

# Re: Illegal instruction – code c000001d (!!! second chance !!!)

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2008-01/msg00103.1>

---

- *From:* "Doron Holan [MSFT]" <[doronh@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:doronh@xxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 4 Jan 2008 14:27:33 -0800
- 

are you stopping the WDFUSBPIPE (WdfIoTargetStop) on unload? what is the output of !wdflogdump <your driver>?

d

--

Please do not send e-mail directly to this alias. this alias is for newsgroup purposes only.

This posting is provided "AS IS" with no warranties, and confers no rights.

<[vijayaraju.k@xxxxxxxxxx](mailto:vijayaraju.k@xxxxxxxxxx)> wrote in message

<news:c33eb531-b389-45df-bc95-051d8735a4d4@xx>

On Dec 28, 1:54 am, vijayaraj...@xxxxxxxxxx wrote:

On Dec 28, 1:29 am, "Ivan Brugiolo [MSFT]"

<[ivanb...@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:ivanb...@xxxxxxxxxxxxxxxxxxxxxxxx)> wrote:

- > I guess you need NOT to leave anything (IRPs, WorkItems, DPCs, etc)
- > outstanding before unloading.
- > As far as the check-vs-free, my take is that it is timing.
- > Maybe the outstanding IRP below completes before you get unloaded,
- > because the machine/driver is slower.
- > Or, you get lucky to get reloaded to the same address,
- > and, you never notice the problem.

> --

> --

- > This posting is provided "AS IS" with no warranties, and confers no > rights.
- > Use of any included script samples are subject to the terms specified >
- > at <http://www.microsoft.com/info/copyright.htm>



Re: Illegal instruction – code c000001d (!!! second chance !!!)

>>> Use !analyze -v to get detailed debugging information.

>>> BugCheck FC, {f88d4ddc, 2eed163, f88d4d64, 0}

>>> Probably caused by : Wdf01000.sys ( Wdf01000!

>>> FxRequestBase::CompleteSubmitted+89 )

>>> Followup: MachineOwner

>>> -----

>>> kd> !analyze -v

>>>

\*\*\*\*\*\_\*\*

>>> \*

>>> \*

>>> \* Bugcheck

>>> Analysis \*

>>> \*

>>> \*

>>>

\*\*\*\*\*\_\*\*

>>> ATTEMPTED\_EXECUTE\_OF\_NOEXECUTE\_MEMORY (fc)

>>> An attempt was made to execute non-executable memory. The guilty

>>> driver

>>> is on the stack trace (and is typically the current instruction

>>> pointer).

>>> When possible, the guilty driver's name (Unicode string) is printed >>> on

>>> the bugcheck screen and saved in KiBugCheckDriver.

>>> Arguments:

>>> Arg1: f88d4ddc, Virtual address for the attempted execute.

>>> Arg2: 02eed163, PTE contents.

>>> Arg3: f88d4d64, (reserved)

>>> Arg4: 00000000, (reserved)

>>> Debugging Details:

>>> -----

>>> DEFAULT\_BUCKET\_ID: DRIVER\_FAULT

>>> BUGCHECK\_STR: 0xFC

>>> PROCESS\_NAME: System

>>> TRAP\_FRAME: f88d4d64 --- (.trap 0xfffffffff88d4d64)

>>> ErrCode = 00000011

>>> eax=c0000016 ebx=00000000 ecx=00000000 edx=00000002 esi=7dfe1fe8

>>> edi=8233b8fc

>>> eip=f88d4ddc esp=f88d4dd8 ebp=824b76d0 iopl=0 nv up ei pl zr

>>> na pe nc

Re: Illegal instruction – code c000001d (!!! second chance !!!)

