Subject: Re: Object epiphany: A new way of building widget applications
Posted by davidf on Wed, 04 Apr 2001 22:01:00 GMT
View Forum Message <> Reply to Message

Martin Schultz (martin.schultz@dkrz.de) writes:

> PS: A great big thank you to David again for providing all this great
> stuff on his web page - and for developing MPI_Plot which served as a
> basis for the development of my programs.

Boy, I'll tell you, if I had a nickel for every
idea I've had that somebody turned into something better,
I wouldn't be writing IDL code today. :-(

But this sounds great, Martin. I'm looking forward to
exploring it. And I guess I'm getting more serious about
an "object" book. More and more people are getting their
feet wet, but most of us still need some hand-holding to
take the plunge. Maybe we can use this as the "base
structure" for developing object applications.

Cheers,

David

P.S. Let's just say that rich is probably better, but
I've no objections to being known as the "idea man".
It's better than a lot of other things I've been called. :-)

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

Subject: Re: Object epiphany: A new way of building widget applications
Posted by John-David T. Smith on Wed, 04 Apr 2001 22:40:47 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
>
> Hi all,
>
> With almost a week delay, I finally get around to release the first
> version of a new class of IDL objects: the MGS_GUIObject hierarchy.

I think it only fair to let people know that I tend to shy away from
distributed code with people's initials in the name. I know, it sounds
stupid, but I'm not sure I'm the only one. It seems to be a reasonably
common practice here (Craig, you listening?), but one which I think
might be best to avoid, for the following reasons:

1. It conveys a sense of ownership or heavy expectations that are
perhaps unjustified, and not intended. (Can I *change* such a routine,
should I feel guilty, etc...).

2. It takes up space in a name which could perfectly well have been
used for more descriptive characters.

3. If the routine/class/function/widget name following, e.g., JDS, is
so ambiguous as to require the initials to discriminate it from another
of the same name, either the routine/class/function/widget isn't that
useful, or its name is entirely too inspecific. And the way I think
about it, since IDL doesn't do any shadow checking (but cf. idlwave!),
the *best* routine with a given generic name will rise to the top.

4. The author(s) can always be found in a proper documentation header.

That's just my feeling on it. Anyone else have an opinion? I could
suggest lots of descriptive names for this class. SuperGUI?
GUIMaster? WidgetMaster? GUIBuilder? (no pun intended, RSI)...

JD

Subject: Re: Object epiphany: A new way of building widget applications
Posted by davidf on Wed, 04 Apr 2001 23:07:21 GMT

View Forum Message <> Reply to Message

JD Smith (jdsmith@astro.cornell.edu) writes:

> I think it only fair to let people know that I tend to shy away from
> distributed code with people's initials in the name. I know, it sounds
> stupid, but I'm not sure I'm the only one. It seems to be a reasonably
> common practice here (Craig, you listening?), but one which I think
> might be best to avoid, for the following reasons:
>
> That's just my feeling on it. Anyone else have an opinion?

Yes. I've tried it both ways, and it's pretty much
a nightmare without some way to tag it to the author,
although I agree with you that it's extra baggage.

It does make it easy to know that you are using
someone's else's library programs, and whose it is.
You have to remember that more users than you would
ever expect really don't have *any* idea that there is
a difference between a built-in routine and and library
routine and how to order their Path. (And, really,
why *should* they care?)

But ownership is not such a bad idea. You at least
have someone to complain to when things don't work
as you expect. You are, of course, free to modify
any code you get on a free newsgroup. And all these
"initialized" programs are distributed under the Open
Software Foundation license, so you can do whatever
you like to with them, even sell them if you think
you can make any money doing so. (Good luck!) I think
all the authors make extraordinary attempts to keep the
code up to date and maintained. I know that most of the
time I've reported a bug or made a suggestion I had new
code within 24 hours.

I feel a little badly when I see Martin (to take just
the most recent example) essentially duplicating some
of my code. But once you put this stuff in the public
domain you just have to let it go. My library is mostly
maintained for me, and for the work I do for my own
clients. I really don't have the time to worry about
what happens to it later. (And, heck, I've already
got a better idea and I'm on to something else anyway. :-)

But, please, name the programs anything you like! Speaking
for myself, I'm just thrilled to know you are using them.

