
Subject: Re: reading a ninary file

Posted by peter.albert@gmx.de on Mon, 09 Jan 2006 10:12:50 GMT

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

Hi,

you have to know *exactly* what it is the file before you start reading it, i.e. the number and type of each and every variable within the file. However, this seems to be the case as it looks as if you get the long and fix variables well out of the file. As for the strings, I'd suggest reading them as byte arrays and then converting them to string. Mind, however, that you need to know the exact number of characters within each string variable. Thus, it is not enough to know "there is a string followed by a long number", but you need to know "there is a string with 5 characters, followed by ..."

The IDL lines would then be

```
b = bytarr(5)
readu, lun, b
str_var = string(b)
```

Cheers,

Peter

Subject: Re: reading a ninary file

Posted by peter.albert@gmx.de on Mon, 09 Jan 2006 10:22:23 GMT

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

Opps, sorry, I obviously didn't read your posting carefully. You said the strings are in fact just one byte long, and you don't get them right. In that case I'd suggest using a hexeditor first for looking into the file. Under Linux / Unix a call like

```
hexdump -c your_file
```

should print the file's content as hexadecimal and character values, so you should be able to find the position and value of the string values and compare that to what you get with IDL.

Cheers,

Peter

Subject: Re: reading a binary file
Posted by [Klaus Scipal](#) on Mon, 09 Jan 2006 13:02:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi

I am not a specialist in character encoding but I guess you have to find out how it was written. Which encoding was used to convert the characters to a byte?

Characters are normally encoded in ASCII format which knows 128 characters. As computers store data in bytes (i.e. 8 bits) there is room to store another set of 128 characters, i.e. an extended character set. In practice, there are a number of different extended character sets for example for math symbols or extension characters for non-English languages. And to make it even more difficult there are also other encoding systems than ASCII, for example UNICODE.

I don't know if this is your problem but unless you don't know how the data was encoded it will be difficult to "decode" it. I also don't know if the different character sets are compatible and how IDL converts bytes to characters. That's something you have to find out yourself. For more details on character binary encoding check out <http://www.cs.tut.fi/~jkorpela/chars.html#examples>

Of course a trial and error method would be to read out the byte and go into the standard conversion tables and look what makes most sense.

Klaus

<claire.maraldi@gmail.com> wrote in message
news:1136798479.995043.160620@g44g2000cwa.googlegroups.com.. .

> Hello,
>
> I have to read a binary file containing long, fix and string variable
> type. I know that the string variable type are coded only on one byte
> (this have been confirmed by someone in the laboratory, and even if I
> try more bytes there is an "encountered before end of file" error...)
> So I have tried to convert only one byte, and the results are amazing
> characters like "_-", "_ "....
> I know that is not a problem of discrepancy when the binary file is
> read because long and fix variable type are well converted (wether they
> are placed before or after string).
>
> Could explain me what exactly happen please ?
> Thank you
>

Subject: Re: reading a binary file
Posted by [R.Bauer](#) on Sun, 15 Jan 2006 08:32:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all

I got more and more unicode files.

the king is dead long lives the new king

idl has no unicode support at the moment. We should add a feature request about.

cheers
Reimar

Klaus Scipal wrote:

> Hi
>
> I am not a specialist in character encoding but I guess you have to find
> out how it was written. Which encoding was used to convert the characters
> to a byte?
>
> Characters are normally encoded in ASCII format which knows 128
> characters. As computers store data in bytes (i.e. 8 bits) there is room
> to store another set of 128 characters, i.e. an extended character set. In
> practice, there are a number of different extended character sets for
> example for math symbols or extension characters for non-English
> languages. And to make it even more difficult there are also other
> encoding systems than ASCII, for example UNICODE.
>
> I don't know if this is your problem but unless you don't know how the
> data was encoded it will be difficult to "decode" it. I also don't know if
> the
> different character sets are compatible and how IDL converts bytes to
> characters. That's something you have to find out yourself. For more
> details on character binary encoding check out
> <http://www.cs.tut.fi/~jkorpela/chars.html#examples>
>
> Of course a trial and error method would be to read out the byte and go
> into the standard conversion tables and look what makes most sense.
>
> Klaus
>
>
>

> <claire.maraldi@gmail.com> wrote in message
> news:1136798479.995043.160620@g44g2000cwa.googlegroups.com...
>> Hello,
>>
>> I have to read a binary file containing long, fix and string variable
>> type. I know that the string variable type are coded only on one byte
>> (this have been confirmed by someone in the laboratory, and even if I
>> try more bytes there is an "encountoured before end of file" error...).
>> So I have tried to convert only one byte, and the results are amazing
>> characters like "_-", "_ "....
>> I know that is not a problem of discrepency when the binary file is
>> read because long and fix variable type are well converted (wether they
>> are placed before or after string).
>>
>> Could explain me what exactly happen please ?
>> Thank you
>>
