

---

Subject: Re: Another VM conundrum

Posted by [David Fanning](#) on Thu, 11 Dec 2003 19:46:11 GMT

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

---

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).

```
; Which release of IDL is this?  
thisRelease = Float(!Version.Release)
```

```
; Decomposed color off if device supports it.  
CASE StrUpCase(!D.NAME) OF  
  'X': BEGIN  
    Device, Get_Visual_Depth=thisDepth  
    IF thisRelease GE 5.2 THEN $  
      Device, Get_Decomposed=thisDecomposed
```

```
Device, Decomposed=0
ENDCASE
'WIN': BEGIN

Device, Get_Visual_Depth=thisDepth
IF thisRelease GE 5.2 THEN $
    Device, Get_Decomposed=thisDecomposed
Device, Decomposed=0
ENDCASE
'MAC': BEGIN
Device, Get_Visual_Depth=thisDepth
IF thisRelease GE 5.2 THEN $
    Device, Get_Decomposed=thisDecomposed
Device, Decomposed=0
ENDCASE
ELSE: thisDepth = 8
ENDCASE
```

; Restore Decomposed state if necessary.

```
CASE StrUpCase(!D.NAME) OF
'X': BEGIN
    IF thisRelease GE 5.2 THEN Device, Decomposed=thisDecomposed
    ENDCASE
'WIN': BEGIN
    IF thisRelease GE 5.2 THEN Device, Decomposed=thisDecomposed
    ENDCASE
'MAC': BEGIN
    IF thisRelease GE 5.2 THEN BEGIN
        Device, Decomposed=thisDecomposed
```

Cheers,

David

--

David W. Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Phone: 970-221-0438, E-mail: david@dfanning.com  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Toll-Free IDL Book Orders: 1-888-461-0155

---

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

---

---

Subject: Re: Another VM conundrum

Posted by [Rick Towler](#) on Thu, 11 Dec 2003 23:05:06 GMT

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

---

"JD Smith" wrote in 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

>>>

<snip>

> 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?

FWIW, I have been throwing in the same line with the direct graphics applications I have been compiling for the VM and haven't run into any issues. Tested on 8 and 32bit windows systems, 24bit X (solaris), and 8bit X (win32 X server).

-Rick

---

---

Subject: Re: Another VM conundrum  
Posted by [btt](#) on Fri, 12 Dec 2003 14:18:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

JD Smith wrote:

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

>

>

SNIP

> 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?

>

Hello,

I tested the routine appended below as a VM on MacOSX X11. It seems to work fine when other windows cover/uncover and the window is minimized/restored.

I noticed that the terminal session I used changed from tcsh to idl and the the x11 terminal shows two windows: an xTerm (which I have to have up and running but don't use for the idl command prompt) and viola! Test\_VM. So, unlike Windows, you can track down the the widget even when it's hiding.

I wonder if you might have in mind a different kind of test - something we code build as VM-ware and test out. I'd be happy to give it a whirl if needed.

Is this potential weakness/problem limited to direct graphics?

Ben

;;;;;BEGIN CODE

PRO Test\_VM

device,DECOMPOSED=0,RETAIN=2

file = FILEPATH('rose.jpg', SUBDIRECTORY=['examples','data'])

truelImage = read\_image(file)

dim = SIZE(truelImage,/Dim)

bytelImage = bytscl(hanning(50,50))

Base = Widget\_Base(title = 'Test\_VM')

Draw = Widget\_Draw(base, xsize = dim[1], ysize = dim[2])

Widget\_Control, base, /realize

Widget\_Control, draw, get\_value = drawID

Wset, drawID

TV, truelImage, true = 1

TV, bytelImage

XMANAGER, 'test\_vm', base

END

;;;;;END CODE

---

---

Subject: Re: Another VM conundrum

Posted by [David Fanning](#) on Fri, 12 Dec 2003 15:25:50 GMT

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

---

Ben Tupper writes:

> I tested the routine appended below as a VM on MacOSX X11. It seems to  
> work fine when other windows cover/uncover and the window is  
> minimized/restored.  
>  
> I noticed that the terminal session I used changed from tcsh to idl and  
> the the x11 terminal shows two windows: an xTerm (which I have to have  
> up and ruu=ning but don't use for the idl command prompt) and viola!  
> Test\_VM. So, unlike Windows, you can track down the the widget even  
> when it's hiding.

Well, this is not exactly testing the functionality that  
I was talking about. Here is a modification that does.  
Here is what you want to do.

1. Maximize your IDLDE window.

2. Type "TEST\_VM". The application appears on the display,

and, yes, it has its own application icon on the task bar at the bottom.

