

## Re: [Q] TRacking down asserts by NdisTest

**Source:**

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2004-04/1014.html>

---

**From:** Gramps ([gramps\\_at\\_old-fart.com](mailto:gramps_at_old-fart.com))

**Date:** 04/19/04

Date: Mon, 19 Apr 2004 13:56:25 -0700

Thanks, Mitesh. In looking at the dump, the assert was triggered at `ndistest!INdtReceivePacketCommon+0x133`. Before this, a call was made to `NdtPrintx`. A pointer was pushed on the stack for the following string:

```
0: kd> da b90337ce
b90337ce "%s: Error--not enough data indi"
b90337ee "cated.FirstBuffer = %p.PacketDat"
b903380e "aLength = %d.PacketSize = %d."
```

Sure enough, the IM driver I'm testing got a call to its `PtReceive` where the `PacketSize` was `0x27c` but the `LookAheadBufferSize` was `0x200`. Alas, I don't know what the underlying miniport's maximum lookahead buffer size was using (oh, if only `ndiskd` worked on XP SP1) but since `Ndistest` can't set this through `OID_GEN_MAXIMUM_LOOKAHEAD`, it must be prepared to call `MiniportTransferData` to get the rest of the packet.

Os is it? Does `Ndistest` handle receive lookahead indications where the lookahead buffer size may be smaller than the total packet size? Does it support calls to the driver's `MiniportTransferData`?

"Mitesh Desai [MSFT]" <[miteshd@online.microsoft.com](mailto:miteshd@online.microsoft.com)> wrote in message  
news:%23j91gVjJEHA.3688@TK2MSFTNGP10.phx.gbl...

```
> NDISTest 4.03.30 symbols are available in the packages at
> http://www.microsoft.com/whdc/hwtest/device/default.msp?area=network (for
> 4030030_32.exe, symbols are in i386\symbols\sys directory, for 4030030_64 in
> ia64\symbols\sys).
>
> NDISTest does not write anything in the registry before it triggers the
> breakpoint.
>
> From a memory dump file it is not easy to figure out the reason for the
> assert (because of compiler optimization asserts for different reasons get
> combined into a single assert). If the breaking stack has your miniport on
> it you can probably figure out what was happening. If its on receive code
> paths (generally it is), see if you can find the packets which were just
> indicated. The NDISTestHelp.chm document explains the NDISTest packet
> format. You can verify if the packet length matches the expected length in
> NDISTest protocol header if the data is valid. If its on the send code path,
> its generally because packets are pending in the miniport.
>
```

> *Mitesh*  
> --  
> *This posting is provided "AS IS" with no warranties, and confers no rights.*  
> *"Gramps" <gramps@old-fart.com> wrote in message*  
> *news:ePST1jiJEHA.1224@TK2MSFTNGP11.phx.gbl...*  
> *Howdy!*  
>  
> *I trying to track down the cause of an assert within NdisTest 4.3.30.*  
> *Unfortunately, this assert happened on a machine that wasn't set up for the*  
> *debugger. So, the assert caused the system to crash*  
> *(KERNEL\_MODE\_EXCEPTION\_NOT\_HANDLED) so all I have is the memory dump. We're*  
> *trying to repro this with KD enabled on the machine, but...*  
>  
> *Here's a couple of question:*  
> *Is the .pdb for ndistest.sys available anywhere?*  
> *Does ndistest.sys save anything meaningful in a register before it triggers*  
> *the int 3?*  
> *Assuming 2 is true, then is there documentation that would descibe the*  
> *register value for the reason for the assert?*  
> *Thanks!*  
>  
>