Cheers,

David

P.S. I've tried to get people to make changes to some of
my programs, but many people prefer to have me maintain
the library. That works, of course, until I decide to retire. :-)

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

Subject: Re: Object epiphany: A new way of building widget applications
Posted by Mark Hadfield on Wed, 04 Apr 2001 23:18:04 GMT

View Forum Message <> Reply to Message

"JD Smith" <jdsmith@astro.cornell.edu> wrote in message
news:3ACBA2EF.493F496F@astro.cornell.edu...
> Martin Schultz wrote:
>>
>> With almost a week delay, I finally get around to release the first
>> version of a new class of IDL objects: the MGS_GUIObject hierarchy.
>
> I think it only fair to let people know that I tend to shy away from
> distributed code with people's initials in the name. I know, it sounds
> stupid, but I'm not sure I'm the only one. It seems to be a reasonably
> common practice here (Craig, you listening?), but one which I think
> might be best to avoid, for the following reasons:

As one of the pioneers of this trend (he says modestly) may I present the
opposing viewpoint:

It's namespace management, pure and simple. It's desirable because IDL lacks
built-in facilities.

> 1. It conveys a sense of ownership or heavy expectations that are
> perhaps unjustified, and not intended. (Can I *change* such a routine,
> should I feel guilty, etc...).

Well, it's not *meant* to convey this.

> 2. It takes up space in a name which could perfectly well have been
> used for more descriptive characters.

Unfortunate, but true.

> 3. If the routine/class/function/widget name following, e.g., JDS, is
> so ambiguous as to require the initials to discriminate it from another
> of the same name, either the routine/class/function/widget isn't that
> useful, or its name is entirely too inspecific.

Didn't you have a "queue" & a "stack" class on your WWW page? Well. someone
did. I have written a stack class too. I call it "mgh_stack". It's a fairly
inspecific class, so I give it a suitably inspecific name and add my
initials on the front. What should I call it,
"this_is_a_different_stack_from_jds_one"? Or
"general_purpose_stack_with_whizzy_get_and_put_methods"?

> And the way I think
> about it, since IDL doesn't do any shadow checking (but cf. idlwave!),
> the *best* routine with a given generic name will rise to the top.

The one that rises to the top is somewhat unpredictable. (Well, strictly speaking it's predictable because yuu can control your PATH, though I have noticed recently that Windows 2000 expands path entries preceded by + in *reverse* alphabetical order, which caused me some grief.) The thing is, I don't remember exactly what is where on my PATH and I don't like relying on the search order. I have been bitten by duplicated routine names a number of times: CALDAT and CREATE_STRUCT are two I can remember.

> 4. The author(s) can always be found in a proper documentation header.

Sure, but it's not about claiming ownership, it's about namespace management.

But hey, there's room for all points of view. If you don't prepend your initials and I do, then our routine names will never clash.

Is there any other MGH out there?

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

Subject: Re: Object epiphany: A new way of building widget applications
Posted by John-David T. Smith on Thu, 05 Apr 2001 03:03:38 GMT

View Forum Message <> Reply to Message

Mark Hadfield wrote:
>
> "JD Smith" <jdsmith@astro.cornell.edu> wrote in message
> news:3ACBA2EF.493F496F@astro.cornell.edu...
>> Martin Schultz wrote:
>>>
>>>   With almost a week delay, I finally get around to release the first
>>> version of a new class of IDL objects: the MGS_GUIObject hierarchy.
>>
>> I think it only fair to let people know that I tend to shy away from
>> distributed code with people's initials in the name. I know, it sounds
>> stupid, but I'm not sure I'm the only one. It seems to be a reasonably
>> common practice here (Craig, you listening?), but one which I think
>> might be best to avoid, for the following reasons:
>
> As one of the pioneers of this trend (he says modestly) may I present the
> opposing viewpoint:
>
> It's namespace management, pure and simple. It's desirable because IDL lacks
> built-in facilities.

