

Reimar Bauer (R.Bauer@fz-juelich.de) writes:

- > I like very much the I of idl it stands for interactive.
- > For my opinion objects can't be interactive they are always well planned.
- >
- > The people starting with idl are almost very different in learning and
- > working with idl. In most cases my feeling is if they can theireselfs work
- > a bit on a solution without reading many books or asking a lot of people
- > for help they like idl more.

I'm not sure how this discussion got turned in the direction of objects. I'm pretty sure it started out as a discussion of how to write a modal dialog widget. But the fact that it did encourages me. I've learned in my programming therapy sessions that talking about something is the first step in the process of accepting it. :-)

Objects are definitely well planned. But the fact that they aren't interactive stems entirely from the nature of the objects we have been given in IDL, not from an inherent property of objects. The point is (and I know this for a fact) that if you have the right object library, objects can be as interactive--and as easy to build things with--as any other tool we have in IDL.

In fact, Dave and I have been building object tools in the past several days (CTV, CWINDOW, CWDELETE, etc.) that mimic basic IDL interactive commands, but which put the full power of objects into the hands of people who really need to know nothing about objects. We find the problem with objects is not the lack of interactivity, but the great temptation to be too clever with objects. They can, in fact, be TOO interactive!

Let me give you an example. Our basic image class, IMAGE2D, creates an image class that basically combines all the functionality of my programs TVIMAGE, FSC\_COLOR, SELECTIMAGE, and XCOLORS. Thus, the image can size itself into any window, with or without preserving its aspect ratio, can read and load new image files, can manipulate its own color table, can draw and maintain its own overlays (map coordinates, for example), can tell you the actual data value for any point you select on the image, can automatically apply any image processing object to itself (including those you might write in the future), and can support an unlimited (and

user-defined) number of undos and redos.

This all comes for free when you type "CTV, image".

And, as a bonus, if you don't have a window open when you type the command, a GRAPHICSWINDOW will open so that you have a completely resizable graphics window with your image inside it.

That's pretty interactive. And it can also be surprising.

Yesterday I noticed that when I loaded a PNG image that had a built-in color table that the image was drawing itself before I was ready for it to be drawn. It turned out that it was drawing itself because I was loading the PNG's color table in the image's color tool, and the color tool was smart enough to send a message to the image to redraw itself whenever its colors changed. So I was fooling around adding the capability to unregister a message.

The point is that a LOT of work goes into the first layer of objects that you write (what we call the Catalyst Library). But that as you work on the library, you spend less and less time on the low-level routines where I was working yesterday and more and more time building more complex objects and applications.

In the large satellite observation system we built with our first attempt at this library, we spent about 60% of our time designing and building the 25-30 core objects for our library and about 40% of our time building the actual application. I think everyone can appreciate that if your library is basic and powerful enough to be re-usable, that application building is going to be MUCH faster.

And in the meantime, even the object neophytes can take advantage of some of the object's really neat interactive functionality. :-)

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: Interactive Objects, Was: Simple GUI question  
Posted by [Paul Van Delst\[1\]](#) on Wed, 23 Apr 2003 14:45:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

"Richard G. French" wrote:

```
>  
> On 4/21/03 10:47 AM, in article MPG.190db260977ddbea989b5c@news.frii.com,  
> "David Fanning" <david@dfanning.com> wrote:  
>  
>> I'm not exactly sure why people are afraid of objects.  
>> Certainly the basics of object programming can be taught  
>> in half a day. I make sure I include at least this much time  
>> in any IDL programming class I teach.  
>>  
>  
> I have used IDL for a long time and I have not yet taken the time to learn  
> how to use objects in any language. I've gotten as far as using an  
> occasional widget or two, and in some cases a rather complex set of compound  
> widgets, but not objects. Nearly all of my programming involves either  
> scientific image processing or computations, and the sequence of things is  
> typically as follows:
```

