CC BY

Zeste de savoir

La version stable de Rust 1.26.1 est désormais disponible !

22 mars 2019

Table des matières

% LA VERSION STABLE DE RUST 1.26.1 EST DÉSORMAIS DISPONIBLE ! % Songbird % 09 juin 2018

1. Introduction

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

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

```
$ 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 1.26.1 sur GitHub !

2. Quoi de neuf ?

Dans ce billet nous nous intéresserons à la version `1.26.1` qui, comme pour toutes les versions mineures, vient stabiliser ce qui a été apporté dans les mises à jour précédentes.

3. RLS n'interfère plus avec les commandes de build

La version de RLS fournie avec la `1.26.0` utilisait le même répertoire de sortie que Cargo en ligne de commande, ce qui occasionnait une recompilation intégrale du projet si vous utilisiez les deux à tour de rôle. Le problème était d'autant plus gênant sous Windows, puisque RLS (ou le compilateur) ne relâchait pas le verrou qu'il avait sur les fichiers à traiter. Le bug n'a pas encore été fixé mais ce dernier survient moins souvent grâce à un premier correctif.

4. Mauvais formatage de rustfmt

Précédemment, rustfmt indentait excessivement les chaînes de caractères littérales multi-lignes. Ce problème est désormais corrigé.

5. Renvoyer quelque chose depuis `main` avec `impl Trait` ne fonctionnera plus

Jusqu'ici, le compilateur s'assurait uniquement que les services du trait `std::process::Termination` étaient implémentés par le type renvoyé. Ce comportement est, désormais, révolu puisque `rustc` n'acceptera plus que les types concrets en retour, sur la version stable. Quant à l'utilisation de `impl Termination`, elle n'est, actuellement, autorisée que dans le canal nightly.

Par exemple, ce morceau de code ne fonctionnera plus en `1.26.1` :

```
1 fn main() -> impl Copy {}
```

Le second, en revanche, ne posera aucun problème, puisque la signature de la fonction ne déclare aucun type inconnu par le biais de `impl Trait`.

```
1 fn main() -> Result<(), std::io::Error> {
2     Ok(())
3 }
```

Seuls les types concrets peuvent être déclarés dans la signature (pour le type renvoyé, à bon entendeur).

6. Le turbofish ne fonctionnera plus pour les arguments typés avec `impl Trait`

En `1.26.0`, il était possible de spécifier le type des arguments des méthodes qui utilisaient `impl Trait`. Cependant, comment le turbofish (`::<u32>`, en l'occurrence) devrait interagir avec un type qui n'a pas encore été inféré (i.e. `impl Trait`) ? En réponse à cela, l'utilisation du turbofish a été temporairement prohibée, tant que l'équipe Rust n'a pas trouvé de solution à ce problème de sémantique.

```
struct Foo;

impl Foo {
    fn bar(&self, _arg: impl Copy) {}
}

fn main() {
    Foo.bar::<u32>(0);
}
```

7. La comparaison des valeurs flottantes et constantes a été revue

La constante `std::f64::NAN`, lorsque comparée à un autre nombre flottant, était considérée systématiquement supérieure à l'autre opérande, ce qui renvoyait jusqu'ici `true`. Ce bug a également été corrigé.

D'après l'équipe Rust, bien que cela ait occasionné une modification, il est peu probable que cette dernière ait des conséquences sur le comportement des programmes.

```
use std::f64::NAN;
const FOO: bool = ::std::f64::NAN >= ::std::f64::NAN;
// On 1.26.0
assert_eq!(FOO, true);
// On 1.26.1
assert_eq!(FOO, false);
```

8. rustup devrait désormais accepter les installations partielles

Lors du cycle de développement de la `1.26.0`, des changements ont été effectués sur la manière dont la documentation de la bibliothèque standard était générée, stoppant par la même occasion

la production de la documentation des composants pour un certain nombre de plateformes. En conséquence, lorsque `rustup update` était lancée sur ces dernières, rustup refusait d'installer partiellement les composants de la chaîne. Les utilisateurs concernés devront alors exécuter `rustup install stable` au lieu d'une simple mise à jour avec `rustup update` pour que rustup ignore la documentation manquante.

Le bug a malheureusement été corrigé trop tard pour que le correctif soit inclut dans la `1.26.0` et a donc été ajouté dans le patch de la `1.26.1`.

```
$ rustup update
info: syncing channel updates for 'stable-x86_64-unknown-freebsd'
info: latest update on 2018-05-10, rust version 1.26.0 (a77568041
    2018-05-07)
error: component 'rust-docs' for 'x86_64-unknown-freebsd' is
    unavailable for download
```

9. Conclusion

10. Source

Le blog de l'équipe Rust

11. Voir aussi

— Encodez/décodez du toml avec toml-rs !