>> And the way I think
>> about it, since IDL doesn't do any shadow checking (but cf. idlwave!),
>> the *best* routine with a given generic name will rise to the top.
>
> The one that rises to the top is somewhat unpredictable. (Well, strictly
> speaking it's predictable because yuu can control your PATH, though I have
> noticed recently that Windows 2000 expands path entries preceded by + in
> *reverse* alphabetical order, which caused me some grief.) The thing is, I
> don't remember exactly what is where on my PATH and I don't like relying on
> the search order. I have been bitten by duplicated routine names a number of
> times: CALDAT and CREATE_STRUCT are two I can remember.

I of course am very sensitive to this notion, which is why Carsten and I developed an effective method for dealing with it in IDLWAVE. But in any case, I was merely speaking metaphorically. If I write a routine called "stack", and you write a routine called "stack", one or the other will probably come into dominant usage. Is this ideal? No. Should we attempt to relieve namespace collision by thinking ahead? Certainly.

>> 4. The author(s) can always be found in a proper documentation header.
>
> Sure, but it's not about claiming ownership, it's about namespace
> management.
>
> But hey, there's room for all points of view. If you don't prepend your
> initials and I do, then our routine names will never clash.
>
> Is there any other MGH out there?

If you need a prefix to differentiate your namespace, then by all means, chooose one. I was just arguing against that prefix being your initials. Here is an decent argument, simultaneously *for* namespace management, and *against* using your initials:

http://tiny-tools.sourceforge.net/emacs-code-body.html#about _lisp_symbol_naming

It's for lisp, but the same arguments apply. It's also pretty simplisitic, but the basic tone captures it I think. So, for your example, suppose you have a stack class which is fairly general. Why not super_stack, or fast_stack, or objStacker, etc? Yes, IDL started this whole ball rolling with their IDL_Blah series of classes, but I guess I just feel like a more open approach is available to us here.

If I were a company, JD, Inc., I would give my products a strong brand identity: JDI_Widget.pro. I'm not a company, and for this I'm glad. I don't make money from the things I contribute, nor do I guarantee their utility. If they solve your problem, great. If you rip them into pieces to something altogether different with them, great.

I'm not saying you *shouldn't* brand your contributions in the same way,
but just pointing out a (perhaps not universal) connotation that
branding engenders.

JD

Subject: Re: Object epiphany: A new way of building widget applications
Posted by Martin Schultz on Thu, 05 Apr 2001 08:43:54 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
>
>> With almost a week delay, I finally get around to release the first
>> version of a new class of IDL objects: the MGS_GUIObject hierarchy. ...
>
> I think it only fair to let people know that I tend to shy away from
> distributed code with people's initials in the name. I know, it sounds
> stupid, but I'm not sure I'm the only one. It seems to be a reasonably
> common practice here (Craig, you listening?), but one which I think
> might be best to avoid, for the following reasons:
>
Well, to be honest, it makes me a little sad that there are several
articles in this thread, but none dealing with the software itself,
and all sidetracking on a somewhat futile discussion about initials
and namespace. (but thanks Craig for your email)

But I feel I need to take up this topic anyhow, and I disagree with
you, JD.

> 1. It conveys a sense of ownership or heavy expectations that are
> perhaps unjustified, and not intended. (Can I *change* such a routine,
> should I feel guilty, etc...).
I think you have the wrong feelings here. Mine are rather along the
lines of "I am using fsc_...; hey - that's a great piece of code, this
guy is worth remembering". And when I want to change things, I just
do, because he explicitly allows me to do so in his license. And then
I can rename it to mgs_... because then I know it's my version, and if
something goes wrong, I am to blame first.

>
> 2. It takes up space in a name which could perfectly well have been
> used for more descriptive characters.
I'm not sure of this. I think, Mark hit the ball here saying that
"descriptive" prefixes tend to produce namespace conflicts. And in a
way, the author's initials are also descriptive: they tell you who
wrote the piece even if you are currently not in idlwave mode and

simply look at a directory listing.

>
> 3. If the routine/class/function/widget name following, e.g., JDS, is
> so ambiguous as to require the initials to discriminate it from another
> of the same name, either the routine/class/function/widget isn't that
> useful, or its name is entirely too inspecific. And the way I think
> about it, since IDL doesn't do any shadow checking (but cf. idlwave!),
> the *best* routine with a given generic name will rise to the top.

