Subject: Undefining a variable a pointer points to
Posted by Benjamin Hornberger on Fri, 14 May 2004 21:07:51 GMT

Hi all,

is there a way to undefine the variable a pointer points to? E.g., if I
initialize a pointer by

p = ptr_new(/allocate_heap)

p is a valid pointer, but *p is undefined. If I now do

*p = 10

is there a way to undefine it again afterwards, i.e. get back to the
status right after the call to ptr_new(/allocate_heap)?

Thanks for your help,
Benjamin

Subject: Re: undefining a variable
Posted by David Fanning on Tue, 12 Sep 2006 14:08:23 GMT

Ingo von Borstel writes:

> is there a way to undefine a variable in IDL?
> I cannot seem to find anything in the docs.

   http://www.dfanning.com/programs/undefine.pro

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: undefining a variable
Posted by Ingo von Borstel on Tue, 12 Sep 2006 14:20:47 GMT

Hi,

>
>> is there a way to undefine a variable in IDL?
>> I cannot seem to find anything in the docs.
>
>   http://www.dfanning.com/programs/undefine.pro

Thanks a lot! That was really quick and helpful

Ingo

--
Ingo von Borstel <newsgroups@planetmaker.de>
Public Key: http://www.planetmaker.de/ingo.asc

If you need an urgent reply, replace newsgroups by vgap.

Subject: Re: undefining a variable
Posted by cgguido on Tue, 12 Sep 2006 15:27:39 GMT

View Forum Message <> Reply to Message

undefine.pro is clever, but sometimes it is sufficient to set that var
to a scalar value. That frees up the memory, though it won't free up
the name :-) I haven't needed undefine so far so I thought I'd post my
2 cents.

Gianguido

Subject: Re: undefining a variable
Posted by David Fanning on Tue, 12 Sep 2006 15:35:24 GMT

View Forum Message <> Reply to Message

Gianguido Cianci writes:

> undefine.pro is clever, but sometimes it is sufficient to set that var
> to a scalar value. That frees up the memory, though it won't free up
> the name :-) I haven't needed undefine so far so I thought I'd post my
> 2 cents.

UNDEFINE wasn't written because it was "needed". It
was written because it was "elegant". :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: undefining a variable
Posted by cgguido on Wed, 13 Sep 2006 01:25:45 GMT

David Fanning wrote:

> UNDEFINE wasn't written because it was "needed". It
> was written because it was "elegant". :-)
>

Sorry, I mean't clever AND elegant. Of course. But the stuff you guys
come up with is always more elegant (and clever) than by dumb for-loop
approaches ;-) I am getting too used to it and forgot to mention it....

Gianguido

Subject: Re: undefining a variable
Posted by MarioIncandenza on Wed, 13 Sep 2006 16:21:59 GMT

Elegance certainly has its place. When you are dusting off a routine
you may have written eight years ago, and you say, 'Now why the devil
did I just set that entire huge structure = 0?', you'd better hope you
commented your code (boo!), or else used a nice intuitive command like
UNDEFINE() (yay!).

David Fanning wrote:
> Gianguido Cianci writes:
>
>> undefine.pro is clever, but sometimes it is sufficient to set that var
>> to a scalar value. That frees up the memory, though it won't free up
>> the name :-) I haven't needed undefine so far so I thought I'd post my
>> 2 cents.
>
> UNDEFINE wasn't written because it was "needed". It
> was written because it was "elegant". :-)
>

> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")