3. Click the "Test It" button. A TextBox dialog appears. Before you type text into this dialog, click in your IDLDE window. Whoops! The dialog disappears behind the IDLDE window and it is going to be *\*hard\** to find! You should be able to get the Test\_VM application back in front of the IDLDE, but you won't be able to *\*do\** anything with it until you find that damn modal dialog!

You will need the TextBox program from my web page to run the following program:

<http://www.dfanning.com/programs/textbox.pro>

Cheers,

David

```
.*****  
,  
Pro Test_VM_Event, event  
Widget_Control, event.top, Get_UValue=labelID  
theText = TextBox(Title='Can you see me now?', Label="Type Something")  
Widget_Control, labelID, Set_Value=theText  
END
```

PRO Test\_VM

device,DECOMPOSED=0,RETAIN=2

file = FILEPATH('rose.jpg', SUBDIRECTORY=['examples','data'])

```
truelImage = read_image(file)  
dim = SIZE(truelImage,/Dim)  
byteImage = bytscl(hanning(50,50))
```

```
Base = Widget_Base(title = 'Test_VM', column=1)  
Draw = Widget_Draw(base, xsize = dim[1], ysize = dim[2])  
labelID = Widget_Label(base, Value='Text Here', /Dynamic)  
button = Widget_Button(base, Value='Test It')
```

```
Widget_Control, base, /realize, Set_UValue=labelID  
Widget_Control, draw, get_value = drawID
```

Wset, drawID  
TV, truelmage, true = 1  
TV, bytelmage

XMANAGER, 'test\_vm', base  
END

,\*\*\*\*\*  
,

--

David W. Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Phone: 970-221-0438, E-mail: david@dfanning.com  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: Another VM conundrum  
Posted by [JD Smith](#) on Fri, 12 Dec 2003 16:24:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 12 Dec 2003 07:18:02 -0700, Ben Tupper wrote:

> JD Smith wrote:  
>> On Thu, 11 Dec 2003 12:46:11 -0700, David Fanning wrote:  
>>  
>>  
> SNIP  
>  
>> 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?  
>>  
>>  
> Hello,  
>  
> I tested the routine appended below as a VM on MacOSX X11. It seems to  
> work fine when other windows cover/uncover and the window is  
> minimized/restored.  
>  
> I noticed that the terminal session I used changed from tcsh to idl and  
> the the x11 terminal shows two windows: an xTerm (which I have to have



> up and running but don't use for the idl command prompt) and viola!  
> Test\_VM. So, unlike Windows, you can track down the the widget even  
> when it's hiding.  
>  
> I wonder if you might have in mind a different kind of test - something  
> we code build as VM-ware and test out. I'd be happy to give it a whirl  
> if needed.  
>  
> Is this potential weakness/problem limited to direct graphics?  
>

Thanks Ben. Glad to hear it works well enough. The test I wanted was just the "device,DECOMPOSED=0,RETAIN=2" you use, and could have been done in the VM or normal version. The reason the VM even comes into the question at all is that, with command-line IDL, you can just instruct the user to use a DEVICE setting which works in their IDL startup file, whereas in the IDLVM, no startup file is ever consulted, so you must compile in a best-guess DEVICE call.

JD

---

Subject: Re: Another VM conundrum  
Posted by [btt](#) on Fri, 12 Dec 2003 17:21:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

> Ben Tupper writes:

>

snip

>>

>> I noticed that the terminal session I used changed from tcsh to idl and  
>> the the x11 terminal shows two windows: an xTerm (which I have to have  
>> up and running but don't use for the idl command prompt) and viola!  
>> Test\_VM. So, unlike Windows, you can track down the the widget even  
>> when it's hiding.

>

>

> Well, this is not exactly testing the functionality that  
> I was talking about. Here is a modification that does.  
> Here is what you want to do.

>

> 1. Maximize your IDLDE window.

>

> 2. Type "TEST\_VM". The application appears on the display,  
> and, yes, it has it's own application icon on the task  
> bar at the bottom.

>

> 3. Click the "Test It" button. A TextBox dialog appears.  
> Before you type text into this dialog, click in your  
> IDLDE window. Whoops! The dialog disappears behind the  
> IDLDE window and it is going to be \*hard\* to find!  
> You should be able to get the Test\_VM application  
> back in front of the IDLDE, but you won't be able  
> to \*do\* anything with it until you find that damn  
> modal dialog!  
>

Ok! I'm with you now. The textbox window was buried as you describe.

On a different note, I have to mention that I seriously doubt that anyone using IDL on MacOSX is using IDLDE. Even cutting and pasting is badly disabled in the editor window.

Ben

---