

---

Subject: Re: no backwards compatibility in IDL 5.6  
Posted by [notspecified](#) on Fri, 28 Feb 2003 15:04:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 28 Feb 2003 06:59:05 -0700, "R.G. Stockwell"  
<[sorry@noemail.now](mailto:sorry@noemail.now)> wrote:

> Craig Markwardt wrote:  
>> notspecified@dev.null (Matt Feinstein) writes:  
>>  
>>> I think this explains it adequately. In older versions, ATAN with a  
>>> complex argument returned a useful number --but the number it  
>>> returned didn't happen to be the arctangent of a complex argument!  
>>> Perhaps people should take a close look at Abramowitz and Stegun,  
>>> equation 4.4.39.  
>>  
>>  
>> Matt, let me say that I totally agree. The original behavior of ATAN  
>> was the correct implementation of the incorrect algorithm.  
>>  
>>  
>>> FWIW, if you write a program that uses incorrect, undocumented  
>>> behavior, you are asking for trouble. RSI can be blamed for not  
>>> providing a fast ARG or PHASE function, but this is a venial sin, at  
>>> worst. IMHO.  
>>  
>>  
>> Here is where I completely disagree. RSI covered up their original  
>> "oops" with another even bigger oops. There is no excuse to break an  
>> existing, working, interface in minor-release software. I realize  
>> that having ATAN do the correct "Abramowitz & Stegun" thing is more  
>> elegant, but I still argue that compatibility and maintainability  
>> always trumps elegance, at least in minor releases. RSI had their  
>> chance at elegance the first time around.  
> ...  
>>  
>> Craig  
>>  
>  
>  
> Be it foolish, but I don't agree with you. I think RSI did the right  
> thing by "finally" correcting the behaviour of the atan(), even if it  
> does break existing code. Because now all the user's bitch and moan :),  
> and fix their code, and never have a problem again with atan().  
> If they left it as it was, then for the next 100 years, new users would  
> be bitching and moaning about how the atan() doesn't work for  
> complex numbers.  
> As a case in point, check out recent discussions on the /center keyword

> in convol. :)

There is a classic essay entitled 'The Rise of Worse is Better' by Richard Gabriel on how different software design philosophies balance requirements for simplicity, correctness, consistency, and completeness. Gabriel isn't shy about arguing for his own point of view ( 'Incorrectness is not allowed'), and caricatures the views of people who disagree-- but the questions raised are fundamental, and people -will- disagree about these things.

My own view is that incorrectness -has- to be fixed. I understand that, since we all make mistakes at random, this means inevitable, random, and sometimes painful bumps in the road for simplicity and consistency. But, to me, the alternatives seem worse.

Matt Feinstein does not include his email address  
in the text of usenet postings.

-----

Harvard Law of Automotive Repair: Anything that goes away  
by itself will come back by itself.

---