Subject: Re: bizarre number transformation
Posted by James Kuyper on Thu, 25 Jul 2002 13:51:14 GMT
View Forum Message <> Reply to Message

merle wrote:
>
> Hello,
>
> I ran into a number transformation error yesterday that is still
> confusing me this morning. At first I thought I was doing something
> silly with String() & StrTrim(), but then I wrote a little program
> (see huh.pro) with no conversions that still contained the problem.
> FYI, I'm using IDL Version 5.5 Win32 (x86).
>
> The problem is that the number 443496.984 is being turned into the
> number 443496.969 from basic assignments using Float() or Double(),
> despite the fact that even floats should easily be able to handle a
> number this large (floats can handle "ï¿½10^38, with approximately six
> or seven decimal places of significance").
^^^^^^^^^^^^^^^

> Since I knew that I had successfully read in numbers much greater than
> 443496.984 in the past, I created temp.dat with just the number
> 443496.984 in it, and read this into a variable, x3. If x3 is cast as
> a float, it doesn't work, i.e. the number is 443496.969. But, if x3
> is cast as a double, then it contains the correct value. Why isn't a
> float sufficient (443496.984 << 10^38 and contains only 3 decimal
> places)? And, why doesn't x2=Double(443496.984) produce the correct
> result?

The number of decimal places after the decimal point is irrelevant. What matters is the number of significant digits in your number. 443496.984 has 9 significant digits. As you've found, 4-byte floating point numbers are typically good for only about 7 significant digits. If you want more precision than that, you'll have to use double precision variables.