Subject: Re: 0=1 (Double precision/Long64)
Posted by penteado on Thu, 25 Feb 2010 18:08:34 GMT
View Forum Message <> Reply to Message

On Feb 25, 1:56 pm, wlandsman <wlands...@gmail.com> wrote:

> So b is equal to both a and a+1.   My guess is that the values are
> getting converted to double precision prior to the equality test.
> But the LONG64 variable has more precision than a double precision
> variable, and that precision is lost during the conversion.
>
> I'm not sure that there a good general solution for comparing between
> different data types.   But one needs to be careful when comparing
> LONG64 and double variables.

I think that converting integer types to floating point types is the usual way languages deal with operations that mix them, so this is not an IDL specific issue. Probably because it happens more often that the floating number is not an integer, and the integer is small enough to be represented exactly in the floating type.

Note that long64(b) is not equal to a, because double types are not precise to 1 part in 19. Double precision is only good to about 15 digits. For a number of that size in a double, only additions of the order of 1000 would change the value of b.

For that number to fit in a floating type you would need a quadruple precision type (128 bits), which gets to 34 digits. But IDL does not currently have such a type.