
Subject: Re: removing objects from a container
Posted by [David Fanning](#) on Fri, 09 Jun 2006 18:19:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ben Tupper writes:

> David - please look out the window at your roses for a few minutes...

OK, I even gave the dog a good scratch while I was at it. I *do* feel better. :-)

> I can't quite figure the REMOVE method out. IDLgrROIgroup inherits from
> (eventually) IDL_CONTAINER so the REMOVE method is supposed to work. I
> don't see in the docs that IDLgrROIgroup overrides the method - but
> something is different.
>
>
> IDL> group = OBJ_NEW('IDLgrROIgroup')
> IDL> roi = OBJ_NEW('IDLgrROI')
> IDL> group->Add, roi
> IDL> group->REMOVE, /ALL
> % IDLGRROIgroup::REMOVE: Incorrect number of arguments.
> % Execution halted at: \$MAIN\$
> IDL> group->IDL_CONTAINER::REMOVE,/ALL
>
>
> Is this a bug or just a thorn?

Well, bad news. Here is my IDL session, with *no* error message!

```
IDL> group = OBJ_NEW('IDLgrROIgroup')
IDL> roi = OBJ_NEW('IDLgrROI')
IDL> group->Add, roi
IDL> group->REMOVE, /ALL
```

IDL 6.2 on Windows. (I'm not getting a new version of IDL until there is a good reason to. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: removing objects from a container
Posted by [Michael Galloy](#) on Fri, 09 Jun 2006 19:07:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Ben Tupper writes:

>

>> David - please look out the window at your roses for a few minutes...

>

> OK, I even gave the dog a good scratch while I was at it. I *do* feel

> better. :-)

>

>> I can't quite figure the REMOVE method out. IDLgrROIgroup inherits from

>> (eventually) IDL_CONTAINER so the REMOVE method is supposed to work. I

>> don't see in the docs that IDLgrROIgroup overrides the method - but

>> something is different.

>>

>>

>> IDL> group = OBJ_NEW('IDLgrROIgroup')

>> IDL> roi = OBJ_NEW('IDLgrROI')

>> IDL> group->Add, roi

>> IDL> group->REMOVE, /ALL

>> % IDLGRROIgroup::REMOVE: Incorrect number of arguments.

>> % Execution halted at: \$MAIN\$

>> IDL> group->IDL_CONTAINER::REMOVE,/ALL

>>

>>

>> Is this a bug or just a thorn?

>

> Well, bad news. Here is my IDL session, with *no* error message!

>

> IDL> group = OBJ_NEW('IDLgrROIgroup')

> IDL> roi = OBJ_NEW('IDLgrROI')

> IDL> group->Add, roi

> IDL> group->REMOVE, /ALL

>

> IDL 6.2 on Windows. (I'm not getting a new version of IDL until

> there is a good reason to. :-)

>

> Cheers,

>

> David

I get IDL 6.2 on Mac working and IDL 6.3 on Mac failing with the same error message.

I don't see anything in the docs about a change.

Mike

--

www.michaelgalloy.com

Subject: Re: removing objects from a container
Posted by [Karl Schultz](#) on Fri, 09 Jun 2006 23:38:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 09 Jun 2006 14:08:01 -0400, Ben Tupper wrote:

```
> Hello,
>
> David - please look out the window at your roses for a few minutes...
>
> I can't quite figure the REMOVE method out. IDLgrROIgroup inherits from
> (eventually) IDL_CONTAINER so the REMOVE method is supposed to work. I
> don't see in the docs that IDLgrROIgroup overrides the method - but
> something is different.
>
>
> IDL> group = OBJ_NEW('IDLgrROIgroup')
> IDL> roi = OBJ_NEW('IDLgrROI')
> IDL> group->Add, roi
> IDL> group->REMOVE, /ALL
> % IDLGRROIGROUP::REMOVE: Incorrect number of arguments.
> % Execution halted at: $MAIN$
> IDL> group->IDL_CONTAINER::REMOVE,/ALL
>
>
> Is this a bug or just a thorn?
>
```

It looks like a new bug was introduced here in 6.3 while trying to fix another problem.

The graphic objects have a PARENT property that is managed by the IDLgrContainer class as graphic objects are added or removed to/from containers like grModel. IDLgrROIgroup can't subclass from IDLgrContainer because IDLgrROIgroup inherits from IDLanROIgroup, which inherits from IDL_Container (IDLanROIgroup does not store displayable graphic objects) So, IDLgrROIgroup inherits from IDL_Container for its container functionality, and IDL_Container doesn't manage a PARENT property.

So, prior to 6.3, the PARENT property of an IDLgrROI was never getting updated as it was added or removed from an IDLgrROIgroup.

We fixed this bug in 6.3 by updating the PARENT property in the

already-overridden Add method and adding a new override for the Remove method that would clear the PARENT property when an IDLgrROI object was removed.

