
Subject: Re: How to read two-bytes variables from a file saved in Mac?

Posted by [btt](#) on Tue, 24 Apr 2001 13:25:53 GMT

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

Hello,

We use IPLAB to collect images, but we are migrating toward IDL for analysis. I have written three routines (QUERY_IPLAB, READ_IPLAB and WRITE_IPLAB, see attached) that work using IDL for MAC. I have a vague memory of these working on UNIX, also. I am quite interested in making these routines platform independent and would like to know if Craig's suggestion solved the problem for you.

Ben

Craig Markwardt wrote:

```
>
> "Xiong Hu" <xhu@conrad.ece.uiuc.edu> writes:
>
>> Hi! Does somebody know how to read the two-bytes variables from a file saved
>> in Macintosh?
>>
>> I am trying reading the image data in PC platform. The image data saved in
>> IPlab in Mac. And I found out the difference of the way the data be saved in
>> between PC and Mac. For example, a two- bytes integer will be saved like:
>> 0[low byte]1[high byte] in PC platform. But in Mac, it will be saved like:
>> 1[high byte]0[low byte].
>>
>> So in PC when I try to read a file saved in Mac, I need to read byte by byte
>> and do transform in the right way to get the right value of each pixel. But
>> I think it is not a good way. I wonder if there are any "keywords" or
>> "command" can help me solve this problem directly and efficiently.
>
> You are dealing with an issue called "endianness," which refers to how
> a particular CPU stores multi-byte values.
>
> You should investigate the SWAP_* keywords to OPEN, one of them should
> be able to help you. If you need to selectively swap data (ie some
> values are byte-swapped and others are not), then you can investigate
> the BYTEORDER function, which is used after reading the data.
>
> Craig
>
> --
> -----
> Craig B. Markwardt, Ph.D.      EMAIL:  craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
> -----
```

```

--
Ben Tupper
Bigelow Laboratory for Ocean Sciences
180 McKown Point Rd.
W. Boothbay Harbor, ME 04575
btupper@bigelow.org
;+
; NAME:
;   QUERY_IPLAB
;
; PURPOSE:
;   The purpose of this function is to check the format of a IPLAB image file before trying to
;   read it. The header of the file can optionally be returned. If the file is a IPLab image format
;   3.1 or greater and in data format then a 1 is returned, otherwise a 0 is returned.
;
; CALLING SEQUENCE:
;   result = QUERY_IPLAB([File])
;
; ARGUMENTS:
;   File   A scalar string containing the name of the file to query. If not provided, a
;          dialog_pickfile prompt permits the user to select one automatically.
;
; KEYWORDS:
;   HDR   Set this keyword equal to a named variable to return the header structure
;         which contains information about the file.
;   GROUP Set this keyword equal to the group leader for the widget dialog. This
;         value is only necessary if the routine is used in a widget application and the
;         file name is not specified.
;
; EXAMPLE:
;   To interactively locate the file to query and return the file header...
;   GoodFlag = QUERY_IPLAB(HDR = HDR)
;
; COMMENTS:
;   Here's an example of the HDR structure
;   IDL> help, hdr,/str
; ** Structure <d183d00>, 13 tags, length=2128, refs=1:
;   VERSION      STRING  '3.1a'
;   IPLABFORMAT  BYTE    0
;   DATATYPE     BYTE    0
;   WIDTH        LONG    640
;   HEIGHT       LONG    480
;   RESERVED1    BYTE    0
;   OVERLAYINFILE BYTE    0
;   RESERVED2    LONG    16842753
;   NFRAMES      INT     5
;   RESERVED3    BYTE    Array[25]

```

```

; NREGMARKS    BYTE    0
; RESERVED4    BYTE    Array[24]
; CLUT         BYTE    Array[2048]
;
; REFERENCE:
; For more information on the IPLAB format see...
; 'IPLAB Spectrum: Scientific Imaging Software for the Macintosh, User's Guide'
; Signal Analytics Corporation, 1996.
;
; MODIFICATION HISTORY:
; Written by Ben Tupper, 27JULY2000
; Bigelow Laboratory For Ocean Science
; tupper@bigelow.org
;
; 2AUG2000 Improved file type checking by examining the Version input by byte elements
;-