[ snip reasons he uses IDL...mostly same as mine btw ]

```
> To follow David's analogy, I sometimes think that my widget programming  
> style ends up being a design for an all-in-one blender, buzz saw, and CD-ROM  
> burner, without doing any of them well. Object programming leaves me at a  
> loss in terms of these fundamental design principles. The usual thing that I  
> see is an example that says: "Start with an object that does one little  
> thing, such as squaring a number or an array." But then I end up with about  
> 100 lines of code which, in the end, just do what I can already do in one  
> line of IDL - square a number or an array. What I need to see is an example  
> of simple objects that can do something useful in IDL that I can't already  
> do in a few lines of code. That might help me get over the hump so that I  
> could join David's bandwagon.
```

My personal opinion is that it depends on what you're doing. Your example above is a good one - using objects for every little thing leads to unimaginable code bloat (even for someone as programming-ly verbose as I am.) I've experimented with using objects in IDL and found them to be totally UNuseful for what I use IDL for -- mucking about with data, prototyping algorithms, and producing eye-candy for papers/proposals. I'm not knocking object oriented coding at all -- I try to use OO coding principles in my Fortran95 code as much as possible and it has helped enormously with code reuse and ease of use; but I also have 1000's more lines of code.

My wish for IDL is to produce more complete and easy-to-use display tools as part of the package. I was recently on a field trip where I had only matlab available to inspect the data and I was impressed with a simple little thing like how the equivalent of the plot command worked - you get a window where you can zoom in/out, annotate, etc -- all there right at your mouse/fingertips. I couldn't help but wonder why IDL doesn't have something like that (I won't mention Insight here....). I am aware there are plenty of user-produced things like that (I clobbered together my own a while back). Given what I use IDL for, and

having seen what other data analysis packages (not restricted to matlab) can do, the price paid for an IDL license makes me hope that the introduction of objects leads to the production of some really powerful, fast, integrated, and easy to use tools in IDL in the future.

> I should be working on my NASA proposal....

I should be working on my NOAA report....

paulv

--

Paul van Delst  
CIMSS @ NOAA/NCEP/EMC  
Ph: (301)763-8000 x7748  
Fax:(301)763-8545

---

Subject: Re: Interactive Objects, Was: Simple GUI question  
Posted by [David Fanning](#) on Wed, 23 Apr 2003 14:54:11 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Richard G. French (rfrench@wellesley.edu) writes:

> 3) A KEY POINT for me is that I want absolutely reproducible results without  
> having to do any interaction. [...] In other words, I want to get rid of  
> the 'I' in IDL as my final step, and be able to run a batch job that  
> gives me the final result.

I would have to agree that there is little point  
in writing widget programs if what you want to end  
up with is a set-in-stone program with no interactivity.  
Why waste your time adding something you don't want?

The whole point of widgets is interactively. I think a  
case could be made that the whole point of objects is  
interactivity with a memory. In other words, changes  
you make to an object persist until you change it.  
You can certainly save an object as it is, and call  
it up years later (I presume from your described programming  
style that you don't update your IDL version either, so I  
am positive this will work) and have it do exactly the  
same thing it just did. That's how persistent those suckers  
are.)

(I took a natural history course with my youngest son  
last summer and the first thing we learned about were these  
strange little critters -- I can't recall their name now,

although I'm sure my son could -- that hang out in the moss on desiccated boulders. They can stay dormant for hundreds of years, but throw a little water on them and they are up for a swim! You would like these little guys, Dick. :-)

- > Now, how about objects? I know that a lot of these are really cool! I like
- > being able to display images in a resizable window and having the image
- > follow along. I like being able to click on an axis of a plot and reformat
- > the style. I like using objects that are self-contained and do a
- > well-defined task. My problem, when thinking about actually writing one of
- > these beasts, is that there seem to be too many choices!

Well, I can see how choices might be anathema to your overall programming goals. :-)

I guess I would submit that you like interactivity, too. You just prefer the slow method of interaction in which you type commands over and over again at the IDL command line. (I presume you have no reservations about using the arrow keys.) It might well be that this method is overall much faster than writing a general purpose "fool-around-with-it-until-I-understand-the-damn-thing widget. But one advantage of objects over widgets, I think, is that it seems much easier to write that one new "fool around" method to explore a wild-hare idea.

