Subject: Re: Grumbling about setting double precision
Posted by Kenneth P. Bowman on Wed, 21 Dec 2011 14:20:24 GMT
View Forum Message <> Reply to Message

In article < 31663559.520.1324400314865.JavaMail.geo-discussion-forums@yq dj19 >,
 wlandsman <wlandsman@gmail.com> wrote:

> I am converting a Python program to IDL which includes several hundred double
> precision data values. In Python the data values are written as follows:
>
> a = [2721.64363164731, -1615.73035059635]
>
> As described by David Fanning in
> http://www.idlcoyote.com/math_tips/double.html to make these double precision
> in IDL, it is not enough to convert the vector to double
>
> IDL> print,double(a),f='(2f18.10)'
> 2721.6435546875 -1615.7303466797
>
> and it is not enough to put a "d" at the end of one of the values
>
> IDL> a = [2721.64363164731d, -1615.73035059635]
> IDL> print,double(a),f='(2f18.10)'
> 2721.6436316473 -1615.7303466797
>
> Instead, I must spend an hour in my editor putting a "d" at the end of every
> single data value.
>
> But it would sure be nice to be able to somehow tell the IDL compiler to
> interpret numeric values as double precision. --Wayne

You could replace "," with "D," and replace "]" with "D]".

That should only take a minute or two with find and replace.

Ken

Subject: Re: Grumbling about setting double precision
Posted by David Fanning on Wed, 21 Dec 2011 14:28:09 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman writes:

> You could replace "," with "D," and replace "]" with "D]".

Yeah, there are probably very few commas in that code! ;-)

Cheers,

David

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

Subject: Re: Grumbling about setting double precision
Posted by greg.addr on Wed, 21 Dec 2011 14:35:44 GMT
View Forum Message <> Reply to Message

Would there be any downside *at all* to having the compiler always read values as doubles and convert down to floats when necessary?

Greg

Subject: Re: Grumbling about setting double precision
Posted by David Fanning on Wed, 21 Dec 2011 14:45:32 GMT
View Forum Message <> Reply to Message

greg.addr@googlemail.com writes:

> Would there be any downside *at all* to having the compiler always
> read values as doubles and convert down to floats when necessary?

Well, let's see. Twice as much memory allocated for
each number comes to mind. :-)

Cheers,

David

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

Subject: Re: Grumbling about setting double precision
Posted by greg.addr on Wed, 21 Dec 2011 15:04:02 GMT
View Forum Message <> Reply to Message

> Well, let's see. Twice as much memory allocated for
> each number comes to mind. :-)

...for each number written out by hand in your code. Even IDL on a 32bit Windows machine will swallow 100 million of those at a go without a hiccup. How many numbers do you type into your programs? :)

Subject: Re: Grumbling about setting double precision
Posted by David Fanning on Wed, 21 Dec 2011 15:21:38 GMT
View Forum Message <> Reply to Message

greg.addr@googlemail.com writes:

> ...for each number written out by hand in your code. Even IDL
> on a 32bit Windows machine will swallow 100 million of those at a go
> without a hiccup. How many numbers do you type into your programs? :)

So, you are saying, don't make all current floats doubles,
but make a double out of anything I define by hand: a=24.5.

But, what happens when I multiply my 10000x10000 floating
point array by my variable: array = array * a.

That will *surely* double the memory, or do we need new
rules for promoting values?

Cheers,

David

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

Subject: Re: Grumbling about setting double precision
Posted by Russell[1] on Wed, 21 Dec 2011 16:15:13 GMT
View Forum Message <> Reply to Message

No new rules, PLEASE!! Imagine what that would do for backwards-

compatibility!??!?

