Subject: Re: strings and memory usage Posted by Conor on Wed, 22 Aug 2007 19:00:56 GMT

View Forum Message <> Reply to Message

On Aug 22, 2:54 pm, Conor <cmanc...@gmail.com> wrote:

- > 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?

For comparison, when I execute:

test2 = fltarr(1000000)

memory usage for IDL goes up by 4 megabytes, according to top - precisely what I would expect it to.

Subject: Re: strings and memory usage Posted by Jean H. on Wed, 22 Aug 2007 19:32:13 GMT View Forum Message <> Reply to Message

humm....

IDL> print, memory()

651318 545 213 659591 IDL > test = strarr(5000000) + 'asdf'IDL> print, memory() 125651409 5000549 216 185651386 IDL > test = 0IDL> print, memory() 5000220 125651409 651325 5000552 IDL > test = strarr(5000000)IDL> replicate\_inplace, test, 'asdf' IDL> print, memory() 60651418 5000557 5000224 60651418

it is surprising that depending on how you fill the array, the size differs...

Jean

Subject: Re: strings and memory usage Posted by Jim Pendleton, ITT Vi on Wed, 22 Aug 2007 20:08:16 GMT View Forum Message <> Reply to Message

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?

>

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?
>>
>
>
```