

## Cancel IO problems on Server 2003

**Source:**

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2004-04/0040.html>

---

**From:** Ron Jolly (*ron.jolly\_at\_gefanuc.com*)

**Date:** 03/31/04

Date: 31 Mar 2004 07:57:03 -0800

Hello group

I have an old monolithic driver that works OK on NT 4.0 and Win 2000 but has problems on Win Server 2003. I have included a crash dump.

Since the device must wait a long time for device interrupts to occur, the driver manages its own IRPs via standby queues. An API library contained in a DLL is used by application programs to access the device. When a user app wants to wait for a device interrupt, he calls an API call that will Open the device, create an event, start a thread via `_beginthreadex`, instigate an `WaitForMultipleObjects` and wait for an event to notify the user app when a device interrupt occurs. This all works OK on NT 4.0, Win2k and also 2003 Server. The problem occurs on 2003 server if the user app tries to cancel the operation. A stress test I wrote, will attempt to cancel the operation very quickly after creating the API thread in the DLL. If I start a single instance of the app test (IE one process with one thread) everything works fine and will run forever. I can cancel the IO operation all day long. If a second asynchronous instance of the stress test is started while the first instance is still running, a crash will occur after about 5 minutes.

As can be seen in the crash dump, the API library in the function `RFM2gReceiveFromQueue`, has called `WaitForMultipleObjects`. The `WaitForMultipleObjects` has returned with a `WAIT_OBJECT_0 + 1` (we received a cancel). We then call the `CancelIo` function from within the API DLL.

At that point the system traverses a path to a crash. I notice that one of the final system calls is always `IoAcquireCancelSpinLock`.

This is an old driver and I thought that maybe some of the old DDK APIs being used in the driver may be having problems on Server 2003. I went through the driver code and reviewed the places where I am protecting queues with my own spinlocks. I found all calls to `KeAcquireSpinLock()` and replaced them with `KeAcquireInStackQueuedSpinLock` and all of the other analogous calls such as replace `KeAcquireSpinLockAtDpcLevel` with `KeAcquireInStackQueuedSpinLockAtDpcLevel`, `KeReleaseSpinLock` with

KeReleaseInStackQueuedSpinLock etc.

This of course would be incompatible with NT 4.0, but I wanted to see if it fixed any problems on Server 2003. It did not fix the crash problem on Server 2003.

I ran PREfast on the driver and no problems were found.

The DDK used to build the driver is the 3790 version and the DLL and application is built with Visual Studio .Net 2003.

I would like to take advantage of some of the newer DDK API's (like cancel safe queues), but it must still work with NT 4.0 so I don't think I am able to use these APIs.

Are there any Hotfixes available on Windows Server 2003 that I may need?

The system is a Gigabyte GA-8IPXDR motherboard using a Dual Xeon 2.8 Ghz with 1 gigabyte of main memory. Hyperthreading is Disabled in the BIOS.

Begin crash dump

////////////////////////////////////

Symbol search path is: C:\WINDOWS\Symbols  
Microsoft (R) Windows Debugger Version 6.3.0011.2  
Copyright (c) Microsoft Corporation. All rights reserved.

Loading Dump File [C:\WINDOWS\MEMORY.DMP]  
Kernel Complete Dump File: Full address space is available

Symbol search path is: C:\WINDOWS\Symbols  
Executable search path is:  
Windows Server 2003 Kernel Version 3790 MP (2 procs) Free x86 compatible  
Product: Server, suite: TerminalServer SingleUserTS  
Built by: 3790.srv03\_rtm.030324-2048  
Kernel base = 0x804de000 PsLoadedModuleList = 0x8057b6a8  
Debug session time: Tue Mar 30 15:39:57 2004  
System Uptime: 0 days 0:09:10.437  
Loading Kernel Symbols

.....  
Loading unloaded module list

...  
Loading User Symbols

.....  
\*\*\*\*\*

\*  
\*  
\* Bugcheck Analysis  
\*  
\*

\*  
\*\*\*\*\*

Use !analyze -v to get detailed debugging information.