- > How many methods
- > should I specify? What about classes?

Here is the real problem for people just starting out, I think. I've been extremely reluctant to write this object programming in IDL book I keep talking about because I didn't want real programmers to laugh at me. I was so dumb and excited about objects I just started using them! It wasn't until the past year or so that I began to understand the power and benefit of using them intelligently.

But Dave and I spend a couple of hundred dollars a month on transatlantic telephone calls arguing the merits of this design or that design. An image data object, to give you just one example, is particularly annoying. Should an image base class transparently handle 8-bit and 24-bit images? Or should these be two separate classes, primarily because they are handled differently in processing steps? We have been back and forth probably a thousand times on this one. It is probably one of the few times when we are *\*both\** right!

We think one of the advantages of this Catalyst Library

we are writing is that some of these choices have been made for you. In other words, we have done a lot of thinking about the design framework. We would be the first to tell you it is not perfect, but it does allow you to get on with it, without a lot of hemming and hawing about whether or not this is the \*BEST\* design. I think only lots and lots of experience can tell you that.

- > As someone just beginning with
- > objects, it is hard to get a good sense for how to break up a problem into
- > manageable pieces, and to construct individual pieces that make sense.

I totally agree with this sentiment. My personal view (not shared, as far as I can tell, by the good folks at RSI) is that the only way you can get scientists migrating to objects is to give them objects that are similar to what they already know. Throwing them into objects by making them write object graphics programs is a fools errand, it seems to me. (But what do I know? I'm unemployed at the moment. :-)

- > To follow David's analogy, I sometimes think that my widget programming
- > style ends up being a design for an all-in-one blender, buzz saw, and CD-ROM
- > burner, without doing any of them well. Object programming leaves me at a
- > loss in terms of these fundamental design principles. The usual thing that I
- > see is an example that says: "Start with an object that does one little
- > thing, such as squaring a number or an array." But then I end up with about
- > 100 lines of code which, in the end, just do what I can already do in one
- > line of IDL - square a number or an array. What I need to see is an example
- > of simple objects that can do something useful in IDL that I can't already
- > do in a few lines of code. That might help me get over the hump so that I
- > could join David's bandwagon.

I'm pretty sure writing objects does NOT save you time up-front. Typically, a great deal of time and code goes into producing the lowest-level objects. If time is saved, it is on the back-end where applications can be built with the lowest-level objects. If objects are built properly, you find yourself spending less and less time with low-level objects and more and more time with high-level objects. It's then that the benefits kick in. I'm aware that by this time it might be too late for someone who absolutely has to have that NASA proposal done next week... :-)

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: Interactive Objects, Was: Simple GUI question  
Posted by [R.Bauer](#) on Wed, 23 Apr 2003 15:26:29 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

sometimes I believe using objects is the same kind of question as using word or latex to write a book

I know that I will have perfectly results using latex and that it could be compiled without a gui interface on a lot of platforms and no GUI is necessary to write the chapters. And I know a few years later I can do some changes or some enhancement and then I am able to recompile it.

But if I discuss this with our secretary she has a very different mind about this. She would do all the work in word.

Another object language which I know is Labview. We have several data acquisition programs written in this language. And in difference to idl you are writing the sources with a graphical user interface. This is extremly good designed. Because of programming with icons it is very easy to learn.

If I find an object idl program which is useful for us I like to use it. If it's necessary to do an enhacement I will do this too.

regards

Reimar

--

Forschungszentrum Juelich  
email: [R.Bauer@fz-juelich.de](mailto:R.Bauer@fz-juelich.de)  
<http://www.fz-juelich.de/icg/icg-i/>

=====

a IDL library at Forschungszentrum Juelich  
[http://www.fz-juelich.de/icg/icg-i/idl\\_icglib/idl\\_lib\\_intro.html](http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html)

---



Subject: Re: Interactive Objects, Was: Simple GUI question  
Posted by [mvukovic](#) on Thu, 24 Apr 2003 16:52:04 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning <david@dfanning.com> wrote in message  
news:<MPG.191056febe370ae989b63@news.frii.com>...