Re: Illegal instruction – code c000001d (!!! second chance !!!)

```
>>> cs=0008 ss=0010 ds=0023 es=0023 fs=0030 gs=0000
>>> efl=00010246
>>> f88d4ddc 044e add al,4Eh
>>> Resetting default scope

>>> LAST_CONTROL_TRANSFER: from 8051c0d3 to 804f8aef

>>> STACK_TEXT:
>>> f88d4cec 8051c0d3 000000fc f88d4ddc 02eed163 nt!KeBugCheckEx+0x1b
>>> f88d4d4c 8053f90c 00000008 f88d4ddc 00000000 nt!MmAccessFault+0x8e7
>>> f88d4d4c f88d4ddc 00000008 f88d4ddc 00000000 nt!KiTrap0E+0xcc
>>> WARNING: Frame IP not in any known module. Following frames may be
>>> wrong.
>>> f88d4ddc aa030245 7db48928 7db3f600 8233b8fc 0xf88d4ddc
>>> f88d4e04 aa0075d3 827aa51b 824c09f8 00000000 Wdf01000!
>>> FxRequestBase::CompleteSubmitted+0x89
>>> f88d4e20 aa00768d 014b76d0 8236e928 f88d4e4c Wdf01000!
>>> FxIoTarget::RequestCompletionRoutine+0x195
>>> f88d4e30 804ef5ed 00000000 827aa460 824b76d0 Wdf01000!
>>> FxIoTarget::_RequestCompletionRoutine+0x35
>>> f88d4e4c 804f054c 00000000 827aa460 8236e928 >>> nt!IopUnloadSafeCompletion
>>> +0x1d
>>> f88d4e7c f7f51ee5 827aa460 81c0fe10 82750028 nt!IopfCompleteRequest
>>> +0xa2
>>> f88d4ee4 f7f52b57 81fee4d8 00000000 827507d8 USBPORT!
>>> USBPORT_CompleteTransfer+0x373
>>> f88d4f14 f7f53754 026e6f44 827500e0 827500e0 USBPORT!
>>> USBPORT_DoneTransfer+0x137
>>> f88d4f4c f7f54f6a 82750028 80541ac8 82750230 USBPORT!
>>> USBPORT_FlushDoneTransferList+0x16c
>>> f88d4f78 f7f62fb0 82750028 80541ac8 82750028 >>>
USBPORT!USBPORT_DpcWorker
>>> +0x224
>>> f88d4fb4 f7f63128 82750028 00000001 82489008 USBPORT!
>>> USBPORT_IsrDpcWorker+0x37e
>>> f88d4fd0 80540f7d 8275064c 6b755044 00000000 USBPORT!USBPORT_IsrDpc
>>> +0x166
>>> f88d4ff4 80540c4a f8930c18 00000000 00000000 nt!KiRetireDpcList+0x46
>>> f88d4ff8 f8930c18 00000000 00000000 00000000 nt!KiDispatchInterrupt
>>> +0x2a
>>> 80540c4a 00000000 00000009 bb835675 00000128 0xf8930c18

>>> STACK_COMMAND: kb

>>> FOLLOWUP_IP:
>>> Wdf01000!FxRequestBase::CompleteSubmitted+89
>>> aa030245 eb0c jmp Wdf01000!
>>> FxRequestBase::CompleteSubmitted+0x97 (aa030253)

>>> SYMBOL_STACK_INDEX: 4
```

Re: Illegal instruction – code c000001d (!!! second chance !!!)

>>> SYMBOL\_NAME: Wdf01000!FxRequestBase::CompleteSubmitted+89  
>>> FOLLOWUP\_NAME: MachineOwner  
>>> MODULE\_NAME: Wdf01000  
>>> IMAGE\_NAME: Wdf01000.sys  
>>> DEBUG\_FLR\_IMAGE\_TIMESTAMP: 4549b23a  
>>> FAILURE\_BUCKET\_ID: 0xFC\_Wdf01000!FxRequestBase::CompleteSubmitted+89  
>>> BUCKET\_ID: 0xFC\_Wdf01000!FxRequestBase::CompleteSubmitted+89  
>>> Followup: MachineOwner  
  
>>> Can anybody help me in this regard? I don't know How to debug >>> further  
>>> and what to do next as the error is not in my driver?  
>>> Please somebody help me to get rid of this problem, It is very >>> urgent  
>>> for me.  
>>> Your valuable suggestions are always appreciable.  
>>> Thanks in advance.

>>> ~Vijji– Hide quoted text –

>> – Show quoted text –

> Hi Ivan,  
> Thanks for the Immediate reply.  
> You are absolutely correct. My driver gets reloaded because, I have  
> firmware download in my driver hence I need to Recycle/Reload the  
> driver after downloading the firmware by using  
> "WdfUsbTargetDeviceCyclePortSynchronously".

> So what do you think I can do to fix this issue? Why is it effects  
> only in Free Build of the driver?  
> Thnaks in advance.