BugCheck A, {f772, 2, 1, 8074807e}

\*\*\* WARNING: Unable to verify checksum for rfm2gdll\_std.dll  
\*\*\* ERROR: Symbol file could not be found. Defaulted to export  
symbols for MSVCR71D.dll -  
Probably caused by : ntkrnlmp.exe ( nt!KiTrap0E+224 )

Followup: MachineOwner

-----  
1: kd> !analyze -v  
\*\*\*\*\*  
\*  
\*  
\* Bugcheck Analysis  
\*  
\*  
\*  
\*\*\*\*\*

IRQL\_NOT\_LESS\_OR\_EQUAL (a)  
An attempt was made to access a pageable (or completely invalid)  
address at an  
interrupt request level (IRQL) that is too high. This is usually  
caused by drivers using improper addresses.  
If a kernel debugger is available get the stack backtrace.  
Arguments:  
Arg1: 0000f772, memory referenced  
Arg2: 00000002, IRQL  
Arg3: 00000001, value 0 = read operation, 1 = write operation  
Arg4: 8074807e, address which referenced memory

Debugging Details:

-----  
WRITE\_ADDRESS: 0000f772  
  
CURRENT\_IRQL: 2  
  
FAULTING\_IP:  
hal!KeAcquireQueuedSpinLockRaiseToSynch+3e  
8074807e 8902 mov [edx],eax  
  
DEFAULT\_BUCKET\_ID: DRIVER\_FAULT  
  
BUGCHECK\_STR: 0xA  
  
LAST\_CONTROL\_TRANSFER: from 804f4559 to 8074807e

STACK\_TEXT:

b756fcf4 804f4559 80510f0e b756fd08 813e6ab8  
hal!KeAcquireQueuedSpinLockRaiseToSynch+0x3e  
b756fcf8 80510f0e b756fd08 813e6ab8 85905bc0  
nt!IoAcquireCancelSpinLock+0x9  
b756fd0c 805d8931 858a0658 b756fd64 0a4cfcf8 nt!IoCancelIrp+0x2d  
b756fd54 804dfd24 000007d0 0a4cfcf8 00000000 nt!NtCancelIoFile+0xb7  
b756fd54 7ffe0304 000007d0 0a4cfcf8 00000000 nt!KiSystemService+0xd0  
0a4cfce4 77f4235b 77e6cd10 000007d0 0a4cfcf8  
SharedUserData!SystemCallStub+0x4  
0a4cfce8 77e6cd10 000007d0 0a4cfcf8 000007d0 ntdll!ZwCancelIoFile+0xc  
0a4cf00 10003a65 000007d0 0a4cff80 00000000 kernel32!CancelIo+0x12  
0a4cfe64 10002cab 00222538 00000001 0a4cff5c  
rfm2gDll\_stdcl!RFM2gReceiveFromQueue+0x475

[c:\vsnet\rtbuild\rfm2gDll\_stdcl\rfm2gDll\_stdcl.c @ 2962]  
0a4cff80 10203266 10009288 00000000 00000000  
rfm2gDll\_stdcl!RFM2gCallbackDispatcher+0x7b

[c:\vsnet\rtbuild\rfm2gDll\_stdcl\rfm2gDll\_stdcl.c @ 2232]  
WARNING: Stack unwind information not available. Following frames may  
be wrong.

0a4cffb8 77e4a990 00225590 00000000 00000000  
MSVCR71D!beginthreadex+0x196  
0a4cffe8 00000000 102031b0 002254d8 00000000  
kernel32!BaseThreadStart+0x34

FOLLOWUP\_IP:

nt!KiTrap0E+224  
804e2f58 833d40a3578000 cmp dword ptr [nt!KiFreezeFlag  
(8057a340)],0x0

SYMBOL\_STACK\_INDEX: 1

FOLLOWUP\_NAME: MachineOwner

SYMBOL\_NAME: nt!KiTrap0E+224

MODULE\_NAME: nt

IMAGE\_NAME: ntkrnlmp.exe

DEBUG\_FLR\_IMAGE\_TIMESTAMP: 3e8015c6

STACK\_COMMAND: .trap ffffffff756fc80 ; kb

BUCKET\_ID: 0xA\_W\_nt!KiTrap0E+224

Followup: MachineOwner

-----  
1: kd> ln 804f4559

microsoft.public.development.device.drivers: Cancel IO problems on Server 2003

```
(804f4550) nt!IoAcquireCancelSpinLock+0x9 | (8053aeb0)
nt!IoAcquireVpbSpinLock
1: kd> ln 8074807e
(80748040) hal!KeAcquireQueuedSpinLockRaiseToSynch+0x3e |
(80748090) hal
```

KeReleaseInStackQueuedSpinLock

```
////////////////////////////////////
End crash dump
```

Any feedback welcome

Thanks  
Ron Jolly  
VMIC – GE Embedded Systems