
Subject: Reading binary files made by Python
Posted by [Liberum](#) on Mon, 10 Jul 2006 09:45:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have created som binary files using python and tried to read them with IDL. I used the binary_template as well but IDL completely misinterprets the data and returns wrong values and values that were never written in the file. Now Matlab is able to read these files without problem. My question is why is IDL failing to do this? I know that the binaries are Float32 and the dimensions are correct. I am doing something wrong?

Sincerely,
Sheldon

Subject: Re: reading binary files
Posted by [Wox](#) on Tue, 19 Jun 2007 13:00:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 19 Jun 2007 00:49:17 -0700, Vince Oliver
<vince@complex.elte.hu> wrote:

```
> Firest try was using Assoc function. It was somehow natural choice
> since the number of spectra is large. So:
>
> data = 'c:\users\spectra10.bin'
> spectra = Assoc(lun, Fltarr(2,7323))
> aSpectra = spectra[0]
> print, aSpectra[0,0],aSpectra[1,0]
> print, aSpectra[0,7322],aSpectra[1,7322]
> print
> aSpectra = spectra[1]
> print, aSpectra[0,0],aSpectra[1,0]
> print, aSpectra[0,7322],aSpectra[1,7322]
>
> The outout is
>
> 8.20937e-041    91.0000
> 1.07971e-021 3.60000e+008
>
> 2.11875e-022 8.20937e-041
> 2.57653e-020 2.40000e+008
>
>
> It was clear that I should put an offset keyword in Assoc like spectra
```

```

> = Assoc(lun, Fltarr(2,7323),4). The output in this case is:
>
> 91.0000 2.04773e-023
> 3.60000e+008 2.11875e-022
>
> 8.20937e-041 8.20937e-041
> 2.40000e+008 5.08605e-021
>
>
> So the 1st spectrum is properly read out but the second not. It seems
> that there is an offset between each spectra. How to read out spectra
> properly in this case?
>
> #####

```

From IDL's help:

"Unformatted data files generated by FORTRAN programs under UNIX contain an extra long word before and after each logical record in the file. ASSOC does not interpret these extra bytes but considers them to be part of the data. This is true even if the F77_UNFORMATTED keyword is specified..."

I have no experience with this, but what happens if you use (e.g. for 3rd spectrum):

```

specnr=2
nlen=7323
n=2*nlen+1
specoff=n*specnr*4
spectra = Assoc(lun, Fltarr(n),specoff)

```

Subject: Re: reading binary files
 Posted by [Wox](#) on Tue, 19 Jun 2007 13:06:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

```

> I have no experience with this, but what happens if you use (e.g. for
> 3rd spectrum):
>
> specnr=2
> nlen=7323
> n=2*nlen+1
> specoff=n*specnr*4
> spectra = Assoc(lun, Fltarr(n),specoff)

```

Sorry, this:

```
specnr=2
nlen=7323
n=2*nlen
specoff=(n+1)*specnr*4
spectra = Assoc(lun, Fltarr(n),specoff)
```

I assume here that the data is: DWORD|fltarr(n)|DWORD|fltarr(n)...

Subject: Re: reading binary files
Posted by [Vince Oliver](#) on Tue, 19 Jun 2007 14:16:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

we are maybe on the good way. I wrote this

OpenR, lun, data, /Get_Lun, /F77_UNFORMATTED

```
specnr=2
nlen=7323
n=2*nlen
specoff=(n+1)*specnr*4l ; this is letter L not number 1
spectra = Assoc(lun, Fltarr(n),specoff)
```

```
aSpectra = spectra[0]
```

```
for i = 0l, 10, 2 do begin
    print, aSpectra[i], aSpectra[i+1]
endfor
```

the output is:

```
1.00407e-021 8.20937e-041
8.20937e-041    91.0000
2.39148e-009    94.0000
3.81471e-009    96.0000
5.10172e-009    98.0000
2.44573e-008   100.000
```

Should be like

```
91. 2.39147746E-09
94. 3.81470944E-09
96. 5.10172171E-09
....
```

> From where these 3 numbers are on the begining? I can put

```
for i = 3l, 10, 2 do begin
    print, aSpectra[i], aSpectra[i+1]
endfor
```

but the point is that I miss 3 last numbers at the end because of these numbers.