Unfortunately, the form IDLgrROIGroup->Remove, /ALL was not implemented correctly in the new Remove method. And it probably was not documented because it was supposed to work exactly like the superclass method.

I'll open a change request and fix this for next release.

The only workaround I can think of for now is:

```
group->Remove, group->Get(/ALL)
```

This would work correctly with or without the bug.

If you do

```
group->IDL_CONTAINER::REMOVE,/ALL
```

then the PARENT property of all the ROI's that were in the group will be left referring to the group object. If that doesn't bother you, then this would be another way to do it, but I don't think it is a good idea, as it leaves potential dangling object references.

Karl

Subject: Re: removing objects from a container
Posted by [Robbie](#) on Sat, 10 Jun 2006 23:38:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Indeed, this problem had caused me much grief and frustration in the past. I posted a bug report to ITT that a consequence of PARENT being null is that IDLgrROIGroup::GetCTM() will not return the affine transform with respect to the root scene. This meant that my code which deals with converting mouse clicks to atom coordinates has a special case for IDLgrROIGroup (and IDLgrColorbar). The same code must now also detect IDL 6.3 or above to account for the new behaviour.

So, yes! I'm definately getting different behaviour in IDL 6.3. The curious thing is that the fix was applied **after** IDL 6.3 Beta and isn't documented anywhere I could see. Am I playing with fire here?

Subject: Re: removing objects from a container
Posted by [Karl\[1\]](#) on Sun, 11 Jun 2006 02:12:24 GMT

Robbie wrote:

> Indeed, this problem had caused me much grief and frustration in the
> past. I posted a bug report to ITT that a consequence of PARENT being
> null is that IDLgrROIGroup::GetCTM() will not return the affine
> transform with respect to the root scene. This meant that my code which
> deals with converting mouse clicks to atom coordinates has a special
> case for IDLgrROIGroup (and IDLgrColorbar). The same code must now also
> detect IDL 6.3 or above to account for the new behaviour.

Yes, you were the one who reported the original bug, I think, and we fixed it for IDL 6.3.

This is a slight change in subject from the Remove method discussion because you're bringing up the issue of the PARENT property.

It is pretty hard to fix a bug while maintaining backwards-compatibility, because we're purposely fixing the previous bad behavior by changing it.

I don't know how your special case code works, but one option may be to simply leave it in place and it may work the same with the 6.3 bug fix in place. Other options include checking the PARENT property for NULL and taking different directions based on that, or implementing an IDL version check.

Bugs happen and we do our best to fix them. While we'd like IDL applications to be free of version-specific paths and workarounds, the reality is otherwise, but we strive to keep it to a minimum.

> So, yes! I'm definately getting different behaviour in IDL 6.3. The
> curious thing is that the fix was applied *after* IDL 6.3 Beta and
> isn't documented anywhere I could see. Am I playing with fire here?

I don't think so. We do fix bugs after Beta. Otherwise, there wouldn't be much point to a Beta program, right? :-) I am, however, not certain why the fix didn't make it into the release notes or why it didn't otherwise get communicated. All I can do is apologize for that and point out that you'd have to take the same steps to adapt your application to the fix anyway, even if there had been an entry in the release notes about the fix. So, except for the communications problem, you should be in better spot by getting the fix you requested.

Karl
(ittvis)

Subject: Re: removing objects from a container

Posted by [Robbie](#) on Sun, 11 Jun 2006 04:43:40 GMT

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

Going off topic again, Karl, you've incidently raised a very good point. Actually checking if the PARENT is null is probably better than version checking IDL. There are also other reasons why the PARENT might be null and I need to account for this. My code will be more stable if I avoid version checking.

- > So, except for the communications
- > problem, you should be in better spot by getting the fix you requested.

I am very grateful that the bug was fixed. I only checked the release notes after I had isolated the bug, so there was certainly no harm done. Then I realised that writing a bug report about this issue was really silly and you guys have better and more important things to worry about.

Cheers,

Robbie

Subject: Re: removing objects from a container
Posted by [btt](#) on Mon, 12 Jun 2006 14:36:56 GMT

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

Karl Schultz wrote:

- >
- > I'll open a change request and fix this for next release.
- >
- > The only workaround I can think of for now is:
- >
- > group->Remove, group->Get(/ALL)
- >
- > This would work correctly with or without the bug.
- >
- > If you do
- >
- > group->IDL_CONTAINER::REMOVE,/ALL
- >
- > then the PARENT property of all the ROI's that were in the group will be
- > left referring to the group object. If that doesn't bother you, then this
- > would be another way to do it, but I don't think it is a good idea, as it
- > leaves potential dangling object references.
- >

Thanks, Karl.

I shall heed your advice and modify my code accordingly.

Cheers,
Ben
