

Re: Driver Expert

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2004-09/0487.html>

From: Philip Lukidis (*pagefault0x0_at_hotmail.com*)

Date: 09/09/04

Date: Thu, 9 Sep 2004 09:35:03 -0400

"Dennis Burns" <dburns@rtessentials.com> wrote in message
news:eAlsRQeIEHA.1244@TK2MSFTNGP15.phx.gbl...

> *Hey guys, some of what you're discussing is hard for me to appreciate.*

What

> *about the value of:*

> *Integrated environment with Visual Studio*

> *SoftICE and Visual SoftICE*

> *Bounds Checker*

> *Performance Analyzer*

> *Test Coverage*

>

> *I attended a Compuware Webinar and was pretty impressed, but then again*

I'm

> *a pushover.*

>

> *More comments?*

> *Dennis*

>

[snip]

Myself I was talking about Driverworks (DW) and not Driverstudio as a whole. On Driverworks, I'll just add that *any* framework requires that you understand WDM and your own driver model very very well. Since relatively few people use DW, you'll have to convert your questions to WDM when posting to any group. After using DW for over 1 year, I'm somewhat divided. On the one hand, you do have the source code, so any issues can be addressed. Of course, you have to do so for each issued version, unless you file a bug report. From my perspective, some things seem to be lacking, as far as I can see. Although there is a `IO_REMOVE_LOCK` embedded in their PnP device object class, it is not used at all (so far as I see), leaving you to detect this and make sure you use it. They use an I/O incrementing system similar to the toaster driver in the DDK, but that is not as secure as a real `IO_REMOVE_LOCK`. `IoSetCompletionRoutineEx` is not used at all (maybe it's controversial on just how useful it is). But then `IoSetCompletionRoutineEx` not used at all in the DDK samples. Cancel safe queues are not used (although a class using them is present). And so on. OTOH, I really do

prefer C++ in some situations. There are quite a few samples, although I've used only a few of them. When the DDK samples are lacking, these samples can be useful. DW also provides an STL library implementation, which, depending on your feelings, can be quite useful or blasphemous. I happen to think that it can be useful in some situations. And for what is implemented very well, you have a class which can be used in other parts of your driver, and other projects. My final position is that you still need to know WDM very well, and go over the PnP/power implementation, and other relevant parts which you use. The responsibility lies with you to do this. Many times, fixes can be done by deriving from DW classes and overriding the offending function. This can get potentially messy from a visual point of view.

I find True Coverage and True Time to be very useful indeed. I have an exception here: they don't work with the checked OS build. I would like to use TC in the checked OS (or at least with the checked HAL and kernel). The source of this information is Compuware tech support. To be absolutely fair, they said they were migrating TC and TT to their "OSINFO" methodology, which would allow for checked/BETA OS usage, at least as much as SoftICE does.

To be frank, I have not used Boundschecker in some time, so maybe someone who has used it in Driverstudio 3.1 could comment. (I used older versions only)

WinDBG vs SoftICE...well, each debugger has their own strengths, and I can only provide a small sample here. SoftICE can be useful for long, debugging sessions, especially as the stepping speed is not excruciatingly slow like WinDBG (no I can't use 1394 for WinDBG). Not to mention the UI in the latest WinDBG was designed by a groups of dedicated Satanists. One other thing that I like about SoftICE is that my source code is embedded in the symbols. WinDBG requires source code for each driver version that you want to debug, but your major revisions will have labels in VSS anyways. However, minor revisions may not, so having the source in the SoftICE symbols is convenient. Also, apparently Visual SoftICE can be used over networks...quite useful, though I never needed it yet. SoftICE can work with checked OS builds, with the use of an OSINFO file, which contains the hooking information. Maybe someone can comment on how up to date OSINFO is for BETA OS/hotfixes, as I don't know. WinDBG allows for the same, as long as the symbol server is up to date (so it has always been for me). The MS symbol server provides BETA OS symbols.

One weakness with SoftICE: I can't select processor# for a stack trace on that processor. WinDBG allows me to see stack traces on each processor, by selecting the processor via the ~x command. This does not exist in SoftICE, AFAIK. It's quite straightforward to convert between virtual and physical addresses in SoftICE, and more involved in WinDBG. I'm sure each debugger has their own strength as far as commands go, and I surely cannot cover them here.

microsoft.public.development.device.drivers: Re: Driver Expert

I have not heard of performance analyzer...do you mean True Time? If so, reference my comments above.

As for integration with Visual Studio, I don't really care too much for that. I use build and keep my life simple.

HTH,

Philip Lukidis