Subject: Re: crazy loops
Posted by David Fanning on Wed, 13 Feb 2002 13:35:36 GMT
View Forum Message <> Reply to Message

M Carmen (mcgonzal@uv.es) writes:

> I have a question related with loops.....can someone teel me why
>
> if i do a loop that goes to a maximum of 0.6:
> for r=0.0, 0.6, 0.1 do begin & print, r & endfor
>
> I get:
> 0.000000
> 0.100000
> [.....]
> 0.500000
> 0.600000
>
> and if now i change the maximum value of the range to 0.7:
> for r=0.0, 0.7, 0.1 do begin & print, r & endfor
>
> I get:
> 0.000000
> 0.100000
> [.....]
> 0.500000
> 0.600000
>
> EXACTLY THE SAME!!!

Well, the short answer is "Because of the way computers represent floating point numbers." There have been numerous posts on this topic in the past. You might try searching the Google archives for "Set Precision", for example. The bottom line, however, is that it is not a good idea to use floating point values as counters, since you can't rely on their exact value. This has nothing to do with IDL. It is entirely related to how computers work.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: crazy loops
Posted by Craig Markwardt on Wed, 13 Feb 2002 14:04:10 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> writes:

> M Carmen (mcgonzal@uv.es) writes:
>
>> I have a question related with loops.....can someone teel me why
>>
>> if i do a loop that goes to a maximum of 0.6:
>> for r=0.0, 0.6, 0.1 do begin & print, r & endfor
>>
...
>>
>> EXACTLY THE SAME!!!
>
> Well, the short answer is "Because of the way computers
> represent floating point numbers." There have been numerous
> posts on this topic in the past. You might try searching
> the Google archives for "Set Precision", for example. The
> bottom line, however, is that it is not a good idea to use
> floating point values as counters, since you can't rely on
> their exact value. This has nothing to do with IDL. It is
> entirely related to how computers work.

And to prove David's point, try this little example:

```
IDL> print, 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 - 0.7
  5.96046e-08
```

The quantities 0.1 and 0.7 can't be represented exactly in a floating
point, so there will inevitably be some truncation errors. Going to
double precision doesn't always help (though it does here).

Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Re: crazy loops
Posted by Robert Stockwell on Wed, 13 Feb 2002 17:14:03 GMT
View Forum Message <> Reply to Message

M Carmen wrote:

> Hello,
>
> I have a question related with loops.....can someone teel me why
>
> if i do a loop that goes to a maximum of 0.6:
> for r=0.0, 0.6, 0.1 do begin & print, r & endfor
>
> I get:
> 0.000000
> 0.100000
> [.....]
> 0.500000
> 0.600000
>
> and if now i change the maximum value of the range to 0.7:
> for r=0.0, 0.7, 0.1 do begin & print, r & endfor
>
> I get:
> 0.000000
> 0.100000
> [.....]
> 0.500000
> 0.600000
>
> EXACTLY THE SAME!!!
>
> I have IDL Version 5.4
>
> Thanks :) !
> M Carmen

One should exercise great care when comparing the results of floating

point calculations:

(If these wrap around on your read, i apologize.)

run these lines:

IDL> limit = 0.7 & for r=0.0, limit, 0.1 do print, r,limit,r le limit,format='(f50.25,f50.25,f8.4)'
 0.0000000000000000000000000 0.69999998807907104492187500 1.0000

```
 0.10000000149011611938476560    0.69999998807907104492187500  1.0000
 0.20000000298023223876953120    0.69999998807907104492187500  1.0000
 0.30000001192092895507812500    0.69999998807907104492187500  1.0000
 0.40000000596046447753906250    0.69999998807907104492187500  1.0000
 0.50000000000000000000000000    0.69999998807907104492187500  1.0000
 0.60000002384185791015625000    0.69999998807907104492187500  1.0000
IDL> limit = 0.6 & for r=0.0, limit, 0.1 do print, r,limit,r le limit,format='(f50.25,f50.25,f8.4)'
0.00000000000000000000000000    0.60000002384185791015625000  1.0000
0.10000000149011611938476560    0.60000002384185791015625000  1.0000
0.20000000298023223876953120    0.60000002384185791015625000  1.0000
0.30000001192092895507812500    0.60000002384185791015625000  1.0000
0.40000000596046447753906250    0.60000002384185791015625000  1.0000
0.50000000000000000000000000    0.60000002384185791015625000  1.0000
0.60000002384185791015625000    0.60000002384185791015625000  1.0000
```

The for loop checks r le upper limit, and then executes if true.
Here, 0.6 is actually a little gt 0.600000000000, and 0.7 is just less than
0.70000000000000000000000000.
Hence the seemingly goofy results.

Cheers,
bob

PS to explain why you do not see the limit = 0.7 case, for that iteration
r = 0.70999997854232788085937500 and the limit is 0.69999998807907104492187500.

Subject: Re: crazy loops
Posted by Andre Kyme on Wed, 13 Feb 2002 22:11:46 GMT
View Forum Message <> Reply to Message

M Carmen wrote:

> Hello,
>
> I have a question related with loops.....can someone teel me why
>
> if i do a loop that goes to a maximum of 0.6:
> for r=0.0, 0.6, 0.1 do begin & print, r & endfor
>
> I get:
> 0.000000
> 0.100000
> [.....]
> 0.500000
> 0.600000

>
> and if now i change the maximum value of the range to 0.7:
> for r=0.0, 0.7, 0.1 do begin & print, r & endfor
>
> I get:
> 0.000000
> 0.100000
> [.....]
> 0.500000
> 0.600000
>
> EXACTLY THE SAME!!!
>
> I have IDL Version 5.4
>
> Thanks :) !
> M Carmen

Hi M Carmen,
I asked this very question not so long ago on this newsgroup (probably around October/November last year) -
you might want to check out that thread. Basically the answer was "never use a float as the loop variable."
Also, Jeff Hester probably has some helpful :-) comments to pass on. I know I was particularly encouraged
and assisted by his replies,

Andre
