Subject: GuiLock and GuiUnlock objects are of different type.
Posted by Oblivion on Sun, 07 Feb 2021 16:49:41 GMT
View Forum Message <> Reply to Message

Hi,

GuiLock is defined as a struct, and GuiUnlock as a class. I think the source code should be consistent.

My recommendation:

```
class GuiLock { // <--- changed struct to class
public:         // <--- added.
        GuiLock()  { EnterGuiMutex(); }
        ~GuiLock() { LeaveGuiMutex(); }
};

class GuiUnlock {
 int n;

public:
 GuiUnlock()  { n = LeaveGuiMutexAll(); }
 ~GuiUnlock() { EnterGuiMutex(n); }
};
```

Best regards,
Oblivion

Subject: Re: GuiLock and GuiUnlock objects are of different type.
Posted by mirek on Sun, 07 Feb 2021 16:58:39 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Sun, 07 February 2021 17:49Hi,

GuiLock is defined as a struct, and GuiUnlock as a class. I think the source code should be consistent.

My recommendation:

```
class GuiLock { // <--- changed struct to class
public:         // <--- added.
        GuiLock()  { EnterGuiMutex(); }
        ~GuiLock() { LeaveGuiMutex(); }
```

```
};

class GuiUnlock {
 int n;

public:
 GuiUnlock()  { n = LeaveGuiMutexAll(); }
 ~GuiUnlock() { EnterGuiMutex(n); }
};
```

Best regards,
Oblivion

IDK. I am using struct in almost all cases where either all members are naturally public or I do not care about hiding by private (which is 99% of time in end applications). Is that wrong? For me the difference between struct and class is the default access, nothing else...

Subject: Re: GuiLock and GuiUnlock objects are of different type.
Posted by Oblivion on Sun, 07 Feb 2021 17:02:15 GMT
View Forum Message <> Reply to Message

Quote:

IDK. I am using struct in almost all cases where either all members are naturally public or I do not care about hiding by private (which is 99% of time in end applications). Is that wrong? For me the difference between struct and class is the default access, nothing else...

Effectively, nothing is wrong there, it is solely about access type, of course. As I noted, it is just for the sake of some consistency in source code. And source code parsers (intellisense, et.c) may categorize them differently. It "may" impair the visiblity of the "lock pair".

Best regards,
Oblivion