```

```

FUNCTION query_iplab, File, HDR = HDR, Group = Group

```

```

On_Error, 2

```

```

; if there is an IO error, make sure to return, 0
On_IOError, Trouble

```

```

; if the file is undefined, then prompt with a dialog
If N_elements(File) EQ 0 Then Begin
  File = Dialog_PickFile(Group = Group)
  If File[0] EQ "" Then Return, 0
EndIf

```

```

; open the file
Openr, u, File[0], /get_lun, /Binary

```

```

; check if this is the proper format
Version = ' '
Format = 0B

```

```

ReadU, U, Version

```

```

VersionTest = Byte(Version)
If N_elements(VersionTest) LT 3 Then Begin
  If (FSTAT(U)).Open Then Free_LUN, U
  Return, 0
EndIf

```

```

If Float(VersionTest[0]) LT 3 Then Begin
  If (FSTAT(U)).Open Then Free_LUN, U
  Return, 0

```

EndIf

```
ReadU,U, Format
If Format[0] NE 0B Then Begin
  If (FSTAT(U)).Open Then Free_LUN, U
  Return, 0
EndIf
```

```
  ;back up the pointer
Point_LUN,U, 0
```

```
  ;prep the header structure
Hdr = { Version:' ', $
  IPLabFormat:0B,$
  DataType:0B,$
  Width:1L,$
  Height:1L,$
  Reserved1:0B,$
  OverlayInFile:0B,$
  Reserved2: BytArr(4),$
  nFrames :1,$
  Reserved3: Bytarr(25),$
  nRegMarks:1B,$
  Reserved4: Bytarr(24),$
  CLUT:Bytarr(2048)}
```

```
;read the header
ReadU,u,Hdr
FREE_LUN, U
```

Return, 1

```
TROUBLE:
OK = Error_Message(/TraceBack)
If (FSTAT(U)).Open Then Free_LUN, U
Return, 0
```

END

```
;+
; NAME:
; Read_IPLAB
;
; PURPOSE:
; This function returns an IPLAB format image. Overlays and
```

```

; registration marks cannot be read. Overlays are in PICT2
; vector format. See Reference below.
;
; CALLING SEQUENCE:
; Result = READ_IPLAB(file)
;
; ARGUMENTS:
; File Set this argument equal to the full file/path for the
; IPLAB format image.
; If not provided, the user is prompted by Dialog_PickFile
; If the user cancels, then -1 is returned.
;
; KEYWORDS:
; HDR Set this keyword equal to a named variable to return
; the file header structure which describes the file contents.
; Below is a sample of a header structure showing the field values.
; Note that even though there is an OVERLAYINFILE it is not read.
;
; VERSION      STRING  '3.1a'
; IPLABFORMAT  BYTE    0
; DATATYPE     BYTE    0
; WIDTH        LONG    640
; HEIGHT       LONG    480
; RESERVED1    BYTE    0
; OVERLAYINFILE BYTE    1
; RESERVED2    LONG    16842753
; NFRAMES      INT     5
; RESERVED3    BYTE    Array[25]
; NREGMARKS    BYTE    0
; RESERVED4    BYTE    Array[24]
; CLUT         BYTE    Array[2048]
;
; ALPHA Set this keyword to return a 32 bit image with the
; first image plane [0] equal to the alpha channel.
;
; REFERENCE:
; IPLAB Spectrum, Scientific Imaging Software for the MacIntosh.
; Signal Analytics Corporation, 1996, User's Guide, v 3.1 , rev. 5.8.
;
; REQUIREMENTS:
; QUERY_IPLAB is called.
;
; MODIFICATION HISTORY:
; Written by Ben Tupper, Fall, 1999
; Bigelow Lab for Ocean Sciences
; tupper@seadas.bigelow.org
; pemaquidriver@tidewater.net
;
;

