
Subject: thread bug....and fix

Posted by [aftershock](#) on Tue, 30 Jul 2019 12:53:55 GMT

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

Hi,
Bug report.
It seems that
handle can be 0 at unexpected places,
So I added additional checks for that to fix.
See added lines.
Wait function got stuck at WaitForSingleObject when handle was 0.

```
int Thread::Wait()
{
    if(!IsOpen())
        return -1;
    int out;
#ifdef PLATFORM_WIN32
    dword exit;
    if(!GetExitCodeThread(handle, &exit))
        return -1;
    if(exit != STILL_ACTIVE)
        out = (int)exit;
    else
    {
        if (!IsOpen()) // ADDED THIS LINE
            return Null;
        if(WaitForSingleObject(handle, INFINITE) != WAIT_OBJECT_0)
            return Null;
        if (!IsOpen()) // ADDED THIS LINE
            return Null;
        out = GetExitCodeThread(handle, &exit) ? int(exit) : int(Null);
    }
    Detach();
#endif
#ifdef PLATFORM_POSIX
    void *thread_return;
    if(pthread_join(handle, &thread_return))
        out = Null;
    else
        out = (int)(intptr_t)thread_return;
    handle = 0;
#endif
    return out;
}
```

Subject: Re: thread bug....and fix
Posted by [mirek](#) on Wed, 31 Jul 2019 10:44:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Would be nice to show some example of "unexpected place".

Also I would rather have a copy of the whole file than unformatted text :)

Mirek

Subject: Re: thread bug....and fix
Posted by [aftershock](#) on Wed, 31 Jul 2019 10:57:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

It is where I added the lines.
GetExitCodeThread(handle....
handle was 0 in the debugger.
Application Verifier... detected those things.

Subject: Re: thread bug....and fix
Posted by [mirek](#) on Fri, 02 Aug 2019 09:34:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

aftershock wrote on Wed, 31 July 2019 12:57It is where I added the lines.
GetExitCodeThread(handle....
handle was 0 in the debugger.
Application Verifier... detected those things.

I mean, what is your code?

Somehow I fail to grasp how calling Wait with NULL handle could have happened.

To be more specific, as long as there is no catastrophic failure of client code, handle can get 0 in 3 ways:

- thread creation fails. Why do you call Wait in that case?
- Detach is called. Why do you call Wait in that case?
- no attempt to start thread (default constructor). Why do you call Wait in that case?

So I am really curious what has happened here. In other words, if above patch is really fixing something, then this requires much more careful investigation because it would mean many things are not what they should be.....

(Ideally, testcase package demonstrating the problem would be welcome).

Mirek

Subject: Re: thread bug....and fix
Posted by [aftershock](#) on Fri, 02 Aug 2019 17:28:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Could debugger display false values?

Is that class multithreaded?

Subject: Re: thread bug....and fix
Posted by [mirek](#) on Sat, 03 Aug 2019 09:23:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

aftershock wrote on Fri, 02 August 2019 19:28 Could debugger display false values?

Can happen. When in doubt, use DUMP/RDUMP/DDUMP.

Quote:

Is that class multithreaded?

Of itself, no.
