



La version stable de Rust 1.26.2 est  
désormais disponible !

---

22 mars 2019



# Table des matières

1.	<a href="#">Introduction</a>	1
2.	<a href="#">Quoi de neuf?</a>	1
3.	<a href="#">Bug dans la règle des emprunts mutables</a>	1
4.	<a href="#">Conclusion</a>	2
5.	<a href="#">Source</a>	2
6.	<a href="#">Voir aussi</a>	3

% LA VERSION STABLE DE RUST 1.26.2 EST DÉSORMAIS DISPONIBLE! % Songbird %  
23 juin 2018

## 1. Introduction

Rust est un langage de programmation système axé sur la *sécurité*, la *rapidité* et la *concurrency*.

Pour mettre à jour votre version stable, il suffit d'exécuter la commande habituelle.

```
1 $ rustup update stable
```

Si vous ne disposez pas de rustup, vous pouvez en obtenir une copie sur [la page de téléchargement](#) du site officiel. N'hésitez pas également à consulter la [release note de la 1.26.2](#) sur GitHub!

## 2. Quoi de neuf?

De nouveaux correctifs ont été apportés, principalement axés sur les expressions (et les vérifications sous-jacentes) permises par le compilateur. Comme pour la [1.26.1](#), cette mise à jour n'a pour but que de stabiliser les fonctionnalités existantes.

## 3. Bug dans la règle des emprunts mutables

En dehors des blocs `unsafe`, le compilateur ne permet pas [deux accès en écriture sur une même ressource](#) en simultané. Toutefois, un bug est survenu, suite à un recyclage apporté par la [1.26.0](#) visant à [améliorer l'ergonomie de `match`](#), permettant ainsi de casser la règle précédemment citée.

## 4. Conclusion

Ici, par exemple, nous disposons de deux accès en écriture sur `bar` au même moment.

```
1 let mut foo = Some("foo".to_string());
2 let bar = &mut foo;
3 match bar {
4     Some(baz) => {
5         bar.take(); // Should not be permitted, as baz has a unique
6         // reference to the bar pointer.
7     },
8     None => unreachable!(),
9 }
```

En réponse à cela, la version 1.26.2 vient corriger ce problème. Désormais, la vérification est interrompue et renvoie une erreur lorsque le cas survient.

```
1 error[E0499]: cannot borrow *bar as mutable more than once at a
2   time
3   --> src/main.rs:6:9
4   5 |         Some(baz) => {
5     |             --- first mutable borrow occurs here
6   6 |             bar.take(); // Should not be permitted, as baz has a
7     |             ...
8     |             ^^^ second mutable borrow occurs here
9   ...
10  9 |     }
11     | - first borrow ends here
12 error: aborting due to previous error
```

## 4. Conclusion

En définitive, l'équipe Rust a pris la décision de publier ce patch le plus rapidement possible pour éviter d'étendre inutilement le laps de temps pendant lequel les compilateurs stables sont affectés par ce bug.

## 5. Source

[Le blog de l'équipe Rust](#) ↗

6. *Voir aussi*

## 6. Voir aussi

— La version stable de Rust `1.26.1` est désormais disponible! [↗](#)