... stuff deleted ...

- > ... An image data object, to give you
- > just one example, is particularly annoying. Should an image
- > base class transparently handle 8-bit and 24-bit images? Or
- > should these be two separate classes, primarily because they
- > are handled differently in processing steps? We have been
- > back and forth probably a thousand times on this one. It is
- > probably one of the few times when we are *\*both\** right!

... more stuff deleted ...

Here I would interject that the proper approach is to make an abstract image object with the basic interface defined but blank, except for common tools. Then make two objects (one for 8 and the other for 24 bit images) where the interfaces follow that of the abstract object. In addition, these two can extend the basic image object based on the additional properties of each image representation.

Now, strictly speaking, you do not need to make an abstract object in IDL, (I do it by making the init function return 0), but I like to do it to help me organize thoughts, and also spice-up debugging :-)

The big advantage I found was that once I define the interface and functionality using the abstract class, and then implement one object that inherits the base class and test it in various applications, IT BECOMES MUCH EASIER (sorry for shouting) to implement a second object. It is almost guaranteed to work!

So, for example, first implement a simple ``\_image" object (I use preceeding underscores for things that are supposed to be somewhat hidden) with many empty methods, but also with much of the pointers to data structures defined. It also contains some non-empty methods that would identically work for 8- and 24-bit images. Also, all the empty methods in \_image print out a warning that they are not supposed to be called (they will be overridden shortly).

Next, implement the 8-bit image object that inherits \_image, implements the various methods (and thus overrides the empty methods of \_image), store the data in pointers, etc. Now you can test the 8-bit image object and start using it, and finding all the mistakes that you made. On cleanup, don't forget to call the \_image cleanup,



and on init don't forget to first call `_image`'s init.

So far, that was a lot of extra work. The real benefit comes when you come to implement the 24-bit image. You first take the 8-bit image object, and re-implement the routines that need re-implementing for the 24-bit image. The common procedures to both objects should go to `_image`.

But now, because the interface has been defined and tested beforehand on the 8-bit object, you are almost certain that when you start using the 24-bit object that it will work. That is the real benefit of this type of design. It leads to quite a bit of re-usability.

You can find all this and more in the ``pattern design" book (if I recall the title correctly). That book helped me reach the object nirvana :-)

But, maybe you guys did consider that approach.

Mirko

---

Subject: Re: Interactive Objects, Was: Simple GUI question  
Posted by [David Fanning](#) on Thu, 24 Apr 2003 17:25:28 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Mirko Vukovic (mvukovic@taz.telusa.com) writes:

> [Much deleted...]  
>  
> You can find all this and more in the ``pattern design" book (if I  
> recall the title correctly). That book helped me reach the object  
> nirvana :-)  
>  
> But, maybe you guys did consider that approach.

Well, something very much like that, yes. :-)

You are right about the book `_Design Patterns_` by Erich Gamma, et. al. Fabulous book and excellent bed-time reading for anyone writing object oriented programs in *\*any\** language.

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: Interactive Objects, Was: Simple GUI question  
Posted by [MC](#) on Fri, 25 Apr 2003 03:21:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

My wish list:

- 1) Whatever is available in object graphics is also made available in direct graphics. And leave the choice of style to the individual.
- 2) A book on object which is easy to understand. E.g. after reading about 10 pages, one have enough to start and be on his/her own thereafter. I started learning IDL that way.
- 3) Package all objects into icons which people like me can just drag and drop to create the GUI and function one needs.

Out to lunch or possible?

