
Subject: Re: Intentionally inconsistent use of "Atomic" in Mt.h?

Posted by [mirek](#) on Wed, 20 Feb 2008 18:39:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Werner wrote on Sat, 16 February 2008 11:36luzr wrote on Sat, 16 February 2008 17:03Well, but you should not directly assign Atomic to int, you should rather use AtomicRead...

A misunderstanding? If so, sorry for expressing the issue insufficiently!

I'm talking about the all the "Atomic..." functions in Mt.h, i. e.

AtomicDec,
AtomicInc,
AtomicRead,
AtomicWrite,
AtomicXAdd.

They all have an "Atomic&" as parameter but treat it with an "int" and/or return an "int".

First of all I think this is miniscule problem... not worthy of such detailed examination.

Anyway, notice this: The set of functions is complete. You never need to assign int directly to Atomic or back.

Imagine some platform that e.g. lacks interlocked operations. In that case you would need to simulate Atomic using mutex:

```
struct Atomic { Mutex lock; int value; }
```

There are no direct conversions to/from Atomic now... But using AtomicRead/Write and others, returning int, everything works.

Sure, things are not ideal, perhaps Atomic should rather be class from the beginning and all operations its methods....

Mirek
