

# Re: Intel rejects Vista, will stay with XP and wait for windows7

---

*Source:* <http://sci.tech--archive.net/Archive/sci.electronics.design/2008-06/msg05291.html>

---

- *From:* JosephKK <[quiettechblue@xxxxxxxx](mailto:quiettechblue@xxxxxxxx)>
  - *Date:* Sun, 29 Jun 2008 10:49:25 -0700
- 

On Thu, 26 Jun 2008 12:56:28 -0700 (PDT), Le Chaud Lapin <[jaibuduvin@xxxxxxxx](mailto:jaibuduvin@xxxxxxxx)> wrote:

On Jun 26, 9:36 am, pantel...@xxxxxxxx wrote:

I seriously doubt MS has the power to do that.  
I have designed cards for in the PC, who will stop me from writing an XP driver?  
And with Intel moving away from Vista, sure Intel will want to sell their chips.  
If more people use XP then Vista they will provide XP drivers, no matter what.  
MS is just a con that charges big money for a 1\$ DVD copy of bad software,  
they have no other power then that of money.  
Anyways, even if MS sues and succeeds then they will go Unix.  
The Asus eeePC, running Linux on Intel, already is such a big success that it makes up 30% of Asus sales.  
People want performance, not bloat.  
Sign on the wall.

If, while using Vista, you get the same feeling that you get while sitting on wicker chair...(yes, it holds you up, but still seems flimsy), that has to do with an internal Class War going on inside Microsoft. In one word...

.NET

If anything is to cause the disintegration of sound engineering practices inside of Microsoft, it is .NET. .NET programmer population has been growing faster than that of hard-core C/C++ programmers, and they bring a mode of thinking that can only be described as "frighteningly sloppy for an engineer". .NET programmers who fall victim to this mode of thought are under the opinion that disciplined thought in engineering (software or otherwise), is not really

## Re: Intel rejects Vista, will stay with XP and wait for windows7

necessary if you write the software so that you do not have to be disciplined while thinking. Everything is extensible. Everything is changeable. Everything can communicate and indicate its whims and needs to everything else.

"Why can't we components just all get along?"

The result, as you can imagine, is a mess. Don Box was one of the original proponents of COM, which started the mess, and it got worse and worse, first with COM+, then DCOM, and now .NET, with all the .NET junk that goes with it, like C#, C++/CLI (in which Microsoft tried very hard to hi-jack C++ but failed), and VB.NET. Microsoft has been dishonest in saying "It makes programming easier...", but the truth is that it breaks down a certain rigidity that is required of reusable components in software. Actually, Microsoft would like to get away from writing libraries for multiple languages because of the xN effect in terms of cost.

If you want an analogy in hardware, imagine a capacitor that had no specification, not even capacitance. You start with a base thing called "capacitor" and reuse that thing over and over in all your circuits. Then, \*at run-time\*, other components in the circuit have a conversation with your generic do-it-all capacitor. "Are you electrolytic?" "What's your humidity range?"

```
If (out_of_range(humidity))  
blink_red_warning_LED_to_notify_user();
```

I know this sounds absurd, but this is the way some .NET programmers "write code." More conversations that might be had inside the circuit:

1. WARNING – Humidity out of range for this type of capacitor.
2. WARNING – excessive noise detected.
3. WARNING – Zener diode used when normal diode should have been used.
4. WARNING – unrecognized bit pattern on demux...what should I do now?
5. WARNING – race condition detected, maybe, you might wanna check..
6. WARNING – input impedance too low, at least 5kohm expected since this is my new-and-improved circuit

You get the idea – the code is \*CONSTANTLY\* having conversations with the user. Note that I say "user", not making a distinction between the engineer who designed the circuit, or the 5-year-old who is looking at her play oven and wondering why so many red lights are flashing. It's as if they forget that whole point was to engineer a finished product that does not have humanized conversations with every user that walks by. Absurd, isn't it?

Then, when they finish the product, they notice that the software complains far too much to be let loose in the field, so out comes the duct tape, which they wrap around the entire product (or catch all

Re: Intel rejects Vista, will stay with XP and wait for windows7

exceptions for you Java/C++ folks), then the plaster, then they buff it , paint it, shrink wrap it, and beef up the customer support department because, after all, they do love to have conversations.

There are actually tens of thousands of programmers who see nothing wrong with this mode of thinking. If you propose to them that the engineer might structure the systems so that absurd questions never need be asked, that absurd statements never need be made, that all those ridiculous WARNING diodes be done away with...that the device simply be designed right..they regard you with contempt as if you do not appreciate the benefit of a circuit being able to tell you if it was designed poorly.

:))

–Le Chaud Lapin–

Good rant. You nearly hit the nail. Think about it.