On Jun 19, 3:06 pm, Wox <nom...@hotmail.com> wrote:

>> I have no experience with this, but what happens if you use (e.g. for
>> 3rd spectrum):

```
>
>> specnr=2
>> nlen=7323
>> n=2*nlen+1
>> specoff=n*specnr*4
>> spectra = Assoc(lun, Fltarr(n),specoff)
```

> Sorry, this:

```
>
> specnr=2
> nlen=7323
> n=2*nlen
> specoff=(n+1)*specnr*4
> spectra = Assoc(lun, Fltarr(n),specoff)
>
> I assume here that the data is: DWORD|fltarr(n)|DWORD|fltarr(n)...
```

Subject: Re: reading binary files

Posted by [Paul Van Delst\[1\]](#) on Tue, 19 Jun 2007 14:47:29 GMT

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

Vince Oliver wrote:

```
> we are maybe on the good way. I wrote this
>
>
> OpenR, lun, data, /Get_Lun, /F77_UNFORMATTED
>
> specnr=2
> nlen=7323
> n=2*nlen
```

```
> specoff=(n+1)*specnr*4l ; this is letter L not number 1
> spectra = Assoc(lun, Fltarr(n),specoff)
```

what about something like,

```
spectra = ASSOC(lun,{RecBegin:0L, Data:FLTARR(2,n), RecEnd:0L})
spectrum = spectra[0]
```

?

Do the spectrum.data contain the correct info? Additionally, are the values of spectrum.recbegin and spectrum.recend the same as 2*n*4?

Remember that *each* output record (for Fortran sequential access, unformatted) has a marker both at the beginning *and* the end specifying the record length (hopefully in bytes, not "words").

ASSOC is most often used for direct access, unformatted; but there's no reason you can't use it here also.

>> I assume here that the data is: DWORD|fltarr(n)|DWORD|fltarr(n)...

Nope, it should be

RECLen1|fltarr(2,n)|RECLen1|RECLen2|fltarr(2,n)|RECLen2....

One of the reasons for the terminating record length marker (what I called "RecEnd" in the structure above) is so that the Fortran BACKSPACE command works for sequential access files.

cheers,

paulv

--

Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC

Eddy Merckx

Subject: Re: reading binary files
Posted by [Vince Oliver](#) on Tue, 19 Jun 2007 15:06:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

paul, when I do

```
spectra = ASSOC(lun,{RecBegin:0L, Data:FLTARR(2,n), RecEnd:0L})
spectrum = spectra[0]
print, spectrum
```

I indeed have:

```
SPECTRUM    STRUCT  = -> <Anonymous> Array[1]
{   58584   91.0000 2.04773e-023
    94.0000 5.73142e-023
    96.0000 1.12218e-022
    98.0000 2.17056e-022
   100.000 4.00652e-022
...}
```

How to get these elements?

On Jun 19, 4:47 pm, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:

> Vince Oliver wrote:

>> we are maybe on the good way. I wrote this

>

>> OpenR, lun, data, /Get_Lun, /F77_UNFORMATTED

>

>> specnr=2

>> nlen=7323

>> n=2*nlen

>> specoff=(n+1)*specnr*4l ; this is letter L not number 1

>> spectra = Assoc(lun, Fltarr(n),specoff)

>

> what about something like,

>

> spectra = ASSOC(lun,{RecBegin:0L, Data:FLTARR(2,n), RecEnd:0L})

> spectrum = spectra[0]

>

> ?

>

> Do the spectrum.data contain the correct info? Additionally, are the values of

> spectrum.recbegin and spectrum.recend the same as 2*n*4?

>

> Remember that *each* output record (for Fortran sequential access, unformatted) has a

> marker both at the beginning *and* the end specifying the record length (hopefully in

> bytes, not "words").

