
Subject: PROPOSAL: Monitor mutex in objects
Posted by [kohait00](#) on Thu, 15 Jul 2010 09:42:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

hi guys,

in C# there exists a nice feature,

```
lock(objectinstance)
{
//method code on now protected object
}
```

one can protect another object of being somehow accessed / modified, while oneself is treating with it. the object itself doesn't know about that and does not need to take care on providing any synchronisation features inside.

would it be possible to make something like that in U++ as well?
as far as i know, C# compiler helps here somehow..

meanwhile:

here is another maybe useful construction, came in mind while i was dealing with Mt.h..

```
#define LOCKER UPP::Mutex __locker
#define LOCKED INTERLOCKED_(__locker)
```

```
class MyClass
{
void Method()
{
LOCKED
{
//your per object locked code
}
}
};
private:
LOCKER;
}
```

it's unlike the pure INTERLOCKED, which would generate a StaticMutex in every method the LOCKED is used.

if it is sensefull...