MC

---

---

Subject: Re: Interactive Objects, Was: Simple GUI question  
Posted by [David Fanning](#) on Fri, 25 Apr 2003 03:42:05 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

MC (markchan@shaw.ca) writes:

- > My wish list:
- >
- > 1) Whatever is available in object graphics is also made available in direct
- > graphics. And leave the choice of style to the individual.

We could start by making it possible to specify the length of minor tick marks, for God's sake!

- > 2) A book on object which is easy to understand. E.g. after reading about 10
- > pages, one have enough to start and be on his/her own thereafter. I started
- > learning IDL that way.

Uh, do you mind me asking whose book you were reading. :-)

- > 3) Package all objects into icons which people like me can just drag and
- > drop to create the GUI and function one needs.

Didn't RSI have a product like this? Dropped now, I guess.  
No one bought it. :-)

- > Out to lunch or possible?

Oh, you might get a book. Sooner or later. :-)

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: Interactive Objects, Was: Simple GUI question  
Posted by [Paul Van Delst\[1\]](#) on Fri, 25 Apr 2003 13:13:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

MC wrote:

- >
- > My wish list:
- >
- > 1) Whatever is available in object graphics is also made available in direct
- > graphics. And leave the choice of style to the individual.

<RANT>

Hmm. When I use IDL it's typically to look at data. That pretty much it. You know the usual stuff, plot, surface, contour, etc. And once I have the first plot up, I want to do things like change the scales (e.g. zoom), or rotate the surfaces - you know, play around with the numbers. I realised after a bit of a rant the other day that I don't really care \*how\* the images are created/displayed onscreen, I just want them to appear, and I want to be able to manipulate them.

Why can't the mechanism behind someone type PLOT on the IDL command line be totally transparent? When I plot something I certainly don't care if it was done via direct or object graphics, I just want to see the darn plot.

So, I have to admit it sort of confuses me that RSI introduced all this whiz bang object graphics stuff, but no "value-added" tools that use them. It's almost like buying a

mecchano (sp?) set - you get all the structural members and nuts and bolts and tools, but if you want anything interesting to play with ya gotta build it yourself. I don't want to write a whole bunch of code to do plots, surfaces, contours, whatever in object graphics so I can utilise their superior capabilities - that's what we pay so much moola for in the first place isn't it?

</RANT>

Off for more coffee....

paulv

--

Paul van Delst  
CIMSS @ NOAA/NCEP/EMC  
Ph: (301)763-8000 x7748  
Fax:(301)763-8545

---

---

Subject: Re: Interactive Objects, Was: Simple GUI question

Posted by [Rick Towler](#) on Fri, 25 Apr 2003 17:22:46 GMT

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

---

"Paul van Delst" wrote in message...

> MC wrote:

>>

>> My wish list:

>>

>> 1) Whatever is available in object graphics is also made available

>> in direct graphics. And leave the choice of style to the individual.

I think this is a bit much to wish for... What popular analysis/plotting packages still have an interface as dated as IDL's direct graphics? Are they gaining market share?

I think Paul's comments cut right to it. It doesn't matter how the plots are drawn on the screen, it just needs to be quick and easy.

> I don't really care \*how\* the images are created/displayed onscreen,  
> I just want them to appear, and I want to be able to manipulate them.

>

> Why can't the mechanism behind someone type PLOT on the IDL command  
> line be totally transparent? When I plot something I certainly don't  
> care if it was done via direct or object graphics, I just want to  
> see the darn plot.

>

> So, I have to admit it sort of confuses me that RSI introduced all  
> this whiz bang object graphics stuff, but no "value-added" tools

> that use them.

What is that saying? They didn't build Cleveland in an hour? Be patient, your "value" is coming sooner than you think :)

-Rick

---

---

Subject: Re: Interactive Objects, Was: Simple GUI question

Posted by [mvukovic](#) on Fri, 25 Apr 2003 19:57:26 GMT

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

---

Paul van Delst <[paul.vandelst@noaa.gov](mailto:paul.vandelst@noaa.gov)> wrote in message  
news:<3EA93481.88D1DE12@noaa.gov>...

