Subject: Colors and Virtual Machine
Posted by Phillip Bitzer on Fri, 30 Jun 2006 21:32:33 GMT
View Forum Message <> Reply to Message

OK, IDL gurus, here's another question for you.

I have created (with help from this news group and a couple of David Fanning's procedures) a GUI for a program that maps out data on map projection. It works great if I run it from IDL; nice color pic and the such.

Problem I encounter is when I save it with IDL's save procedure, and run the program in IDL VM. I get a black and white pic. That is, until I mouse over a widget_draw widget. Then I get colors all over the place; the correct colors are in the draw widget, but I also get a skewing of colors elsewhere on the monitor.

I'm sure I didn't include quite enough info, but I'm hoping this is a (somewhat) common problem that somebody has encountered before with some fix that I've overlooked.

Any ideas?

Subject: Re: Colors and Virtual Machine
Posted by JD Smith on Wed, 05 Jul 2006 18:23:31 GMT
View Forum Message <> Reply to Message

> No, a 6.0 VM would still have the bug. Your users would have to upgrade
> their VM's.

Hi Karl,

This brings up a related but different question. How hard do you guys strive to keep the binary .sav format for compiled code backward compatible? I.e. in statements like "This compiled .sav file requires IDL version 6.2 or later", how far in general will "or later" extend? Within major version number sets (e.g. 6.x?). Or is there any specific policy on this?

Obviously, forward compatibility is harder, e.g. allowing a 6.2-compiled .sav to run under v5.X, but this is typically true of source code as well, so there's no real expectation for that to work. However, 99.9% of IDL source code (my guess) is backward compatible --- I'm just wondering how often this compatibility gets broken for the compiled code, due to changes in the .sav format or other ABI issues?

JD

Subject: Re: Colors and Virtual Machine
Posted by Karl Schultz on Wed, 05 Jul 2006 21:09:34 GMT
View Forum Message <> Reply to Message

On Wed, 05 Jul 2006 11:23:31 -0700, JD Smith wrote:

>> [quoted text muted]
>
>
> Hi Karl,
>
> This brings up a related but different question. How hard do you guys
> strive to keep the binary .sav format for compiled code backward
> compatible? I.e. in statements like "This compiled .sav file requires
> IDL version 6.2 or later", how far in general will "or later" extend?
> Within major version number sets (e.g. 6.x?). Or is there any
> specific policy on this?
>
> Obviously, forward compatibility is harder, e.g. allowing a
> 6.2-compiled .sav to run under v5.X, but this is typically true of
> source code as well, so there's no real expectation for that to work.
> However, 99.9% of IDL source code (my guess) is backward compatible
> --- I'm just wondering how often this compatibility gets broken for
> the compiled code, due to changes in the .sav format or other ABI
> issues?
>
> JD

Hey JD,

As you know, save files containing data are always compatible.

For code, our docs say that recompilation is needed when the "internal code format" changes and goes on to say that the format changed back in IDL 5.5 and any save files compiled with IDL versions prior to 5.5 need to be recompiled to run with IDL versions 5.5 and later. I think 5.5 was about 5-6 years ago.

Major releases tend to coincide with significant functionality improvements and it would be too hard to time an internal code format change that is needed right now with major feature releases. Although I do understand the value of a major version number being associated with a stable API/ABI level.

I think that we would advertise very clearly when such a change is made. We did so with 5.5. This situation is a lot like changes to the external programming interface such as the IDL_STRING string length field. I think that we would try very hard to avoid these sorts of changes and make them only when there are very good reasons.

Karl

Subject: Re: Colors and Virtual Machine
Posted by JD Smith on Wed, 05 Jul 2006 21:26:31 GMT
View Forum Message <> Reply to Message

On Wed, 05 Jul 2006 21:09:34 +0000, Karl Schultz wrote:

> On Wed, 05 Jul 2006 11:23:31 -0700, JD Smith wrote:
>
>>> [quoted text muted]
>>
>>
>> Hi Karl,
>>
>> This brings up a related but different question. How hard do you guys
>> strive to keep the binary .sav format for compiled code backward
>> compatible? I.e. in statements like "This compiled .sav file requires
>> IDL version 6.2 or later", how far in general will "or later" extend?
>> Within major version number sets (e.g. 6.x?). Or is there any specific
>> policy on this?
>>
>> Obviously, forward compatibility is harder, e.g. allowing a 6.2-compiled
>> .sav to run under v5.X, but this is typically true of source code as
>> well, so there's no real expectation for that to work. However, 99.9% of
>> IDL source code (my guess) is backward compatible --- I'm just wondering
>> how often this compatibility gets broken for the compiled code, due to
>> changes in the .sav format or other ABI issues?
>>
>> JD
>
> Hey JD,
>
> As you know, save files containing data are always compatible.
>
> For code, our docs say that recompilation is needed when the "internal
> code format" changes and goes on to say that the format changed back in
> IDL 5.5 and any save files compiled with IDL versions prior to 5.5 need to
> be recompiled to run with IDL versions 5.5 and later. I think 5.5 was
> about 5-6 years ago.
>

> Major releases tend to coincide with significant functionality
> improvements and it would be too hard to time an internal code format
> change that is needed right now with major feature releases. Although I
> do understand the value of a major version number being associated with a
> stable API/ABI level.
>
> I think that we would advertise very clearly when such a change is made.
> We did so with 5.5. This situation is a lot like changes to the external
> programming interface such as the IDL_STRING string length field. I think
> that we would try very hard to avoid these sorts of changes and make them
> only when there are very good reasons.

Well that doesn't sound so bad. Just so I'm certain I have this correct, aside from obvious compatibility issues in the source itself (i.e. if we assume the underlying source code itself would properly function), any compiled `.sav' produced with IDL>=v5.5 should work fine with any version of IDL >=v5.5? That's *much* better than I thought, and greatly reduces concerns of a given `.sav' file's usefulness as it sits gathering mold in some corner... it's not much worse than a pile of source code sitting in that corner, and in some ways, better, since it contains a snapshot of any external library code at a point of known compatibility.

Thanks,

JD
