

[Skip to main content](#)

[Stack Overflow](#)

1. [About](#)

2. [Products](#)

3. [For Teams](#)

1. [Stack Internal](#) Implement a knowledge platform layer to power your enterprise and AI tools.

2. [Stack Data Licensing](#) Get access to top-class technical expertise with trusted & attributed content.

3. [Stack Ads](#) Connect your brand to the world's most trusted technologist communities.

4. [Releases](#) Keep up-to-date on features we add to Stack Overflow and Stack Internal.

5. [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](#)

[How to manage lifetime of CString in a Vec?](#)

[Ask Question](#)

Asked 6 years, 3 months ago

Modified [5 years, 5 months ago](#)

Viewed 196 times

2

Is there a better or more idiomatic way to handle the lifetime of a CString in a Vec other than using `into_raw()`?

Here's my code:

```
// This fails because the CString gets dropped and then what is pointed to is invalid and invalidates required_layers
let required_layers: Vec<*const c_char> = vec!(
    CString::new("VK_LAYER_LUNARG_standard_validation").expect("CString err A").as_ptr(),
);

// This works because ss0 stays alive, but is going to be error-prone as dropping ss0 somewhere invalidates required_layers
let ss0 = CString::new("VK_LAYER_LUNARG_standard_validation").expect("CString err A");
let required_layers: Vec<*const c_char> = vec!(
    ss0.as_ptr(),
);

// This works, but now requires from_raw() to retake ownership to avoid leaks
let required_layers: Vec<*const c_char> = vec!(
    CString::new("VK_LAYER_LUNARG_standard_validation").expect("CString err A").into_raw(),
);
```

• [rust](#)

[Share](#)

[Improve this question](#)

Follow

[edited Jun 24, 2020 at 19:49](#)

[pretzelhammer](#)

15.4k1616 gold badges5656 silver badges105105 bronze badges

asked Sep 8, 2019 at 4:21

[Andrew Lentvorski](#)

31622 silver badges1717 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)

3

You can just store the actual CString objects:

```
let a = vec![
    CString::new("VK_LAYER_LUNARG_standard_validation").expect("CString err A")
];
```

And if what you actually need is a contiguous array of pointers, perhaps for passing to a C function that is expecting that sort of thing, you can create a second vector holding the pointers from the strings in the first.

```
let b: Vec<*const c_char> = a.iter().map(|cstr| {
    cstr.as_ptr()
}).collect();
```

The first vector will outlive the second, and it will properly drop the contained CString objects when it is dropped itself.

[Share](#)

[Improve this answer](#)

Follow

[edited Sep 8, 2019 at 6:33](#)

answered Sep 8, 2019 at 6:16

[Benjamin Lindley](#)

104k1111 gold badges210210 silver badges285285 bronze badges

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

1 Comment

Add a comment

jdkramhoft

[jdkramhoft](#) [Over a year ago](#)

How come "the first vector will outlive the second" ? I would have thought this was like the second example in the question (b would be invalidated if a is dropped) -- is this not possible?

2023-10-01T12:26:00.14Z+00:00

0

Reply

- [Copy link](#)

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

- [rust](#)

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

[4](#)

[How to use mutable member Vec?](#)

[4](#)

[Lifetimes when adding temporary values to vectors](#)

[1](#)

[Turning a `Vec<Rc<RefCell<T>>>` into a `&mut T`](#)

[1](#)

[trouble setting the lifetime on a new empty vector](#)

[4](#)

[Insert constructed string into Vec in Rust](#)

[0](#)

[Lifetimes in Rust when using Strings](#)

[2](#)

[Converting a `Vec<&str>` to `Vec<&CStr>` in rust](#)

[0](#)

[Required lifetime parameter on array of vec](#)

[0](#)

[Rust lifetime of string slices within a vector within a struct](#)

[0](#)

[Rust lifetime String to str in Vec loop](#)

Hot Network Questions

- [How to reduce GAM sensitivity to sparse data](#)
- [Storm Sphere and Obstruction](#)
- [Is zero-point energy a mathematical artifact?](#)
- [Can I Automatically List Supported Kernel Parameters in GRUB?](#)
- [Zero Inflated Beta Regression \(or not\)?](#)
- [I2C bus problems when using two distance sensors in Arduino](#)
- ["Creatures your opponents control" after the ability is activated](#)
- [What's the point of swapping polarity between mod slots?](#)
- [When should I use tied notes vs. the whole note duration?](#)
- [TV program where real actors interact with 2-Dimensional characters](#)
- [Schottky OR-ing diodes passing reverse voltage; is it a faulty diode?](#)
- [How does supporting "One China" work without also supporting Taiwan-China reunification?](#)
- [The first Pontryagin class of Milnor's example of exotic 7-sphere](#)
- [PSE Advent Calendar 2025 \(Day 5\): Cinnamon overdose](#)
- [How do I add a text editor to my \(debian-based\) initramfs?](#)
- [Can a 10*10 square be paved with 1*4 rectangular stone plates?](#)
- [Is there any point in sharing a scientific discovery as a "nobody"?](#)

- [When the fuel governor fails, why is autorotation the only option for landing?](#)
- ["Imagine if..." vs. "Imagine..."](#)
- [Installing root certificates by government](#)
- [Critiques of the real vs. nominal definition distinction](#)
- [Where in the world is the chicken?](#)
- [Who is Patrick and why is he referred to at anti-AfD demonstrations?](#)
- [Explicit examples separating Wasserstein distances from Jensen–Shannon divergence?](#)

[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](#)