
Subject: Problem breaking loop (with close button) in main thread

Posted by [awksed](#) on Tue, 18 Oct 2022 15:19:00 GMT

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

Windows 7. U++ 16270. Multi-thread app.

I have a GuiAcquireMutex() function called from a main thread Paint() that calls

WaitForSingleObject(hMutex, dwShortTimeout) // dwShortTimeout = 1 second (full timeout is 60 seconds)

if the mutex is not acquired I call:

Ctrl::GuiSleep(1000) to allow other threads to run and if 60 seconds have not elapsed, jump back to WaitForSingleObject().

I wish to allow the close button to call Close() (that sets boolean g_bQuit) and allow the user close the app (breaking the mutex acquire loop).

Clicking the close button results in the windows message "... is not responding".

Adding Ctrl::ProcessEvents() causes a "WM_PAINT invoked ... while in Paint routine" error.

Is there some way in U++ to allow the close button to call Close() while the main thread is looping (like Windows PumpWaitingMessages())?

Code:

```
DWORD GuiAcquireMutex(HANDLE hMutex, DWORD dwTimeout)
{
    if(hMutex == INVALID_HANDLE_VALUE)
        return ERROR_INVALID_HANDLE;

    long long lNow;
    long long lStart      = GetMilliTime();
    long long lEnd        = lStart + (long long) dwTimeout;
    DWORD    dwShortTimeout = 1000;      // 1 second
    DWORD    dwLastError;
    DWORD    dwWaitResult;
```

Again:

```
dwLastError = 0; // Success
dwWaitResult = WaitForSingleObject(hMutex, dwShortTimeout);

switch(dwWaitResult)
{
```

```

case WAIT_FAILED:

    llnow = GetMilliTime();

    if(llnow < llen)
    {
        if(g_bQuit)
        {
            dwLastError = ERROR_COUNTER_TIMEOUT;

            SetLastError(dwLastError);
            return dwLastError;
        }

        Ctrl::GuiSleep(1000); // Never sleeps for 1000 ms (always returns immediately)
        goto Again;
    }

    dwLastError = GetLastError();
    break;

case WAIT_TIMEOUT:

    llnow = GetMilliTime();

    if(llnow < llen)
    {
        if(g_bQuit)
        {
            dwLastError = ERROR_COUNTER_TIMEOUT;

            SetLastError(dwLastError);
            return dwLastError;
        }

        Ctrl::GuiSleep(1000); // Never sleeps for 1000 ms (always returns immediately)
        goto Again;
    }

    dwLastError = ERROR_COUNTER_TIMEOUT;

    SetLastError(dwLastError);
}

return dwLastError;
}

```

Thanks.