>

> ASSOC is most often used for direct access, unformatted; but there's no reason you can't

> use it here also.
>
>>> I assume here that the data is: DWORD|fltarr(n)|DWORD|fltarr(n)...
>
> Nope, it should be
>
> RECLLEN1|fltarr(2,n)|RECLLEN1|RECLLEN2|fltarr(2,n)|RECLLEN2....
>
> One of the reasons for the terminating record length marker (what I called "RecEnd" in the
> structure above) is so that the Fortran BACKSPACE command works for sequential access
files.
>
> cheers,
>
> paulv
>
> --
> Paul van Delst Ride lots.
> CIMSS @ NOAA/NCEP/EMC Eddy Merckx

Subject: Re: reading binary files
Posted by [Paul Van Delst\[1\]](#) on Tue, 19 Jun 2007 15:28:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Vince Oliver wrote:
> paul, when I do
>
> spectra = ASSOC(lun,{RecBegin:0L, Data:FLTARR(2,n), RecEnd:0L})
> spectrum = spectra[0]
> print, spectrum
>
> I ended have:
>
> SPECTRUM STRUCT = -> <Anonymous> Array[1]
> { 58584 91.0000 2.04773e-023
> 94.0000 5.73142e-023
> 96.0000 1.12218e-022
> 98.0000 2.17056e-022
> 100.000 4.00652e-022
> ...}
>
>
> How to get these elements?

The structure components? Use the "." operator to resolve them. Try,

IDL> help, spectrum.data

For example:

```
IDL> spectrum={rb:-999L, data:findgen(5)+10.0, re:-999L}
IDL> print, spectrum
{   -999   10.0000   11.0000   12.0000   13.0000   14.0000
  -999}
IDL> help, spectrum
SPECTRUM      STRUCT  = -> <Anonymous> Array[1]
IDL> help, spectrum.data
<Expression>  FLOAT   = Array[5]
IDL> print, spectrum.data
   10.0000   11.0000   12.0000   13.0000   14.0000
IDL> print, spectrum.rb, spectrum.re
   -999     -999
```

Check your IDL help for much more info on IDL structures.

cheers,

paulv

--

Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC

Eddy Merckx

Subject: Re: reading binary files
Posted by [Vince Oliver](#) on Tue, 19 Jun 2007 16:10:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

ok ... works seems to me. thanks for help. I copy paste the program
for the case somebody has similar problem

```
data = 'c:\users\\spectra10.bin'
```

```
OpenR, lun, data, /Get_Lun, /F77_UNFORMATTED
```

```
n=7323
nmod = 10
```

```
spectra = ASSOC(lun,{RecBegin:0L, data:FLTARR(2,n), RecEnd:0L})
```

```
for i = 0, nmod-1 do begin
  openw, out, strcompress('test'+string(i)+'.dat'), /get_lun
  spectrum = spectra[i]
```

```

for j = 0, (size(spectrum.data))[2]-1 do begin
  printf, out, spectrum.data[0,j], spectrum.data[1,j]
endfor
close, out
free_lun, out
endfor

```

```

Close, lun
Free_Lun, lun

```

On Jun 19, 5:28 pm, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:

```

> Vince Oliver wrote:
>> paul, when I do
>
>> spectra = ASSOC(lun,{RecBegin:0L, Data:FLTARR(2,n), RecEnd:0L})
>> spectrum = spectra[0]
>> print, spectrum
>
>> I ended have:
>
>> SPECTRUM      STRUCT  = -> <Anonymous> Array[1]
>> {   58584   91.0000 2.04773e-023
>>   94.0000 5.73142e-023
>>   96.0000 1.12218e-022
>>   98.0000 2.17056e-022
>>  100.000 4.00652e-022
>> ...}
>
>> How to get these elements?
>
> The structure components? Use the "." operator to resolve them. Try,
>
> IDL> help, spectrum.data
>
> For example:
>
> IDL> spectrum={rb:-999L, data:findgen(5)+10.0, re:-999L}
> IDL> print, spectrum
> {   -999   10.0000   11.0000   12.0000   13.0000   14.0000
>   -999}
> IDL> help, spectrum

```

```
> SPECTRUM      STRUCT  = -> <Anonymous> Array[1]
> IDL> help, spectrum.data
> <Expression>  FLOAT   = Array[5]
> IDL> print, spectrum.data
>    10.0000   11.0000   12.0000   13.0000   14.0000
> IDL> print, spectrum.rb, spectrum.re
>    -999     -999
>
> Check your IDL help for much more info on IDL structures.
>
> cheers,
>
> paulv
>
> --
> Paul van Delst      Ride lots.
> CIMSS @ NOAA/NCEP/EMC      Eddy Merckx- Hide quoted text -
>
> - Show quoted text -
```
