

6 Software Tips For Hardware Makers



We often hear that the current state of software development is still dysfunctional. Scott Rosenberg recently wrote a book to that effect called [Dreaming In Code](#). He takes a look at the question *Why is software so hard?* According to Scott, Software development's history is marred by poor quality, missed deadlines, and cost overruns, primarily due to a persistent dysfunctional culture.

And he's talking about software written by companies who are in the business of writing software. **Well if software written by software companies is so bad, how bad is the software written by hardware companies?**

Very bad.

I'm sure there are a few exceptions, but companies that I think should know better write atrociously poor software. And frankly, I'm getting tired of it. Here are some basic principles for hardware makers to keep in mind when writing software for their products.

1. The computer belongs to me, not you!

Did you ever notice that my documents are in a folder named **My Documents**. Not **Your Documents** (in Vista, it's just *Documents*, but there's still an implicit *my* in there).

That means that folder belongs to me. Not you.

So I beg you, stop adding your folders in there. There is a proper location for *your* stuff, in the *Application Data* directory. Since you're too lazy to understand how Windows works, I'll write some sample code for you in C# that demonstrates where to place your application's files. I'm sure you can figure out the C++ equivalent.

```
string appDataPath = Environment.GetFolderPath
(Environment.SpecialFolder.ApplicationData);

string yourAppDataPath =
Path.Combine(appDataPath, "YourApplicationName");
if(!Directory.Exists(yourAppDataPath))
{
Directory.CreateDirectory(yourAppDataPath);
}
```

Brush up on the [Environment.SpecialFolder](#) enumeration. It'll come in handy.

For user settings and application cache data, you might also consider using [Isolated Storage](#). Just don't put it in *My Documents*.

2. Don't Assume The User Is An Administrator

While I'm ranting on this topic, I should also mention that any data your application needs to write should not go in

the *Program Folders* directory. Not everybody runs their machine as an administrator and better you learn that lesson now than when Vista is widely disseminated.

3. Learn About the Platform And The Services It Provides You.

I recently bought a USB enabled Universal Power Supply (UPS) that was well reviewed. It appears to be a great hardware product, but the software is crap.

The point of getting a USB enabled UPS is that the UPS can shut down windows gracefully if the power goes out. But rather than intergrating cleanly with [Windows XP Power Management](#), they wrote their own ugly little system tray applet. Why not take advantage of what the OS provides?

To me, this example and the previous example I mentioned with the *Application Data* folder belies a willful ignorance and disregard for writing good software for Windows. It's due to an unwillingness to take the time to learn about the platform.

4. Honor Your Obligations and Promises.

When installing HP drivers and software for my scanner (which sadly doesn't work with Vista at all), the installation process provides me with a checkbox "Add an HP Share-To-Web icon to the desktop?".

I responded with my usual response, *Hell no!*, by unchecking the checkbox. But what happens when the installation is complete? There's the icon on my desktop. Not only that, the icon is impossible to delete. I mean **impossible!**

I tried to right click it, there's no *delete* option. I click *properties* and select *remove icon from desktop* and click *Apply*. The dialog hangs, never returning. Fantastic!

Please, give users a choice. **And when they make a choice, honor it!**

5. We Really Don't Want Your Cruft

Really. Seriously. We don't.

It's simple. If a piece of software is something we really want, we'll take the time to find it and install it. We don't need you to install a bunch of crapware alongside your drivers or main application.

If it needs to be pork barreled in order to get it on our machines, we probably don't want it.

Trust me on this one. The extra endorsement money you get for bundling probably won't make up for the loss in customer satisfaction.

6. If The Software Sucks, We Think The Hardware Sucks.

Again, quite simple. The majority of user interaction with a piece of hardware is really via the software. If the software is clunky and hard to use, or worse, just flat out fails to work. We associate the failure with the entire product, hardware and all. **After all, if a company can't take the time to write quality software, why should we trust in the quality of the hardware?**

Conclusion

So that's it. All I ask is that hardware makers take as much care writing their software as they do building their hardware. Perhaps more care, given how flimsy hardware can be these days.

When the supporting software is good, customers will rave about your products.

Author: Phil Haack

Copied from: http://haacked.com/archive/2007/01/14/6_Software_Tips_For_Hardware_Makers.aspx

Article downloaded from page [eioba.com](#)