Subject: delvar
Posted by Dave Brennan on Thu, 04 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Hi

I was wondering, is there another way of deleting variable to clear memory other than using delvar. The probelm with delvar is that it resets the main program, thus stopping the remainder from being processed.

Thanks for your help

Dave Brennan

Subject: Re: delvar
Posted by Dominic Zarro on Mon, 08 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Actually, my last response was incorrect.
A better way, using pointers is as follows:

```
;-- to delete a variable var

;-- allocate a pointer

*p=ptr_new(/all)

;-- copy the variable into pointer

*p=temporary(var)    ;-- temporary will remove var

;-- free the pointer

ptr_free,p
```

You can easily package the above into a simple procedure that deletes any variable var.

Dominic

Subject: Re: delvar
Posted by thompson on Tue, 09 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Alex Schuster <alex@rosa.mpin-koeln.mpg.de> writes:

> Dominic Zarro wrote:

>> The following may not guarantee that a variable
>> is permanently deleted from IDL memory, but it
>> is useful for undefining the variable:
>>
>> IDL> a = temporary(a)

> Um, you didn't actually try this, did you? ;-)

I think that what Dominic was trying to say here was that if you wrote a simple routine along the lines

```
pro delete_variable, variable
dummy = temporary(variable)
end
```

you could then delete a variable by saying

```
delete_variable, a
```

I know that we've been doing that for a while now, using a similar procedure that Dominic wrote. I don't know how this would work with pointer variables, but it sounds like Dominic's on top of that, too.

Bill Thompson

Subject: Re: delvar
Posted by Martin Schultz on Tue, 09 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan wrote:
>
> [... snipped]

> Which again reminds me - it would be *really* nice if RSI would
> start putting in "Not before version X.XX" information in the
> online documentation for (new) features. It's tiresome enough
> trying to write version-sensitive code, and I just don't need
> the extra work required to check exactly which of the current
> features go back to this or that version...
>
> Stein Vidar
>

How about decreasing the speed with which they release new versions these days? This almost reminds me of Microsoft where you had just typed in the first sentence in your new version of word and the next update (or bug fix) came ... Personally, I would like to see some split between interim versions and major version updates. There should always be a latest official release which would remain stable for at least one year, and there could be additional "sneak preview" releases for those who want to try out new features at once (and get involved in beta or gamma testing). Then there would at least be some official standard as to what keywords are available, and everything else would be at own risk; even with possible changes in the next big release (so they would have a chance to correct inconsistencies without annoying too many people). And for most of us, the
IF (!VERSION ...) block could be held at a reasonable size.

I do agree that listing the version range in the online help would already help a lot! Even better, of course, if we could also find cross links to the correct places for older or newer versions of what we are/were used to.

Martin.

--
---------- -------
Dr. Martin Schultz
Department for Engineering&Applied Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
---------- -------

Subject: Re: delvar
Posted by Alex Schuster on Tue, 09 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Dominic Zarro wrote:

> The following may not guarantee that a variable
> is permanently deleted from IDL memory, but it
> is useful for undefining the variable:
>
> IDL> a = temporary(a)

Um, you didn't actually try this, did you? ;-)

Alex
--
Alex Schuster Wonko@weird.cologne.de PGP Key available
alex@pet.mpin-koeln.mpg.de

Subject: Re: delvar
Posted by steinhh on Tue, 09 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <36c05a04.0@208.206.112.5> "Dominic Zarro"
<zarro@tidalwave.net> writes:

> Actually, my last response was incorrect.
> A better way, using pointers is as follows:
>
> ;-- to delete a variable var
>
> ;-- allocate a pointer
>
> *p=ptr_new(/all)

Hi Dominic,

I guess you mean

```
p=ptr_new(/alloc)
```

(Seeing the "/all" in the ptr_new made me *really* confused,
I think the two extra letters are very clarifying :-)

[..]
>
> You can easily package the above into a simple
> procedure that deletes any variable var.

How about a one-line "syllogism":

```
ptr_free,ptr_new(temporary(var))
```

It's amazing what you have to do to delete a variable in IDL :-)

This discussion reminded me of a procedure I wrote some time in
1994, called like this:

ASSIGN_NOCOPY,destination,source

This was way before temporary() appeared on stage, and it even (still) has checks to function (resorts to copying) with IDL 3.0, which didn't even have the NO_COPY keyword in the WIDGET_CONTROL,GET_UVALUE=var mechanism....

Then came the handles, and then...

Which again reminds me - it would be *really* nice if RSI would start putting in "Not before version X.XX" information in the online documentation for (new) features. It's tiresome enough trying to write version-sensitive code, and I just don't need the extra work required to check exactly which of the current features go back to this or that version...

Stein Vidar
