Subject: Re: IDL 6.0
Posted by Rick Towler on Thu, 27 Mar 2003 19:40:26 GMT
View Forum Message <> Reply to Message

"jlhall" wrote in message

> Does anyone have any substantial information about IDL 6.0? When is the
> planned release? What will the major differences be betwwen 6.0 and the
> current version?

The only official word I have seen is here:

http://www.rsinc.com/newsletter/index.asp?s=1203&wid=110 594

The email version of the newsletter had one extra tidbit:

"Keep your eye on our website for beta release information in the very near future."

-Rick

Subject: Re: IDL 6.0
Posted by Randall Skelton on Thu, 27 Mar 2003 19:55:47 GMT
View Forum Message <> Reply to Message

I don't know anything beyond what the UK sales rep has relayed in a message earlier this week. June 24th is the preview party at RAL and I'll be there. There is also a blurb at http://www.rsinc.com/newsletter/ (looks like rsi is updating their website).

My *hope* is that IDL 6 will extend the current object code base and support (1) operator overloading, (2) proper inheritance mechanisms (i.e. avoiding name space clashes that occur in structures), (3) public / private methods and (4) a C-API interface to object/heap variables. I'm also keen to see an object interface to map drawing...

Cheers,
Randall

On Thu, 27 Mar 2003, jlhall wrote:

> Does anyone have any substantial information about IDL 6.0? When is the
> planned release? What will the major differences be betwwen 6.0 and the
> current version?
>
> All I could find was this page in Chinese... which is still relatively

> mysterious after babelfish translation:
> http://astron.berkeley.edu/~jhall/export/IDL_6.0.pdf
>
> ---------- ------
> Joseph Lorenzo Hall jlhall@sonic.net
>
>

Subject: Re: IDL 6.0
Posted by David Fanning on Thu, 27 Mar 2003 20:02:51 GMT
View Forum Message <> Reply to Message

Randall Skelton (rhskelto@atm.ox.ac.uk) writes:

> My *hope* is that IDL 6 will extend the current object code base and
> support (1) operator overloading, (2) proper inheritance mechanisms (i.e.
> avoiding name space clashes that occur in structures), (3) public /
> private methods and (4) a C-API interface to object/heap variables. I'm
> also keen to see an object interface to map drawing...

Oh, dear. Maybe in 6.1. :-(

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: IDL 6.0
Posted by Randall Skelton on Thu, 27 Mar 2003 23:01:50 GMT
View Forum Message <> Reply to Message

>> My *hope* is that IDL 6 will extend the current object code base and
>> support (1) operator overloading, (2) proper inheritance mechanisms (i.e.
>> avoiding name space clashes that occur in structures), (3) public /
>> private methods and (4) a C-API interface to object/heap variables. I'm
>> also keen to see an object interface to map drawing...
>
> Oh, dear. Maybe in 6.1. :-(

OK, I realize that I am being a tad greedy here but I really do think that IDL objects are too limited in their current scope and it is beginning to become a real problem for me. I do a fair amount of multivariate analysis where a given vector or matrix is really a composite of various different data sets. I solve large systems of this sort and have a variety of routines in c/fortran that work on these composite objects efficiently. In many cases it is as simple as defining an object that holds a vector or array, with a hash table that describes what various parts of the vector or array correspond to, and some ancillary stuff about the measurements. I quite like operator overloading as it makes working with these composite objects (i.e. solving them as linear/non-linear systems) easy code and read. I do this all the time in c++, fortran and (recently) Matlab. I have some great object graphics routines written in IDL to visualize my data but I really have no way of actually operating on my data without resorting to writing very cryptic code. Try re-writing:

```
xh = xh + invert( (1+gamma) * SaI + Kts # k) # $
    ( Kts # (y_obs-y_f) - SaI # (x_h-x_a) )
```

when each addition is 'obj->add(a,b)', scalar multiplication is 'obj->shift(1)', and matrix multiplication 'obj->mm(A,B)'. IMHO, data encapsulation should not prevent people from actually working with their data in a logical way!

