

---

Subject: Re: Physical constants in IDL with !CONST  
Posted by [ben.bighair](#) on Thu, 20 Dec 2012 15:03:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thursday, December 20, 2012 12:37:12 AM UTC-5, Chris Torrence wrote:

> On Wednesday, December 19, 2012 9:43:16 PM UTC-7, Craig Markwardt wrote:

>

>> On Wednesday, December 19, 2012 10:36:24 PM UTC-5, Jeff N. wrote:

>

>>

>

>>> Chris,

>

>>

>

>>>

>

>>

>

>>>

>

>>

>

>>>

>

>>

>

>>> If you're going to put pi in there, which we already have in !pi, you might also consider putting !dtor and !radeg in as well, otherwise people might be wondering why !pi was put into !const but not the other two.

>

>>

>

>>

>

>>

>

>> But if you do, then put the double precision values! !DTOR is worse than useless for me because it's only single precision.

>

>>

>

>>

>

>>

>

>> Craig

>

>  
>  
> Hi all,  
>  
>  
>  
> These are great suggestions.  
>  
>  
>  
> Paul, I just added in all of your constants except "c1" and "c2". There seems to be a discrepancy with the definition of c1. NIST gives it as  $2 \cdot \pi \cdot h \cdot c^2$ , which is a factor of "pi" bigger than your constant. I don't want anyone to get confused and just use the constant blindly, perhaps including an unexpected extra factor of pi. Which one is in "standard" use?  
>  
>  
>  
> Craig, I'll add in the AU, and the mass of the Sun & Earth. And of course "dctor" and "radeg" (double precision!). But don't worry, the older system variables won't go away.  
>  
>  
>  
> Russell, great suggestion on including more info, like the "units". I'd love to do that, but I don't want to make the structure too complicated. What about having a second system variable, say something like !CONST\_UNITS, which could contain additional information?

Hi,

Great idea1 Would it be a pain to call it something like !K instead of !CONST for all of the keyboard-bumblers out there (like me)? It might be nice to include the reference for each definition if they come from different standards.

Have you considered the possibility of using an object in the system variable?

```
mass = 12*!K->get("u", /value)
print, "mass = ", mass, !K->Get("u", /unit)
```

It might be nice if it were the possible to support two kinds of constants - built-in and user defined. The built-in would be read only, but the user defined could be added/modified as the user sees fit at start-up or whenever. At least that would provide a unified interface.

Cheers,  
Ben

---