
Subject: Re: PROPOSAL: Monitor mutex in objects
Posted by [mirek](#) on Sun, 18 Jul 2010 07:05:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Thu, 15 July 2010 05:42hi guys,

in C# there exists a nice feature,

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

one can protect another object of beeing somehow accessed / modified, while oneself is treating with it. the object itself doesnt 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 upp as well?
as far as i know, c# compiler helps here somehow..

meanwhile:

here is another maybe usefull 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 unlicke the pure INTERLOCKED, which would generate a StaticMutex in every method the LOCKED is used.

if it is sensefull...

```
class MyClass {  
    Mutex lock;  
    void Method() {  
        INTERLOCKED_(lock) {  
            ...  
        }  
    }  
};
```

or (usually I like that more)

```
class MyClass {  
    Mutex lock;  
    void Method() {  
        Mutex::Lock ____(lock);  
        ...  
    }  
};
```

I do not think we really need more...