Russell

On Dec 21, 10:21 am, David Fanning <n...@dfanning.com> wrote:
> greg.a...@googlemail.com writes:
>> ...for each number written out by hand in your code. Even IDL
>> on a 32bit Windows machine will swallow 100 million of those at a go
>> without a hiccup. How many numbers do you type into your programs? :)
>
> So, you are saying, don't make all current floats doubles,
> but make a double out of anything I define by hand: a=24.5.
>
> But, what happens when I multiply my 10000x10000 floating
> point array by my variable: array = array * a.
>
> That will *surely* double the memory, or do we need new
> rules for promoting values?
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Grumbling about setting double precision
Posted by greg.addr on Wed, 21 Dec 2011 16:16:21 GMT
View Forum Message <> Reply to Message

Ok, a simple assignment would have to be converted to float (unless specified with the D) to retain the usual IDL behaviour:

a=24.5 {internally a=float(24.5d) }

but

b=[24.5d,9999.9] would then work as expected, since the assigned values (which are apparently copied into the dblarr after being read somewhere else) would already be doubles. In this case, the array type comes from the first element as usual (float unless specified otherwise), so

c=[24.5,9999.9] should likewise produce a satisfactory float array.

The other example from your page,

  array = Make_Array(3, Value=9999.9, /Double)

is a tricky one. Normally you would use

  array = Make_Array(3, Value=9999.9d)

to make a dblarr() with a specific value, which is fine (and /double for a zeroed array). Here, you're specifically converting the float to a double - I think there's no harm to leave that effect as it is.

Wouldn't that work?

cheers,
Greg

Subject: Re: Grumbling about setting double precision
Posted by David Fanning on Wed, 21 Dec 2011 16:22:53 GMT
View Forum Message <> Reply to Message

greg.addr@googlemail.com writes:

> Wouldn't that work?

I don't know. When things start getting really inelegant
in any program I've ever written, with a lot of special
rules and considerations, it's usually a sign I'm headed
down the wrong road. :-)

Cheers,

David

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

Subject: Re: Grumbling about setting double precision
Posted by Kenneth P. Bowman on Wed, 21 Dec 2011 17:26:20 GMT
View Forum Message <> Reply to Message

In article <MPG.295b9ae6d12361b798995a@news.giganews.com>,
 David Fanning <news@dfanning.com> wrote:

> Kenneth P. Bowman writes:
>
>> You could replace "," with "D," and replace "]" with "D]".
>
> Yeah, there are probably very few commas in that code! ;-)
>
> Cheers,
>
> David

David, you're always such a pessimist!

Ken

Subject: Re: Grumbling about setting double precision
Posted by David Fanning on Wed, 21 Dec 2011 18:13:21 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman writes:

> David, you're always such a pessimist!

I'm just saying, my primary motivation is to prevent
repetitive strain disorder. ;-)

Cheers,

David

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

Subject: Re: Grumbling about setting double precision
Posted by wlandsman on Thu, 22 Dec 2011 01:34:31 GMT
View Forum Message <> Reply to Message

On Wednesday, December 21, 2011 9:20:24 AM UTC-5, Kenneth P. Bowman wrote:

>
> You could replace "," with "D," and replace "]" with "D]".
>
> That should only take a minute or two with find and replace.
>

Thanks, but the problem in my code would be skipping the integer vectors (mixed in with the double precision vectors), which I don't want to be changed. I am sure there is a way to make my editor skip commas after integers, but I suspect that would take me more time to figure out than manually editing the file.

Both MATLAB and Python use double precision as their default data type. Because IDL has been more of an image processing language, it originally made sense for it to have floating point as a default. But with the increase in computer memory capabilities, I think it now makes sense to have a new DEFFLT64 parameter to COMPILE_OPT which, in analogy to DEFINT32, would make 64 bit floating point the default.

Subject: Re: Grumbling about setting double precision
Posted by greg.addr on Thu, 22 Dec 2011 11:48:46 GMT
View Forum Message <> Reply to Message

I think it would simply fix the bug transparently to the user, the only losers being those who rely on the truncation of their 8+ digit values. But perhaps longstanding workarounds are a form of elegance... :)

Subject: Re: Grumbling about setting double precision
Posted by Mark Piper on Tue, 03 Jan 2012 22:04:13 GMT
View Forum Message <> Reply to Message

On 12/21/2011 6:34 PM, wlandsman wrote:

> I think it now makes sense to have a new DEFFLT64 parameter to COMPILE_OPT which, in analogy to DEFINT32, would make 64 bit floating point the default.

I've seen several requests for a DEFFLT64 option in recent years. I'll
see if I can push for it to be included in IDL this year.

mp
