

---

Subject: Re: MT and variables simple question  
Posted by [Didier](#) on Fri, 06 Jun 2014 13:48:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello Koldo,

As i said Quote: if you modify a variable using the var itself or a pointer to it: the result is the same  
==> you have to protect the variable

So if data10 or vars[10] are accessed by several threads you have to put :  
INTERLOCKED {  
    data10 = &vars[10];  
}

BUT ... you have to be careful which INTERLOCK method to use:

INTERLOCKED : uses local static mutex which only protects this code (and only this one) from being accessed by several threads at same time.

INTERLOCKED(mutex) : uses the parameter mutex which allows you to protect several parts of code by calling INTERLOCKED(mutex) each time

If you only wan't to protect variables, then you probably need to use INTERLOCKED(mutex) everywhere the variable is accessed.

Here is what I mean:

```
// If you need to protect data10
Mutex mtx;
..
..
// in code of first thread
INTERLOCKED(mtx) {
    data10 = &vars[10];
}
..
..
// somewhere in another thread's code
INTERLOCKED(mtx) {
    data10 = xxxxx;
}
```

Hope I am clear enough :?

---