Subject: Re: Interesting WHERE function gotcha
Posted by JD Smith on Fri, 07 Feb 2003 16:04:22 GMT
View Forum Message <> Reply to Message

On Thu, 06 Feb 2003 19:35:31 -0700, Craig Markwardt wrote:

> David Fanning <david@dfanning.com> writes:
>>
>> How about this:
>>
>>   Help, veggie_letter, letter
>>   VEGGIE_LETTER   BYTE   = Array[4]
>>   LETTER   BYTE   = Array[1]
>>
>> "LETTER, a byte *array*!? You don't suppose..." Try this:
>>
>>   index = WHERE((Byte(array))[0,*] EQ (Byte('a'))[0], count) Print,
>>   count
>>   2
>>
>> Hummm. V-e-r-y interesting...
>>
>> Now I know how to fix the problem, but I don't know exactly what the
>> problem is. (Although this is not so different from most computer
>> problems, when you come to think of it.) Is the problem that the BYTE
>> function always makes a byte *array* when extracting string arguments?
>> Or is it that the WHERE function acts in a, uh, non-intuitive way when
>> there are two vectors in a boolean expression?
>
> Hi David!
>
> You are being bitten by the "feature" that I love so much. Namely that
> in IDL, when you do "X OPERATION Y", and X and Y are both arrays, then
> the expression is trimmed to the smaller of the two arrays.
>
> So it's not anything special regarding WHERE, or boolean expressions,
> but rather that VEGGIE_LETTER EQ LETTER evaluates to a 1-element array.

Yes, the tension between single element arrays and scalars is felt
deeply, even within the core IDL development group. It's a
regrettable legacy which must, unfortunately, be preserved for
compatibitlity. There are, however, minor changes and improvements:
e.g., with v5.6, KEYWORD_SET now considers a single element array TRUE
only if its single element is non-zero; i.e. it's treated just like a
scalar for this (and only this) function.

JD