Subject: Re: Commutativity of multiplication
Posted by David Fanning on Wed, 25 Oct 2006 20:54:20 GMT
View Forum Message <> Reply to Message

Sven Geier writes:

```
>  There's something odd afoot with ULong variables (in 6.3) that isn't clear
>  to me:
>
>  IDL> help,f,g,h
>  F               LONG      =          500
>  G               ULONG     =            1
>  H               FLOAT     =      1.00000
>  IDL> help,f*h,h*f
>  <Expression>    FLOAT     =      500.000
>  <Expression>    FLOAT     =      500.000
>  IDL> help,f*g,g*f
>  <Expression>    LONG      =          500
>  <Expression>    ULONG     =          500
>
>
>  Is this a bug? A feature? Well-known gotcha? Obscure property? Recently
>  added functionality? Vestigial peculiarity?
```

I'm pretty sure it's not a bug. I don't know about all
the rest of the possibilities. -)

I think the operative rules here are:

1. Maintain the type of data that maintains the most accuracy
2. If things are the same, process from left to right.

A ULONG and a LONG are both four bytes, and "accuracy" is
nebulous. A long could be negative. Is that "more accurate"
than treating the highest byte as a number rather than a
sign? I think you could argue both ways. So let's call it
even. Then, falling back on rule 2 explains the results.

It also explains this:

```
IDL> f = 500L
IDL> g = 1UL
IDL> help, g*(-f)
<Expression>    ULONG     =   4294966796
IDL> help, (-f)*g
<Expression>    LONG      =         -500
IDL> print, long(g*(-f))
        -500
```

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")