Subject: ONE and assignement
Posted by mdelfede on Sat, 28 Jul 2018 07:48:30 GMT
View Forum Message <> Reply to Message

Hi,

I'm right that One<> is missing those operators?

```
template <class TT>
void       operator=(One<TT>&d) = delete;

template <class TT>
void       operator=(One<TT> const &d) = delete;
```

In an application I've noticed that if I do :

```
One<String> A = new String("pippo");
One<String> B;
B = A;
Cout() << "*A = " << *A << "\n";
Cout() << "*b = " << *B << "\n";
```

The internal pointer is copied, NOT picked, and NO COMPILER ERROR IS GENERATED, so when A and B get destroyed the pointer is
double freed and I get a violation access error.
Adding the above operators brings a compiler error, which is the correct behaviour, IMHO.

Ciao

Max