

A struct with the representation hint `repr(transparent)` had two or more fields that were not guaranteed to be zero-sized.

Erroneous code example:

```
#[repr(transparent)]
struct LengthWithUnit<U> { // error: transparent struct needs at most one
    value: f32,             //          non-zero-sized field, but has 2
    unit: U,
}
```

Because transparent structs are represented exactly like one of their fields at run time, said field must be uniquely determined. If there are multiple fields, it is not clear how the struct should be represented. Note that fields of zero-sized types (e.g., `PhantomData`) can also exist alongside the field that contains the actual data, they do not count for this error. When generic types are involved (as in the above example), an error is reported because the type parameter could be non-zero-sized.

To combine `repr(transparent)` with type parameters, `PhantomData` may be useful:

```
use std::marker::PhantomData;

#[repr(transparent)]
struct LengthWithUnit<U> {
    value: f32,
    unit: PhantomData<U>,
}
```