Subject: Re: Commutativity of multiplication
Posted by JD Smith on Thu, 26 Oct 2006 17:47:50 GMT
View Forum Message <> Reply to Message

On Thu, 26 Oct 2006 10:24:32 -0700, Braedley wrote:
>> IDL> print, 4294967295UL + 1UL
>> 0
>>
>> IDL> print,0b - 1b
>> 255
>>
>> JD
>
> IDL must make a choice as to which type to use, since the length of ranges
> of LONG and ULONG are exactly the same, but there is only 50% overlap.
> The result from the multiplication may be within the range of both, in
> which case everything is fine. However, if the result is negative, the
> result will be outside the range of ULONG. Likewise, if the two numbers
> are sufficiently large, the result will be outside the range of LONG. IDL
> doesn't know beforehand what the result will be, and therefore assigns the
> type of the leftmost variable.

It's actually much simpler than that, with no real "decision" involved.
The result of multiplying two integers is exactly the same whether they
are interpreted as signed or unsigned, including if you overflow the size
of the integer (this is the chief reason the two-complement system is so
prevalent). It is up to you (or, in this case, IDL) to decide how to
interpret the resulting number.

```
IDL> print,2L^30 * 2L
 -2147483648
IDL> print,2UL^30 * 2UL
  2147483648
IDL> print,long(2UL^30 * 2UL)
 -2147483648
```

The only case where this isn't so is up-conversion to non-equivalent
types (like floating point), but that's a separate matter, and it is
up to the environment (C compiler, IDL, etc.) as to how to handle
that.

JD