
Subject: Re: Different native pthread.h implementations

Posted by [mirek](#) on Sat, 22 Jan 2011 12:24:10 GMT

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

OK, I guess we need temporary solution quite fast, so please check if this works for you:

```
static String& SockErrorText()
{
    static StaticCriticalSection csect;
    CriticalSection::Lock lock(csect);
    static Index<uintptr_t> thread_index;
    static Array<String> thread_errors;
    static Vector<int> error_ticks;
    int t = msecs();
    if(thread_index.GetCount() >= 1000) {
        for(int i = thread_index.GetCount(); --i >= 0;)
            if(t - error_ticks[i] >= 60000) {
                thread_index.Remove(i);
                thread_errors.Remove(i);
                error_ticks.Remove(i);
            }
    }
#ifdef PLATFORM_WIN32
    uintptr_t tid = GetCurrentThreadId();
#else
#ifdef _MULTITHREADED
    uintptr_t tid = (uintptr_t)Thread::GetCurrentId();
#else
    uintptr_t tid = 0;
#endif
#endif
    int f = thread_index.Find(tid);
    if(f < 0) {
        f = thread_index.GetCount();
        thread_index.Add(tid);
        thread_errors.Add();
        error_ticks.Add();
    }
    error_ticks[f] = t;
    return thread_errors[f];
}
```

Mirek