That's completely idealistic. Hey, JD, I wouldn't have expected such an unreflected thought from you. Besides the create_struct ... problem, just think of the famous colorbar routine. Do you know which one I am referring to? I know of at least 4 of these, and they all have their specific merits. The only "solution" in that sense would probably be that RSI selected one of the conflicting routines to incorporate it in the IDL distribution and maintain it. But I know for sure they are not up to it (and I feel sympathetic).

Also, if I see mgs_colorbar, ... in my routine, I know where to find the source code: it's in my library. If I see fsc_color, it's in lib/david_fanning. This often proves very valuable - even considering the fact that idlwave has these nice features. But idlwave is not the only editor that people are using (unfortunately).

>
> 4. The author(s) can always be found in a proper documentation header.

True, too.

Another thought about this software and the ownership: I had given it several thoughts how to properly acknowledge David for the invaluable "contribution" he made to these programs. Well,
(1) I definitively wanted to change the name, because mgs_field just doesn't work like fsc_field, and you can't use it as a drop-in replacement.
(2) I considered writing David and me down together as co-authors, but that could mean that he will receive the emails from people asking for help, and I am not sure he wants to do it ;-)
(3) I felt that, although significant portions of David's code have entered my programs verbatim (especially in mgs_field and mgs_drawcolor), there have been changes that are substantial enough to claim that these are new programs (several of them are completely new anyway).
(4) I am trying to make sure that David's name appears at least twice in every of these programs ;-) Please tell me if it doesn't.

I encourage everyone of you to do the same (exploiting software via copy and paste) with my programs: that's why they are open software! In fact, Ben came up with this twolist object only a day after I sent a preliminary version to him; and I certainly don't feel bad about him taking pieces of my code. Rather the opposite: I am convinced that this is the fastest and most efficient way to write (object widget) programs: Take pieces from available software and build it together to produce what you need. If that really would produce bad feelings, even fewer people would write and use widgets (or at least no one would talk about them for fear of being convicted for copyright infringement).

Now, we recently had this message about a universal file reader that should be written by the community and be available to the community. I wouldn't have any objections if something like the mgs_...object hierarchy became the basis of a community object library; and I am well aware that this would mean certain standards for code maintenance such as : don't change the interface lightheartedly, don't change the functionality, ... I would be willing to host a library of objects built upon this hierarchy, and - of course - they should then have a common "class" name prefix (perhaps simply "N_" for newsgroup, or "NN_" for anonymous). But I am not too optimistic that you people will jump on this, to be honest. If so, the idea would be to establish a developers mailing list and to introduce some sort of version control with dedicated releases to the public.

But now, please take a look at the programs even though they are called mgs_... and tell me what you think.

Cheers,

Martin

--
[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[

Subject: Re: Object epiphany: A new way of building widget applications
Posted by John-David T. Smith on Thu, 05 Apr 2001 14:02:25 GMT
View Forum Message <> Reply to Message

OK, I feel I need to clarify just a bit more. I didn't mean to hit such a sensitive nerve. I am not *at all* questioning the motives and qualities of the software being released with the naming convention in question. Not even sort of. I appreciate, just like everyone else, the ability to pick through outstanding pieces of work coded by talented and dedicated individuals like yourself. Heck, sometimes I even use contributed routines as-is ;) So I'm sorry if my opinion cut too deep.

Maybe it was a poor choice of timing, but all I really wanted to do was alert contributors of a potential hidden meaning their naming style might convey to a certain (apparently small) portion of their intended audience. I did not mean to appear critical.

As far as the question of inheriting and updating older code (like FSC_FIELD), this is a trickier issue. One thing that is certain to confuse users, however, is multiple *_FIELD routines out there. Which do I use?

Sometimes this is unavoidable. But consider an example. Suppose in some alternate universe I was good enough at lisp to redo the IDLWAVE mode of emacs substantially to suit my own peculiar needs (not that Carsten doesn't bend over backwards to do that now... but this is hypothetical...). I have two options.

1. Fork the codebase, rename the mode JD-IDLWAVE, and begin distributing a competing version.

2. Talk to Carsten about the features I'd like to see, and come to some agreement about who should really be maintaining it, given the new information.

