

---

Subject: Re: Another VM conundrum

Posted by [JD Smith](#) on Thu, 11 Dec 2003 22:14:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 11 Dec 2003 12:46:11 -0700, David Fanning wrote:

> JD Smith writes:

>

>> Having fixed my main-level XManager issues (thankfully fairly trivial),  
>> I now find myself with another problem. Typically, I recommend setting  
>> something like:

>>

>> device,DECOMPOSED=0,TRUE=16,RETAIN=2

>>

>> in your IDL startup file, where RETAIN=2 is good for most Linux  
>> machines (whose window managers don't provide decent backing store),  
>> DECOMPOSED=0 is immortalized on David's site, and TRUE=16 is just for  
>> good measure, for those X servers which \*claim\* to offer DirectColor  
>> but don't really (sadly, not few in the Linux world). However, I leave  
>> it up to the end user to find some combination of device incantations  
>> which work for them.

>>

>> That's fine for interactive usage, but with a runtime/VM application,  
>> the startup file is never consulted, so you need to perform these  
>> device incantations yourself in the code. Has anyone solved this  
>> problem in a platform-independent way? Would something like:

>>

>> device,DECOMPOSED=0,RETAIN=2

>>

>> always be safe, even for older 8-bit PSEUDOCOLOR machines? Or is there  
>> some method of interrogating DEVICE for info before making a final  
>> decision on RETAIN and color-model settings? What guidance from the  
>> IDL color gurus?

>

> Oh, I \*hate\* to put those DEVICE commands into the code! A pure  
> nightmare when you are trying to write code that works \*everywhere\*:  
> Z-buffer, PostScript, on various and sundry displays, etc. Here are a  
> couple of tips (taken from the TVIMAGE code, of course, which \*does\* run  
> everywhere).

>

>

Thanks David. What's different about this problem is I'm not interested in restoring the decomposed state, etc. since only the single program will be running under the VM; in this sense I can be very selfish and just set it to what I need -- if I knew what that was. What I want is a setup such that, for any combination of visual (TrueColor, PseudoColor, etc.), depth (8,16,24bit), and backing behavior (stored by server, stored by OS, stored by IDL), will give "correct" colors, and images which won't be clobbered

when windows are drug across them. I'm thinking I can probably get away with:

```
device,DECOMPOSED=0,RETAIN=2
```

which will always run since you need IDL>=6.0 to run the VM. What I worry about is how this will interact with various other systems (like Windows, Solaris, or MacOSX). Can anyone offer any feedback as to whether this call succeeds in doing what I asked without creating other problems?

Thanks,

JD

---