> ~Vijji– Hide quoted text –

> – Show quoted text –

Hi Ivan,  
Thats verymuch reasonable. I understood about Check–VS–Free build  
matter clearly now.  
But I din't get you on these following lines  
"I guess you need NOT to leave anything (IRPs, WorkItems, DPCs, etc)  
outstanding before unloading."  
What do you mean by that? Something needs to be changed in driver?  
Usually at the time of Firmware download and Power Policy Setting I am  
calling the "WdfUsbTargetDeviceCyclePortSynchronously" routine This

Re: Illegal instruction – code c000001d (!!! second chance !!!)

routine causes my driver gets unloaded by calling surprise removal and then restart/reenumerate the device by calling DriverEntry routine. When it calls surprise remove routine consequently Its also calls ReleaseHardware routine to remove everything that was prepared/started at the time of PrepareHardware. I think nothing remains in the driver,everything will be re-created and re-assigned I hope. Can you please help in understanding and resolving this Issue? How can you solve this problem? Suggest me to solve this problem.

~Vijji– Hide quoted text –

– Show quoted text –

Hi Ivan,

I loaded Checked Build Driver first and uninstalled It then I removed sys and inf files then I reloaded Release Build Driver This time also it failed to load the driver with a different crash dump. This time crash dump is as follows: I just wanted to tell you that, This time My driver Did not loaded Twice as the firmware already running in device. I got all Descriptor values correctly. Still My driver broke/crashed at the same point in Code.

```
*****
*
*
* Bugcheck
Analysis *
*
*
*****
```

Use !analyze -v to get detailed debugging information.

BugCheck 7F, {d, 0, 0, 0}

Probably caused by : Wdf01000.sys ( Wdf01000! FxRequestBase::CompleteSubmitted+89 )

Followup: MachineOwner

-----

nt!RtlpBreakWithStatusInstruction:

80526fc8 cc int 3

kd> !analyze -v

```
*****
*
*
* Bugcheck
Analysis *
```

Re: Illegal instruction – code c000001d (!!! second chance !!!)

Re: Illegal instruction – code c000001d (!!! second chance !!!)

\*  
\*

\*\*\*\*\*

UNEXPECTED\_KERNEL\_MODE\_TRAP (7f)

This means a trap occurred in kernel mode, and it's a trap of a kind that the kernel isn't allowed to have/catch (bound trap) or that is always instant death (double fault). The first number in the bugcheck params is the number of the trap (8 = double fault, etc) Consult an Intel x86 family manual to learn more about what these traps are. Here is a \*portion\* of those codes:

If kv shows a taskGate

use .tss on the part before the colon, then kv.

Else if kv shows a trapframe

use .trap on that value

Else

.trap on the appropriate frame will show where the trap was taken

(on x86, this will be the ebp that goes with the procedure

KiTrap)

Endif

kb will then show the corrected stack.

Arguments:

Arg1: 0000000d, EXCEPTION\_GP\_FAULT

Arg2: 00000000

Arg3: 00000000

Arg4: 00000000

Debugging Details:

-----

BUGCHECK\_STR: 0x7f\_d

DEFAULT\_BUCKET\_ID: DRIVER\_FAULT

PROCESS\_NAME: services.exe

LAST\_CONTROL\_TRANSFER: from 804f79d7 to 80526fc8

STACK\_TEXT:

f8876888 804f79d7 00000003 f8876be4 00000000 nt!

RtlpBreakWithStatusInstruction

f88768d4 804f85c4 00000003 f8876dc2 7e3f5fe8 nt!KiBugCheckDebugBreak  
+0x19

f8876cb4 805401ef 0000007f 0000000d 00000000 nt!KeBugCheck2+0x574

f8876cb4 f8876dc2 0000007f 0000000d 00000000 nt!KiSystemFatalException  
+0xf

WARNING: Frame IP not in any known module. Following frames may be wrong.

f8876d4c f7167245 7dcd2b40 7dd0d0b8 81f9ce04 0xf8876dc2

Re: Illegal instruction – code c000001d (!!! second chance !!!)

