Subject: Re: strings and memory usage
Posted by Jim Pendleton, ITT Vi on Thu, 23 Aug 2007 18:53:26 GMT
View Forum Message <> Reply to Message

Minor correction to my previous post... Since you're using a STRARR,
in this example the IDL_VARIABLE points to an IDL_ARRAY
descriptor which will in turn point to an array of IDL_STRING
descriptors, in turn pointing to the string values themselves.

Jim P.

"Jim Pendleton, ITT Visual Information Solutions" <jimp@no_spam.ittvis.com>
wrote in message news:13cp5thc73h1k49@corp.supernews.com...
> Take a look at the IDL External Development Guide's discussion of
> the IDL_Variable structure and the IDL_String structure. In
> summary, an IDL_Variable (a descriptor) points to an IDL_String, which
> is itself a descriptor and not just a null-terminated byte vector.
>
> If anyone (anyone?!!) recognizes the call "OTS$SCopy_DX_DX()", you'll
> understand the historical reason for this.
>
> Jim P.
>
> "Conor" <cmancone@gmail.com> wrote in message
> news:1187808869.444817.321600@r23g2000prd.googlegroups.com.. .
>> Does anyone know how IDL stores strings? I'm creating some very large
>> string arrays and running out of memory when I shouldn't. So, for the
>> following example I'm using the linux command 'top' to keep track of
>> memory usage on a per-process basis. In the beginning, IDL is using
>> 59 megabytes. Then, I create a string array with 5 million elements
>> like this:
>>
>> test = strarr(5000000) + 'asdf'
>>
>> Now I have a string array with 5,000,000 elements, each with 4
>> characters in it. According to top idl is now consuming 177
>> megabytes! That means that each string takes up an average of 23
>> bytes! To make matters worse, when I delete test (delvar,test) IDL
>> drops back down to 120 megabytes!
>>
>> What in the world is going on? Naievly, I would expect a string array
>> with strings 4 characters long to take up an absolute maximum of 8
>> bytes per element (4 bytes for the characters, 2 bytes for the length,
>> and maybe two bytes for pointers). Why is it taking up 23 bytes???
>> Am I just confused about something? Also, why doesn't the memory
>> usage drop back down to it's original value? I did notice one thing.
>> When I then created more large variables, the memory usage didn't
>> increase right away, so maybe IDL is clearing the memory but not

>> releasing it to the operating system. Still, I find these problems
>> very troubling. Is there something very wrong with the string arrays
>> in IDL, or am I just being silly?
>>
>
>