```

```
; Added Documentation and cleaned up. 29JUNE2000 BT
; Changed BytesPerPixel to 1L for a Integer Image
; Added call to QUERY_IPLAB. 27JULY2000 BT
;-
```

```
Function Read_IPLab, File , $
Alpha = Alpha, Hdr = Hdr, Group = group
```

```
;make sure that the type conforms to IPLAB image data, get the header
Flag = QUERY_IPLAB(File, HDR = HDR, Group = Group)
```

```
If Flag LT 1 Then Return, -1
```

```
    ;prep the image array
Case Hdr.DataType of
0: Begin
    BytesPerPixel = 1L ;byte
    Type = 1
END
```

```
1: BEGIN
BytesPerPixel = 1L ;int
Type = 2
END
```

```
2: BEGIN
BytesPerPixel = 4L ;long
Type = 3
END
```

```
3: BEGIN
BytesPerPixel = 4L ;Float
Type = 4
END
```

```
4: BEGIN
BytesPerPixel = 2L ;16 bit
Type = 1
END
```

```
5: BEGIN
BytesPerPixel = 4L ;24 bit
Type = 1
END
```

```
6: BEGIN
BytesPerPixel = 2L ; 16 bit unsigned
Type = 1
```

```

END

EndCase

Image= Make_Array( Hdr.Width* Hdr.Height*BytesPerPixel*Hdr.nFrames,Type = Type)

;open the file
OpenR,U,File,/get_lun, /Binary

;advance the file pointer past the header info
Point_LUN, U, 2120

;read the image
ReadU,u, Image

; clean the image formatting
If Hdr.nFrames EQ 1 Then Begin

Case 1 of
  BytesPerPixel EQ 1 : Image = Reform(Image,Hdr.Width,Hdr.Height)
  BytesPerPixel GT 1 : Image = Reform(Image, BytesPerPixel,Hdr.Width,Hdr.Height)
EndCase

EndIf Else Image = Reform(Image, Hdr.Width, Hdr.Height,Hdr.nFrames)

;check for alpha
If Not(KeyWord_Set(Alpha)) Then $
  If BytesPerPixel GT 1 Then Image = Image[1:3,*,*]

Free_LUN,u

Return, Image
End

;+
; NAME:
; WRITE_IPLAB
;
; PURPOSE:
; This procedure saves an image and header in the IPLAB image format.
; See reference for details.
;
; CALLING SEQUENCE:
; WRITE_IPLAB, Image, Hdr, File
;
; CATEGORY:
; IPLAB
;

```

```

; ARGUMENTS:
; IMAGE The image data.
; HDR The header structure associated with the image. (See reference and READ_IPLAB and
; QUERY_IPLAB routines.)
; FILE The fully qualified filename for the image. If the file already exists, the original
; is overwritten without warning.
;
; KEYWORDS:
; None
;
; MODIFICATION HISTORY:
; Written by Ben Tupper, Summer 2000.
; Bigelow Laboratory for Ocean Science
; btupper@bigelow.org
;-

```

PRO WRITE_IPLAB, Image, HDR, File

On_Error, 2

On_IOerror, trouble

```

If N_elements(File) EQ 0 Then Begin
  Message, 'File name must be present'
  Return
EndIf

```

```

If N_elements(HDR) EQ 0 OR Size(HDR, /TName) NE 'STRUCT' Then Begin
  Message, 'HDR must be a structure.'
  Return
EndIf

```

```

If N_elements(Image) EQ 0 Then Begin
  Message, 'Image must be present'
  Return
EndIf

```

OpenW, U, File, /Binary, /NoAutoMode , /Get_LUN

```

WRITEU, U, HDR.Version ;version
WRITEU, U, HDR.IPLABFORMAT ;File Format 0 for Data
WRITEU, U, HDR.DataType ;dat type
WRITEU, U, HDR.WIDTH
WRITEU, U, HDR.HEIGHT
WRITEU, U, HDR.Reserved1 ;reserved
WRITEU, U, HDR.OVERLAYINFILE
WRITEU, U, HDR.Reserved2 ;reserved

```

WRITEU, U, HDR.NFRAMES
WRITEU, U, HDR.Reserved3 ;reserved
WRITEU, U, HDR.NRegMarks
WRITEU, U, HDR.RESERVED4
WRITEU, U, HDR.CLUT

WRITEU, U, IMAGE

TROUBLE:

FREE_LUN, U

END

File Attachments

- 1) [query_iplab.pro](#), downloaded 80 times
 - 2) [read_iplab.pro](#), downloaded 86 times
 - 3) [write_iplab.pro](#), downloaded 121 times
-