Version 2 is much trickier, obviously. It takes cooperation, potential conflict resolution, and perhaps compromise. If JD-IDLWAVE were for internal consumption, it probably would not be worth it. But, if JD-IDLWAVE were intended to benefit the larger community, the experience of "real" programmers (unlike myself) has shown that a consistent, continuously evolving work will last longer, receive more attention, and ultimately benefit everyone more, despite the transition to a new maintainer/coder. Fractionation confuses. Unification encourages.

Anyway, these are just very different approaches. And now, back to your regularly scheduled program (and sorry for the interruption).

I close with a truer wisdom:

Honest disagreement is often a good sign of progress.
--Mahatma Gandhi (1869 - 1948)

JD

Subject: Re: Object epiphany: A new way of building widget applications
Posted by Ben Tupper on Thu, 05 Apr 2001 14:11:03 GMT

View Forum Message <> Reply to Message

Hello,

Martin Schultz wrote:

> Hi all,
>
> With almost a week delay, I finally get around to release the first
> version of a new class of IDL objects: the MGS_GUIObject hierarchy.
> Don't shy away immediately! It's far easier than it sounds, and once
> you will have discovered how easy it now becomes to develop widget
> applications, you will get hooked! Ben Tupper has managed to get
> something running within a day.
>

Now, I could take that as a compliment on my programming prowess or I could take that as testimony of the ease of basing widgets on MGS_GUIObject. I know my programming skills well enough to know that it is the latter.

I have migrated to using objects for all items that are GUIs or serve some form of a databasing function. Perhaps I have been an easy sell for Martin because I got hooked on his earlier MGS_CONTAINER object (I rarely use IDL_CONTAINER... and when I do, I always regret it and change it to Martin's subclass.) I have switched to object coding because the people I work for often change their minds about what is needed. The object style coding is perfect for this situation.

Key items for me (non-programmer/science-lackey/guinea pig) are:

1. It is way too easy. When I think of the hours I have spent building GUIs from scratch that could have been done in hours... ugh. How many times have I written the event handlers for GUIs to handle cleaning up the widget, sending events to the proper procedures/methods. It seemed like an nightmarish repetition that varied only slightly in the details. Here it is done once with a uniform nomenclature and a standard operating procedure.

2. I can treat it as a black box. I don't have to know how it does what

it does.   I don't need to fiddle-faddle with all of the MODAL, BLOCKING,
CLEANUP, etc. knobs that make xMANAGER control the GUIs they way I expect
(hope).   All of those are handle by the base gui object behind the
scenes.

3. I can open the black box and override any of the hidden workings (see
#2 above) with out changing the base object code. I can have a special
case without building exception handling into the base (object) code. (#2
and #3 give me the best of both worlds!)

Object-based GUIs are worthy of a serious look by anyone who needs GUIs
but hates to (re)write them. And while your at it, there is no point in
reinventing the wheel, try MGS_GUIobject.

Ben
--
Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net

Subject: Re: Object epiphany: A new way of building widget applications
Posted by davidf on Thu, 05 Apr 2001 15:03:52 GMT

View Forum Message <> Reply to Message

Ben Tupper (pemaquidriver@tidewater.net) writes:

> Object-based GUIs are worthy of a serious look by anyone who needs GUIs
> but hates to (re)write them. And while your at it, there is no point in
> reinventing the wheel, try MGS_GUIobject.

While we are on the subject of naming conventions, the
problem with this name is that is has "object" in it,
not that it is too long because of the pre-pended initials.

I should think MSG_GUIThingy would have attracted more
converts immediately. :-)

Cheers,

David

P.S. Let's just say the next person who says "objects
are easy" is going to be fined by the IDL Expert Programmers

Association for releasing our deepest held secrets!

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

Subject: Re: Object epiphany: A new way of building widget applications
Posted by Ben Tupper on Thu, 05 Apr 2001 15:28:25 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

>
> While we are on the subject of naming conventions, the
> problem with this name is that is has "object" in it,
> not that it is too long because of the pre-pended initials.
>
> I should think MSG_GUIThingy would have attracted more
> converts immediately. :-)

You know, there is merit to a name like MGS_GUIBase... it certainly conveys
the right spirit of this thingy.

Perhaps a rose by any other name still must survive among the thorns.

>
>
> P.S. Let's just say the next person who says "objects
> are easy" is going to be fined by the IDL Expert Programmers
> Association for releasing our deepest held secrets!

Oops.

