Subject: Re: moveable with assert question
Posted by mtdew3q on Mon, 16 Sep 2019 02:08:08 GMT

View Forum Message <> Reply to Message

Hi Mirek and friends:

I solved the problem. It kept telling me something about deleted. Then I switched views from the error view in U++ to the view that gives more detail.

I searched on the Internet and I found this link https://stackoverflow.com/questions/43954808/why-is-the-move-constructor-defined-and-the-assignment-operator-implicitly-delet

I checked my gcc version and then decided to tack on default to one of my methods that didn't have it.

Here is the updated code that works.

```
class Foo: Moveable<Foo>  {

    char * buffer;

public:
 Foo()=default;
    Foo(const char * c);
    Foo( const Foo&) =default;

    Foo(Foo&&) = default;
    Foo& operator=(Foo&&) = default;
    String Get(){
        return buffer;
    }
    ~Foo() {
     AssertMoveable<Foo>();
     delete buffer;
    }
};

Foo::Foo(const char * c) {
        if( c!=NULL)
        {
            buffer= new char [strlen(c) + 1];
            strcpy(buffer, c);
        } else
             buffer = NULL;
    }
```

```
CONSOLE_APP_MAIN
{
    Foo f("foo");
 Foo && r2 = std::move(f);
    Foo &  j = r2 ;

    Vector<Foo> a, b;
    b.Add(r2);

    Foo & myval = b.At(0);
    Cout() << myval.Get();


}
```

thnx : roboloki