In article <4KVt3.65216$jl.41071979@newscontent-01.sprint.ca>,
"Laurent Chardon" <chardons@NOT_THISsprint.ca> writes:
> Thanks David and Martin for your help. So if I understand properly, IDL uses
> fix() by default when converting doubles to integer. That's tricky. Does it
> make sense to do that?
> The solution would be to use c=fltarr(round(size)). Shouldn't IDL do that by
> default?

Definitively NO! Most often you need truncation rather than rounding, e.g.
when you want to create an array with a certain number of elements. Of
course, you are right that in 99.9999999% of these circumstances,
99.9999999 should be interpreted as 100 and not 99, but that's how bits
work together ;-C Unfortunately, we are used to thinking decadal, whereas
practically every machine thinks dual (or for the convenience of the
programmer hexadecimal). You can open any book on digital numerics and you will
see how such roundoff errors are introduced. Unfortunately, you will not always
see lower values, else it would be easy enough to write
    a = fltarr(long(expression) + 1)

Martin

> I'm wondering what's the use of using fix() instead of round() in this
> instance. Does anybod know?
>
> Laurent Chardon
> Trent University
>
>

--
 [[[[[[[[[[
[[Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 441787   [[
[[martin.schultz@dkrz.de   [[
 [[[[[[[[[[