
Subject: Missing GetTickCount64

Posted by [steffen](#) on Thu, 23 Oct 2014 09:50:32 GMT

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

Windows Vista and newer has got a GetTickCount64 function, I was recently making a long running application and had a use for a non overflowing timer value.

So here is a simple expansion of the existing block in Core/Utils:

```
#ifdef PLATFORM_POSIX
dword GetTickCount() {
#if _POSIX_C_SOURCE >= 199309L
    struct timespec tp;
    if (clock_gettime(CLOCK_MONOTONIC, &tp) == 0)
    {
        return (dword)((tp.tv_sec * 1000) + (tp.tv_nsec / 1000000));
    }
    return 0; // ?? (errno is set)
#else
    struct timeval tv[1];
    struct timezone tz[1];
    memset(tz, 0, sizeof(tz));
    gettimeofday(tv, tz);
    return (dword)tv->tv_sec * 1000 + tv->tv_usec / 1000;
#endif
}
```

```
qword GetTickCount64() {
#if _POSIX_C_SOURCE >= 199309L
    struct timespec tp;
    if (clock_gettime(CLOCK_MONOTONIC, &tp) == 0)
    {
        return (qword)((tp.tv_sec * 1000) + (tp.tv_nsec / 1000000));
    }
    return 0; // ?? (errno is set)
#else
    struct timeval tv[1];
    struct timezone tz[1];
    memset(tz, 0, sizeof(tz));
    gettimeofday(tv, tz);
    return (qword)tv->tv_sec * 1000 + tv->tv_usec / 1000;
#endif
}
#endif
```

Subject: Re: Missing GetTickCount64
Posted by [mirek](#) on Wed, 29 Oct 2014 08:50:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

It is a good idea, actually, I have a similar task in RM for some time now too. However, I think we might want that for WindowsXP too?

Subject: Re: Missing GetTickCount64
Posted by [Tom1](#) on Wed, 29 Oct 2014 09:27:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

This is not an easy task. I have spent days or maybe weeks over the last 20 years to get a reliable, precise monotonic timer that works. This relatively recent Microsoft article sheds some light on the task:

<http://msdn.microsoft.com/en-us/library/windows/desktop/dn553408%28v=vs.85%29.aspx>

I guess `QueryPerformanceCounter()` is the way to go, but it is slow to call and has not been entirely trouble free either. I have solved the problem using the 32 bit timer and some overflow detection solutions to build monotonic timer for longer periods. (Sleeping and hibernating might still be a problem though.)

Best regards,

Tom
