
Subject: Re: Binary file created with IDL strings

Posted by [peajai.all.the.way](#) on Wed, 05 Dec 2007 21:12:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Dec 5, 12:12 pm, Vince Hradil <hrad...@yahoo.com> wrote:

> On Dec 5, 11:00 am, peajai.all.the....@gmail.com wrote:

>

>

>

>

>

>> On Dec 5, 11:36 am, Vince Hradil <hrad...@yahoo.com> wrote:

>

>>> On Dec 5, 10:32 am, peajai.all.the....@gmail.com wrote:

>

>>>> I am trying to read in a IDL created binary file in C++. The file
>>>> contains floats, ints, doubles, and strings. I am able to read the
>>>> floats and ints (shorts in C++) with no problem, but once I get to the
>>>> section of the file containing strings, I cannot seem to read in the
>>>> correct number of bytes. I am not concerned with the contents on the
>>>> strings, but there is more data in floats after the strings that I
>>>> need to be able to read properly. If possible, I could seek forward
>>>> however many bytes just to get to the location of the data after the
>>>> strings. Does anyone know how many bytes each character in an IDL
>>>> string is? There are 22 strings, each 20 characters long. How many
>>>> bytes would this be?

>

>>> Each character is a byte.

>

>> Do I need to account for an extra byte at the end of each string for a
>> null terminator?

>

> I should have guessed you were going to ask that 8) I avoided it
> because I'm not 100% sure, but I think IDL does NOT write the null
> terminator like C. Pretty easy to test...- Hide quoted text -

>

> - Show quoted text -

Easy in theory but not in practice, this problem has been vexing me for 2 days. I tried skipping forward 440 bytes (20char*22strings) and 462 bytes ((20char*22strings)+22nulls) and still did not end up in the right place in the file to correctly read the floats follow. The first string seemed to start 8 characters later than I expected it to, and I cannot tell if I need to add 8 extra characters for each string or if something is just weird at the beginning of strings. Mind you, I have IDL code that reads this same file perfectly so I know the file isn't bad and I am following the same order that the IDL code follows to read the data in.

The IDL code actually just calls `readu` once and passes it all of the arrays it is expecting to populate and this includes one string array of 22 elements with 20 chars each. If this file was created in IDL and initialized with an array of strings, should there be extra bytes for the array? Or for each string? How would I be able to tell? Would an array of strings take up more bytes than just simply a set of contiguous strings?
