
Subject: Re: Thread::GetCurrentThreadId() and Thread::GetCurrentThreadHandle()

new methods

Posted by [tojocky](#) on Mon, 10 Jan 2011 14:35:40 GMT

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

Ok, Exists 2 variants:

1. The first variant, proposed by me, which calls system API:

```
class Thread : NoCopy {
#ifndef PLATFORM_WIN32
    HANDLE handle;
#endif
#ifndef PLATFORM_POSIX
    pthread_t handle;
#endif
public:
    bool Run(Callback cb);

    void Detach();
    int Wait();

    bool IsOpen() const { return handle; }

#ifndef PLATFORM_WIN32
    typedef HANDLE Handle;
    typedef DWORD Id;
#endif
#ifndef PLATFORM_POSIX
    typedef pthread_t Handle;
    typedef pthread_t Id;
#endif
    Handle GetHandle() const { return handle; }

    void Priority(int percent); // 0 = lowest, 100 = normal

    static void Start(Callback cb);

    static void Sleep(int ms);

    static bool IsST();
    static bool IsMain();
    static int GetCount();
    static void ShutdownThreads();
    static bool IsShutdownThreads();
#ifndef PLATFORM_WIN32
    static Handle GetCurrentHandle(){
        return GetCurrentThread();
    }
    static Id GetCurrentId(){

```

```

    return ::GetCurrentThreadId();
};

static inline
#endif defined(PLATFORM_POSIX)
static Handle GetCurrentHandle(){
    return pthread_self();
}
static Id GetCurrentId(){
    return pthread_self();
};
#endif

```

```

Thread();
~Thread();

```

```

private:
    void operator=(const Thread&);
    Thread(const Thread&);
};

```

2. The second variant, customized, proposed by Mirek, which need to be faster than call system API is:

```

static thread__ bool __sThreadId;
...
class Thread : NoCopy {
#ifndef PLATFORM_WIN32
    HANDLE handle;
#endif
#ifndef PLATFORM_POSIX
    pthread_t handle;
#endif
public:
    bool Run(Callback cb);

    void Detach();
    int Wait();

    bool IsOpen() const { return handle; }

#ifndef PLATFORM_WIN32
    typedef HANDLE Handle;
#endif
#ifndef PLATFORM_POSIX
    typedef pthread_t Handle;
#endif

```

```

typedef qword Id;
Handle GetHandle() const { return handle; }

void Priority(int percent); // 0 = lowest, 100 = normal

static void Start(Callback cb);

static void Sleep(int ms);

static bool IsST();
static bool IsMain();
static int GetCount();
static void ShutdownThreads();
static bool IsShutdownThreads();
#ifndef PLATFORM_WIN32
static Handle GetCurrentHandle(){
    return GetCurrentThread();
}
#endif
#ifndef PLATFORM_POSIX
static Handle GetCurrentHandle(){
    return pthread_self();
}
#endif
static inline Id GetCurrentId(){
    return (qword)(&__sThreadId);
};

Thread();
~Thread();

private:
    void operator=(const Thread&);
    Thread(const Thread&);
};

The "static thread__ bool __sThreadId" and "GetCurrentId()" method body in the second variant
can be placed in cpp file.

What is the best solution?

Thank you Mirek for Hint!

```
