
Subject: Re: Copying an object
Posted by [davidf](#) on Tue, 31 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

> The best way I have found is to SAVE it then RESTORE it. My routine
> MGH_OBJ_CLONE provides a convenient way to do this.
>
> http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_obj_clone.pro
>
> It sounds kludgy but it works surprisingly well. I have used it widely &
> successfully in my OG code for a year now. Just make sure the IDLgrModel in
> question is not a child of any other IDLgrModel or IDLgrView at the time you
> clone it.

How do you get around the problem, Mark, of restoring
an object and calling methods on it without first
compiling the object method code? Or does your
program design shield you from this problem?

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: Copying an object
Posted by [Liam Gumley](#) on Tue, 31 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning <davidf@dfanning.com> wrote in message
news:MPG.1235bac4bdaa96e89898d1@news.frii.com...
> P.S. Let's just say if Liam *can* write something, I'd be
> very interested in seeing it. :-)

Don't hold your breath ;-)

Cheers,
Liam.

Subject: Re: Copying an object
Posted by [davidf](#) on Tue, 31 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Pavel Romashkin (promashkin@cmdl.noaa.gov) writes:

- > Since object structures are heap variables, couldn't a wizard like David or
- > Liam hack a small piece of code to allow unauthorized access to object
- > structures directly through pointers? Something like
- > a = ptr_valid(Obj_heap_var_number, /cast) ? This would be dangerous but
- > probably useful because then duplication would be a one liner.

Well, probably Liam can, but I'm left scratching my head over what seem to me to be insurmountable problems. Randy Frank, who wrote much of this object code when he was at RSI, has sent me a private e-mail outlining even more problems than I was aware of on my own. For example, he points out that many times the model objects (which is what we were discussing earlier this morning) have parents. What should be done about those? Should the copy have the **same** parents, or any parents? What implications does this have if the parents are destroyed or changed in some way? You would have to answer this question for yourself. It is unlikely that a general algorithm could be written.

Of course, if the GetProperties and SetProperties methods of the object are well written, you could simply do something like this:

```
copyObj = Obj_New('OurObjectClass')
oldObj->GetProperty, All=allSettableProperties
copyObj->SetProperty, _Extra=allSettableProperties
```

But in practical applications (at least the ones I've written) this doesn't always work.

You can also try SAVEing and RESTOREing your objects, but this has the problem that sometimes the methods of restored objects can't be found. (See JD Smith's comments on this on my web page.)

And yet, I find that I need a copy of the object quite often. I haven't found a perfect solution yet, but something like the solution outlined on my web page is where I usually start from. So far, I've been able to hack my way around some kind of solution in every case.

Cheers,

David

P.S. Let's just say if Liam *can* write something, I'd be very interested in seeing it. :-)

--

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: Copying an object

Posted by [davidf](#) on Tue, 31 Aug 1999 07:00:00 GMT

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

Waleed Al-Nuaimy (asger@gsiukltd.freemove.co.uk) writes:

- > Hi. How may one create an exact copy of an object?
- >
- > In particular, I have an IDLgrModel containing other Models each with a
- > polygon object inside, and each little model is subjected to a number of
- > scale/translation/rotations. I want to create another Model identical to
- > this in all respects. Is there a direct way to do this?

Here is one way:

http://www.dfanning.com/tips/copy_objects.html

Cheers,

David

P.S. I'm exploring the possibility of having an European mirror site for my web page, so that you folks on the other side of the pond will have better access. :-)

--

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: Copying an object
Posted by [steinhh](#) on Wed, 01 Sep 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <MPG.1236037e91c3d0539898d6@news.frii.com>
davidf@dfanning.com (David Fanning) writes:

>> The best way I have found is to SAVE it then RESTORE it. My routine
>> MGH_OBJ_CLONE provides a convenient way to do this.
> [...]
> How do you get around the problem, Mark, of restoring
> an object and calling methods on it without first
> compiling the object method code? Or does your
> program design shield you from this problem?

In this case, we're just cloning an **existing** object. Whatever
problems were associated with undefined methods for the original
will apply to the clone. But no **additional** problems..

Regards,

Stein Vidar

Subject: Re: Copying an object
Posted by [Mark Hadfield](#) on Wed, 01 Sep 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning <davidf@dfanning.com> wrote in message
news:MPG.1235bac4bdaa96e89898d1@news.frii.com...

>
> ...insurmountable problems. Randy Frank,
> who wrote much of this object code when he was at RSI,
> has sent me a private e-mail outlining even more problems
> than I was aware of on my own. For example, he points
> out that many times the model objects (which is what
> we were discussing earlier this morning) have parents.
> What should be done about those? Should the copy have
> the **same** parents, or any parents?

Remove the model object from its parent, copy it, and add it again
afterwards. Or am I missing something?

