your own guarantees about lifetimes that neither the compiler nor runtime are able to verify: which is what raw pointers and unsafe are for.

– [eggyal](#)

Aug 7 '20 at 23:06

[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

[Borrow errors for multiple borrows](#)

[Method not compatible with trait with confusing error message](#)

["expected bound lifetime parameter" error when attempting to call a generic function](#)

[What is the proper way to coerce an iterator to return a value instead of a reference \(or vice versa\)?](#)

[Cloning a hashmap using a reference to another hashmap](#)

[How to construct a HashMap with boxed Fn values](#)

[How can I define Rust structure literals similar to Python's?](#)

Hot Network Questions

[Calculating mean follow up from ONLY sample size and range](#)

 Why is the light source not showing but light is being cast on the object

 Why is 20m band waterfall showing signals every 50 kHz?

 Do all observations arise from probability distributions?

 Holding entry is counted as a one turn holding?

[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](#)

site design / logo © 2021 Stack Exchange Inc; user contributions licensed under [cc by-sa](#). rev 2021.12.22.41046