
Subject: Re: Imaginary is narrow minded
Posted by [davidf](#) on Tue, 02 Sep 1997 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Now I'm really brainstorming. Here is another
program for Brian Jackel:

```
FUNCTION IF_IMAGINARY, variable
types = ['UNDEFINED', 'BYTE', 'INTEGER', 'LONG', 'FLOAT', $
        'DOUBLE', 'COMPLEX', 'STRING', 'STRUCTURE', $
        'DCOMPLEX', 'POINTER', 'OBJECT']
s = Size(variable)
thisType = types(s(s(0) + 1))
IF thisType EQ 'COMPLEX' THEN RETURN, Imaginary(variable) $
ELSE RETURN, 0.0
END
```

Then, you can do this:

```
imaginaryPart = IF_IMAGINARY(variable)
```

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com>

Subject: Re: Imaginary is narrow minded
Posted by [davidf](#) on Tue, 02 Sep 1997 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Brian Jackel writes:

```
> <Rant>
> I'm stuck with IDL 4.0, and was wondering if the snazzy new version
> still gives this stupid error
>
> % IMAGINARY: Operand must be of complex type:
>
> Alternatively, I'd love to know _why_ it can't just return a zero value.
> Sometimes the argument is going to be real, and it irritates me to add
> the extra call to make sure the argument is complex. IDL functions are
```

> usually pretty clever about dealing with variable input types, but not here.
> </Rant>

Oh, dear, it's one of those days, apparently. Here, Brian, something to cheer you up.

```
FUNCTION Type, variable
types = ['UNDEFINED', 'BYTE', 'INTEGER', 'LONG', 'FLOAT', $
        'DOUBLE', 'COMPLEX', 'STRING', 'STRUCTURE', $
        'DCOMPLEX', 'POINTER', 'OBJECT']
s = Size(variable)
RETURN, types(s(s(0) + 1))
END

IF Type(thatThing) EQ 'COMPLEX' THEN thisThing=Imaginary(thatThing)$
ELSE thisThing = 0
```

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com>

Subject: Re: Imaginary is narrow minded
Posted by [Christian Soeller](#) on Wed, 03 Sep 1997 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

davidf@dfanning.com (David Fanning) writes:

```
>
> IF Type(thatThing) EQ 'COMPLEX' THEN thisThing=Imaginary(thatThing)$
> ELSE thisThing = 0
>
```

Wouldn't you want to make it return an array of zeroes when called with an array?

Christian Soeller

Subject: Re: Imaginary is narrow minded
Posted by [meron](#) on Wed, 03 Sep 1997 07:00:00 GMT

In article <jackel.140.20AC6E36@danlon.physics.uwo.ca>, jackel@danlon.physics.uwo.ca (Brian Jackel) writes:

> Greetings

>

> <Rant>

> I'm stuck with IDL 4.0, and was wondering if the snazzy new version

> still gives this stupid error

>

> % IMAGINARY: Operand must be of complex type:

>

> Alternatively, I'd love to know `_why_` it can't just return a zero value.

I wrote my own function, `M_IMAGINARY`, which takes care of just this problem, and I use it exclusively instead of the IDL `IMAGINARY`.

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"