Entirely general solutions are very hard to come by in the object-oriented
world but if we know that an object is, say, an IDLgrModel, then we know it
has one parent (at most), we can get a reference to the parent from the
child's PARENT property, and we know the parent has Add and Remove methods.
Of course if I wanted to be perverse I could write an object that inherits

from IDLgrModel (so returns true for OBJ_ISA(...,'IDLgrModel')) but doesn't obey these rules. But I don't want to be perverse.

> You can also try SAVEing and RESTOREing your objects,
> but this has the problem that sometimes the methods of
> restored objects can't be found. (See JD Smith's comments
> on this on my web page.)

That can certainly be a problem if you RESTORE an object that was SAVED in a previous session, but surely it is not an issue in the present case, where we want to make a copy of an object (or object tree) that was created in the present session. The copy operation needn't change any of the object definitions, method definitions or object-method bindings.

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

Subject: Re: Copying an object
Posted by [Mark Hadfield](#) on Wed, 01 Sep 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Following up my own post! Well, someone has to do it.

I forgot to note that MGH_OBJ_CLONE calls another of my routines, RANDOM_NAME (which I may well rename to MGH_RANDOM_NAME some day). They can be found at the following URLs

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_obj_clone.pro
http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/random_name.pro

and are also included below

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

```
;+
; NAME:
;   MGH_OBJ_CLONE
;
; PURPOSE:
;   Given a reference to an object heap variable, this function generates
```

```

; a copy and returns its reference.
;
; CALLING SEQUENCE:
; Result = MGH_OBJ_CLONE(Object)
;
; INPUTS:
; Object:      An object reference.
;
; RESTRICTIONS:
; Cloning child objects is not advisable because it leads to
inconsistencies
; between the parent's and child's information about the relationship.
;
; MODIFICATION HISTORY:
; Mark Hadfield, Feb 1998:
;   Written as OBJ_CLONE.
; Mark Hadfield, Sep 1998:
;   Renamed MGH_OBJ_CLONE.
;-
function MGH_OBJ_CLONE, Object, VERBOSE=verbose

    if 1-obj_valid(Object) then return, obj_new()

    ; Copy the reference to a new variable so that the original will
    ; not be overwritten.

    clone = object

    file = filepath(random_name()+'.idl_object', /TMP)

    if keyword_set(verbose) then message, /INFORM, 'Saving object to file
'+file

    save, clone, FILE=file

    restore, file

    if keyword_set(verbose) then message, /INFORM, 'Deleting file '+file

    file_delete, file

    return, clone

end

;+
; NAME:
; RANDOM_NAME

```

```
;
;
; PURPOSE:
; This function returns a random 8-character string which is very unlikely
; to be repeated.
;
; CALLING SEQUENCE:
; Result = RANDOM_NAME()
;
; MODIFICATION HISTORY:
; Mark Hadfield, April 1997:
;   Written.
;-
function RANDOM_NAME

    common random_name_common, seed

    return, strtrim( string((randomu(seed,1))[0]*2L^31, FORM = '(Z)'), 2)

end
```

Subject: Re: Copying an object
Posted by [Mark Hadfield](#) on Wed, 01 Sep 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

> Hi. How may one create an exact copy of an object?
>
> In particular, I have an IDLgrModel containing other Models each with a
> polygon object inside, and each little model is subjected to a number of
> scale/translation/rotations. I want to create another Model identical to
> this in all respects. Is there a direct way to do this?

The best way I have found is to SAVE it then RESTORE it. My routine
MGH_OBJ_CLONE provides a convenient way to do this.

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_obj_clone.pro

It sounds kludgy but it works surprisingly well. I have used it widely &
successfully in my OG code for a year now. Just make sure the IDLgrModel in
question is not a child of any other IDLgrModel or IDLgrView at the time you
clone it.

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

Subject: Re: Copying an object
Posted by [davidf](#) on Thu, 02 Sep 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Nick Bower (nick.bower@ssec.wisc.edu) writes:

> you'll have to pardon my ignorance here, but I'm guessing that copying
> an existing object is, on the whole, performed less than instantiating a
> new one. So why not just inherit from the one or two objects you are
> likely to want to ever duplicate and add in a "duplicate" method to the
> new child class. wouldn't that make for safer code in the long run than
> trying to cover all bases with a universal object copier?

I think the "duplicate" method is the one we are
trying to figure out how to write. As it happens,
it's quite a bit harder than you might expect
it to be. :-)

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: Copying an object
Posted by [Nick Bower](#) on Thu, 02 Sep 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

you'll have to pardon my ignorance here, but I'm guessing that copying
an existing object is, on the whole, performed less than instantiating a
new one. So why not just inherit from the one or two objects you are
likely to want to ever duplicate and add in a "duplicate" method to the
new child class. wouldn't that make for safer code in the long run than
trying to cover all bases with a universal object copier? just curious
:)

nick