> MC wrote:

>>

>> My wish list:

>>

>> 1) Whatever is available in object graphics is also made available in direct  
>> graphics. And leave the choice of style to the individual.

>

> <RANT>

> Hmm. When I use IDL it's typically to look at data. That pretty much it. You know the  
> usual stuff, plot, surface, contour, etc. And once I have the first plot up, I want to do  
> things like change the scales (e.g. zoom), or rotate the surfaces - you know, play around  
> with the numbers. I realised after a bit of a rant the other day that I don't really care  
> \*how\* the images are created/displayed onscreen, I just want them to appear, and I want to  
> be able to manipulate them.

>

> Why can't the mechanism behind someone type PLOT on the IDL command line be totally  
> transparent? When I plot something I certainly don't care if it was done via direct or  
> object graphics, I just want to see the darn plot.

>

> So, I have to admit it sort of confuses me that RSI introduced all this whiz bang object  
> graphics stuff, but no "value-added" tools that use them. It's almost like buying a  
> meccano (sp?) set - you get all the structural members and nuts and bolts and tools, but  
> if you want anything interesting to play with ya gotta build it yourself. I don't want to  
> write a whole bunch of code to do plots, surfaces, contours, whatever in object graphics  
> so I can utilise their superior capabilities - that's what we pay so much moola for in the  
> first place isn't it?

> </RANT>

>

> Off for more coffee....

>

> paulv

You could use ``live" tools. I tried, and never liked them very

much. The design, interface, and looks just struck me as clunky. But then again, maybe I did not give them enough of a chance.

Mirko

---

---

Subject: Re: Interactive Objects, Was: Simple GUI question  
Posted by [David Fanning](#) on Sat, 26 Apr 2003 03:08:33 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Rick Towler (rtowler@u.washington.edu) writes:

> What is that saying? They didn't build Cleveland in an hour? Be patient,  
> your "value" is coming sooner than you think :)

Yeah, Paul, I'd be sure my maintenance contract was up to date. :-)

Cheers,

David

--

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

---

---

Subject: Re: Interactive Objects, Was: Simple GUI question  
Posted by [btt](#) on Mon, 28 Apr 2003 14:14:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

- lots of interesting stuff snipped -

> An image data object, to give you  
> just one example, is particularly annoying. Should an image  
> base class transparently handle 8-bit and 24-bit images? Or  
> should these be two separate classes, primarily because they  
> are handled differently in processing steps? We have been  
> back and forth probably a thousand times on this one. It is  
> probably one of the few times when we are \*both\* right!  
>

Man,

It is soooo good to hear that I'm not the only one wrestling with this question. What a pain in the backside. I looked at a well developed public domain model (ImageJ) and decide there isn't a right answer. Gaak!

Cheers,

Ben

---

Subject: Re: Interactive Objects, Was: Simple GUI question

Posted by [mvukovic](#) on Mon, 28 Apr 2003 16:53:46 GMT

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

---

David Fanning <david@dfanning.com> wrote in message  
news:<MPG.1913a62044b38f37989b6f@news.frii.com>...

> Rick Towler (rtowler@u.washington.edu) writes:

>

>> What is that saying? They didn't build Cleveland in an hour? Be patient,

>> your "value" is coming sooner than you think :)

>

> Yeah, Paul, I'd be sure my maintenance contract was up

> to date. :-)

>

> Cheers,

>

> David

uh, uh, so when exactly is 6.0 coming out?

Getting impatient,

Mirko

---

Subject: Re: Interactive Objects, Was: Simple GUI question

Posted by [David Fanning](#) on Mon, 28 Apr 2003 17:08:15 GMT

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

---

Mirko Vukovic (mvukovic@taz.telusa.com) writes:

> uh, uh, so when exactly is 6.0 coming out?

I hear the beta is due out "very soon now".

The actual release follows by 6-8 weeks,  
usually.



Randall Skelton is suppose to get a look at it  
June 24th in the UK, if that gives you some idea. :-)

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

---