

A `repr(transparent)` type was also annotated with other, incompatible representation hints.

Erroneous code example:

```
#[repr(transparent, C)] // error: incompatible representation hints
struct Grams(f32);
```

A type annotated as `repr(transparent)` delegates all representation concerns to another type, so adding more representation hints is contradictory. Remove either the `transparent` hint or the other hints, like this:

```
#[repr(transparent)]
struct Grams(f32);
```

Alternatively, move the other attributes to the contained type:

```
#[repr(C)]
struct Foo {
    x: i32,
    // ...
}
```

```
#[repr(transparent)]
struct FooWrapper(Foo);
```

Note that introducing another `struct` just to have a place for the other attributes may have unintended side effects on the representation:

```
#[repr(transparent)]
struct Grams(f32);
```

```
#[repr(C)]
struct Float(f32);
```

```
#[repr(transparent)]
struct Grams2(Float); // this is not equivalent to `Grams` above
```

Here, `Grams2` is not equivalent to `Grams` – the former transparently wraps a (non-transparent) struct containing a single float, while `Grams` is a transparent wrapper around a float. This can make a difference for the ABI.