

---

Subject: Re: Required simple MT lesson

Posted by [dolik.rce](#) on Sat, 21 Sep 2013 17:06:30 GMT

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

---

mubeta wrote on Sat, 21 September 2013 18:01: So, maybe I have some news more.

The crash is caused by an object used for load an external dll library

```
else
{
    RLOG("3");
    commLoop = 0;
    RLOG("3.1");
    RLOGHEXDUMP((const void *)&vBusHandle, sizeof(vBusHandle));
    if(vBusHandle != 0) if(Bus().IsPortOpen(vBusHandle)) Bus().ClosePort(&vBusHandle);
    RLOG("3.2");
}
```

The Object Bus() is used only in one hide thread, of the multithread applications. But it is "declared"? "build"? using dli utility of U++.

For now, I modify the code for don't call the IsPortOpen function if the handle is 0, (and the application don't crash). But I believe that the problem is not solved.

This seems not directly related to multithreading. The error you describe seems as if you try to call the methods of Bus() before Bus().Load("something.dll") or Bus().SetLibName("something.dll") was called. If I'm looking correctly, you only call Bus().Load() inside the loop if openBus == true, while this code is in the else branch.

By the way: You can easily test if the object was loaded correctly simply by code like "if (Bus()) { ... }". Also, it seems like overkill to me to load the dll in each loop iteration, couldn't it be only done once before the loop even starts?

About the INTERLOCKED:

You should wrap it around every piece of code that might access the shared variable at the same time. It is usually safe (and often necessary) to guard all accesses to the variable. Example:

```
// global mutex protecting variable iniKeys
Mutex iniKeysMutex;

void function1() {
    int length;
    INTERLOCKED_(iniKeysMutex){
        length = iniKeys.GetPut("Application").GetPut("UsedCommDriver").GetLength();
    }
    // do something with length ...
}
```

```
void function2() {  
    String driver;  
    INTERLOCKED_(iniKeysMutex){  
        driver = iniKeys.GetPut("Application").GetPut("UsedCommDriver")  
    }  
    // do something with driver ...  
} Here you are sure that you can call function1 and function2 from any thread and even if it  
happens simultaneously, it will be safe.
```

Honza

---