

The `#[rustc_on_unimplemented]` attribute lets you specify a custom error message for when a particular trait isn't implemented on a type placed in a position that needs that trait. For example, when the following code is compiled:

```
#![feature(rustc_attrs)]
```

```
#[rustc_on_unimplemented = "error on `{Self}` with params `<{A},{}`>`" ] // error!  
trait BadAnnotation<A> {}
```

there will be an error about `bool` not implementing `Index<u8>`, followed by a note saying “the type `bool` cannot be indexed by `u8`”.

As you can see, you can specify type parameters in curly braces for substitution with the actual types (using the regular format string syntax) in a given situation. Furthermore, `{Self}` will substitute to the type (in this case, `bool`) that we tried to use.

This error appears when the curly braces do not contain an identifier. Please add one of the same name as a type parameter. If you intended to use literal braces, use `{{` and `}}` to escape them.