Ben

P.S. on second thought, I'm not going to worry about a fine; after all, I'm
still waiting for my IDL_EP junior membership card and coffee mug.

--
Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net

Subject: Re: Object epiphany: A new way of building widget applications
Posted by Martin Schultz on Thu, 05 Apr 2001 15:33:18 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

>
> P.S. Let's just say the next person who says "objects
> are easy" is going to be fined by the IDL Expert Programmers
> Association for releasing our deepest held secrets!
>

OK. I'll accept the fine -- if it amounts to a couple of beers in a
few weeks from now ;-)

Cheers,

Martin

PS: This message is of course not meant to be archived, but where the
hell do I find an option in Netscape to set X-No-Archive to Yes ???

```
--
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
[[ Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie    [[
[[                      Bundesstr. 55, 20146 Hamburg             [[
[[                      phone: +49 40 41173-308                  [[
[[                      fax:   +49 40 41173-298                  [[
[[ martin.schultz@dkrz.de                                        [[
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
```

Subject: Re: Object epiphany: A new way of building widget applications
Posted by davidf on Thu, 05 Apr 2001 16:30:09 GMT

View Forum Message <> Reply to Message

Ben Tupper (pemaquidriver@tidewater.net) writes:

> P.S. on second thought, I'm not going to worry about a fine; after all, I'm
> still waiting for my IDL_EP junior membership card and coffee mug.

The leadership is so damn lax, nobody follows through
on *anything*. :-(

Cheers,

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

Subject: Re: Object epiphany: A new way of building widget applications
Posted by Ben Tupper on Thu, 05 Apr 2001 17:02:34 GMT

View Forum Message <> Reply to Message

They are pretty quick to cash a check though!

David Fanning wrote:

> Ben Tupper (pemaquidriver@tidewater.net) writes:
>
>> P.S. on second thought, I'm not going to worry about a fine; after all, I'm
>> still waiting for my IDL_EP junior membership card and coffee mug.
>
> The leadership is so damn lax, nobody follows through
> on *anything*. :-(
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438 E-Mail: davidf@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

--
Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net

Subject: Re: Object epiphany: A new way of building widget applications
Posted by Craig Markwardt on Thu, 05 Apr 2001 18:55:31 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@astro.cornell.edu> writes:

> Martin Schultz wrote:
>>
>> Hi all,
>>
>>   With almost a week delay, I finally get around to release the first
>> version of a new class of IDL objects: the MGS_GUIObject hierarchy.
>
> I think it only fair to let people know that I tend to shy away from
> distributed code with people's initials in the name. I know, it sounds
> stupid, but I'm not sure I'm the only one. It seems to be a reasonably
> common practice here (Craig, you listening?), but one which I think
> might be best to avoid, for the following reasons:
>
... remainder deleted ...

Hi JD--

I understand what you are saying, but I think you are a little too harsh in criticizing other people for how they name their functions, especially when Martin's code is as cool as it sounds.

I for one do not mind using other people's code which has their initials on it. As has been pointed out before, this helps keep the name spaces separated. We've had enough problems where RSI bonks into user-code with their proprietary or internal functions, that using a little something special appears legitimate to me.

As for my own code, I don't follow any strict naming rules. Generally speaking I tack a "CM" on when I am duplicating the function of somebody else's code (CMPS_FORM, CMHISTOGRAM), or where there is a *potential* for a later clash (CMAPPLY).

Sometimes it's just a namespace issue so I can keep the functions sorted in my brain (all of the MPFIT functions are based on MINPACK, hence the "MP" on the front).

Lastly, when I know that procedures will be typed from the command line I try to make them short and sweet. I picked the "DX" commands to be short, and also because "D" and "X" were near each other on the keyboard. Ergonomics baby!

Craig

--
-- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Duplicate module names. Was: Object epiphany: ...
Posted by Kristian Kjaer on Thu, 19 Apr 2001 18:59:56 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
> I think it only fair to let people know that I tend to shy away from
> distributed code with people's initials in the name....

Fancy that!
I was playing with the thought of posting a request: Would all those generous providers of public IDL code please edit all their code so that no module names are duplicated from one library to the next ...

Anyway, thanks for all the code!
Kristian Kjær, Risø Natl. Laboratory, Denmark