My message to RSI is that I don't really need any more GUI tools, spread-sheet views or slow pointy-clicky interfaces. What I would like is a more complete set of core object programming interfaces so I can program more logically and manipulate/visualize my data more efficiently. I hate to admit it, but I have been using Matlab a lot recently as it's object model is more suited to how I code using data objects. Now if only it had pointers :(

I promise this will be my last 'IDL objects' rant for a while.

Cheers,
Randall

Subject: Re: IDL 6.0
Posted by David Fanning on Fri, 28 Mar 2003 00:30:07 GMT
View Forum Message <> Reply to Message

Randall Skelton (rhskelto@atm.ox.ac.uk) writes:

> OK, I realize that I am being a tad greedy here but I really do think that
> IDL objects are too limited in their current scope and it is beginning to
> become a real problem for me.

I'm not disagreeing with you at all, Randall. Just doing
my bit to set your expectations appropriately. :-)

> My message to RSI is that I don't really need any more GUI tools,
> spread-sheet views or slow pointy-clicky interfaces.

Let's just say I can't wait to hear your report on the
big RAL shindig in June. It should be ah... interesting. :^)

I will say this, though, and I don't think it breaks
any confidences. I've had a peak at a couple of things
that make my heart beat a whole lot faster. But maybe
I'm just young and in love again, now that it's Spring.

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: IDL 6.0
Posted by mmiller3 on Fri, 28 Mar 2003 00:33:04 GMT
View Forum Message <> Reply to Message

>>>> > "Randall" == Randall Skelton <rhskelto@atm.ox.ac.uk> writes:

> My message to RSI is that I don't really need any more GUI
> tools, spread-sheet views or slow pointy-clicky interfaces.
> What I would like is a more complete set of core object
> programming interfaces so I can program more logically and
> manipulate/visualize my data more efficiently.

Here here!

That and access to command line arguments like every other
language on the planet would go a long way to streamlining my IDL
use.

Mike

Subject: Re: IDL 6.0

Posted by David Fanning on Fri, 28 Mar 2003 00:53:28 GMT
View Forum Message <> Reply to Message

David Fanning (david@dfanning.com) writes:

> I will say this, though, and I don't think it breaks
> any confidences. I've had a peak at a couple of things
> that make my heart beat a whole lot faster. But maybe
> I'm just young and in love again, now that it's Spring.

Chairman Coyote is obviously intercepting my messages
again, and adding this sexual innuendo. Of course, I
wrote "peek" in my original message. :-(

Cheers,

David

P.S. Let's just say I was going to say something
sarcastic about Mike Miller's "Here, here", but
decided against it on further reflection. :-(

--
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: IDL 6.0
Posted by Michael Asten on Fri, 28 Mar 2003 01:55:44 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:

>
> My message to RSI is that I don't really need any more GUI tools,
> spread-sheet views or slow pointy-clicky interfaces. What I would like is
> a more complete set of core object programming interfaces so I can
> program more logically and manipulate/visualize my data more efficiently.
> I hate to admit it, but I have been using Matlab a lot recently as it's
> object model is more suited to how I code using data objects. Now if only
> it had pointers :(
>

This viewpoint shows an interesting divergence between different classes of
users. I am of the troglodyte school who do not use objects (do I hear a Coyote

howling from the next hill?). But I do use guis to drive my programs, and I dont like programming guis - I prefer to use ready-made items; I use IDL to get quick results from my data, not to generate fancy code.

I have found (somewhat to my chagrin) that when I have tried to get professional programmers interested in using idl for all the obvious reasons, they wont touch it; a 4GL is beneath them, and anything idl can do, c++/java can do better etc etc. Which is fine if the outcome desired is a versatile customised piece of software. But if the criterion is to get code to process the data, in half the programming time, at half the development price, then the said pro approach is a lousy answer. The bottom line in my experience is is, those who want the finest software tools may as well follow the pros to c++/java. The fraction of users who like idl for its reduced software development time, will probably say with me, bring on better/more versatile libraries of guis and gui-builders.

Regards,
Michael Asten

>
> I promise this will be my last 'IDL objects' rant for a while.
>
> Cheers,
> Randall