Re: Illegal instruction – code c000001d (!!! second chance !!!)

f8876d74 f713e5d3 82b1af20 822f2f40 82b1afd8 Wdf01000!  
FxRequestBase::CompleteSubmitted+0x89  
f8876d90 f713e68d 0132d4b8 81bf95d8 f8876dbc Wdf01000!  
FxIoTarget::RequestCompletionRoutine+0x195  
f8876da0 804ef5ed 00000000 82b1af20 8232d4b8 Wdf01000!  
FxIoTarget::\_RequestCompletionRoutine+0x35  
f8876dbc 8064bab0 00000000 82b1af20 81bf95d8 nt!IopUnloadSafeCompletion  
+0x1d  
f8876de0 804f054c 00000000 82b1af20 f8876e44 nt!  
IovpLocalCompletionRoutine+0xb4  
f8876e10 8064bf38 82363cd0 8205a318 82b1af20 nt!IopfCompleteRequest  
+0xa2  
f8876e7c f82deee5 82b1af20 82363cd0 8269c028 nt!IovCompleteRequest  
+0x9a  
f8876ee4 f82dfb57 8205a318 00000000 8269c7d8 USBPORT!  
USBPORT\_CompleteTransfer+0x373  
f8876f14 f82e0754 026e6f44 8269c0e0 8269c0e0 USBPORT!  
USBPORT\_DoneTransfer+0x137  
f8876f4c f82e1f6a 8269c028 80541ac8 8269c230 USBPORT!  
USBPORT\_FlushDoneTransferList+0x16c  
f8876f78 f82effb0 8269c028 80541ac8 8269c028 USBPORT!USBPORT\_DpcWorker  
+0x224  
f8876fb4 f82f0128 8269c028 00000001 8271b368 USBPORT!  
USBPORT\_IsrDpcWorker+0x37e  
f8876fd0 80540f7d 8269c64c 6b755044 00000000 USBPORT!USBPORT\_IsrDpc  
+0x166  
f8876ff4 80540c4a f7c9d8d4 00000000 00000000 nt!KiRetireDpcList+0x46  
f8876ff8 f7c9d8d4 00000000 00000000 00000000 nt!KiDispatchInterrupt  
+0x2a  
80540c4a 00000000 00000009 bb835675 00000128 0xf7c9d8d4

STACK\_COMMAND: kb

FOLLOWUP\_IP:

Wdf01000!FxRequestBase::CompleteSubmitted+89  
f7167245 eb0c jmp Wdf01000!  
FxRequestBase::CompleteSubmitted+0x97 (f7167253)

SYMBOL\_STACK\_INDEX: 5

SYMBOL\_NAME: Wdf01000!FxRequestBase::CompleteSubmitted+89

FOLLOWUP\_NAME: MachineOwner

MODULE\_NAME: Wdf01000

IMAGE\_NAME: Wdf01000.sys

DEBUG\_FLR\_IMAGE\_TIMESTAMP: 4549b23a

Re: Illegal instruction – code c000001d (!!! second chance !!!)

Re: Illegal instruction – code c000001d (!!! second chance !!!)

FAILURE\_BUCKET\_ID: 0x7f\_d\_VRF\_Wdf01000!  
FxRequestBase::CompleteSubmitted+89

BUCKET\_ID: 0x7f\_d\_VRF\_Wdf01000!FxRequestBase::CompleteSubmitted+89

Followup: MachineOwner

FOLLOWUP\_IP:  
Wdf01000!FxRequestBase::CompleteSubmitted+89  
aa030245 eb0c jmp Wdf01000!  
FxRequestBase::CompleteSubmitted+0x97 (aa030253)

SYMBOL\_STACK\_INDEX: 4

SYMBOL\_NAME: Wdf01000!FxRequestBase::CompleteSubmitted+89

FOLLOWUP\_NAME: MachineOwner

MODULE\_NAME: Wdf01000

IMAGE\_NAME: Wdf01000.sys

DEBUG\_FLR\_IMAGE\_TIMESTAMP: 4549b23a

FAILURE\_BUCKET\_ID: 0xFC\_Wdf01000!FxRequestBase::CompleteSubmitted+89

BUCKET\_ID: 0xFC\_Wdf01000!FxRequestBase::CompleteSubmitted+89

Followup: MachineOwner

Suggest me something, this is very urgent for me.  
Thanks in advance.

~Vijji

.

Re: Illegal instruction – code c000001d (!!! second chance !!!)