

If an `impl` has a generic parameter with the `#[may_dangle]` attribute, then that `impl` must be declared as an `unsafe impl`.

Erroneous code example:

```
#![feature(dropck_eyepatch)]

struct Foo<X>(X);
impl<#[may_dangle] X> Drop for Foo<X> {
    fn drop(&mut self) { }
}
```

In this example, we are asserting that the destructor for `Foo` will not access any data of type `X`, and require this assertion to be true for overall safety in our program. The compiler does not currently attempt to verify this assertion; therefore we must tag this `impl` as `unsafe`.