Subject: Re: Assoc and byteorder keywords
Posted by Liam E. Gumley on Thu, 28 Aug 2003 14:43:18 GMT
View Forum Message <> Reply to Message

"Richard G. French" <rfrench@wellesley.edu> wrote in message
news:BB7363AB.C9D%rfrench@wellesley.edu...
> I have several large binary data sets (image cubes) that I access from
> multiple platforms, using IDL. Typically, I do something like:
>
> Openr,lun,/GET_LUN,'images.bin'
> Images=assoc(lun,fltarr(nx,ny))
>
> I might then want to do something like this:
>
> Norm=(images[5]-images[4])/(images[3]>threshold)
>
> Unfortunately, from some platforms, the images are not in the correct byte
> order. Because byteorder is a procedure and not a function, I am forced
into
> locutions such as:
>
> im5=images[5]
> Byteorder,im5, /SWAP_IF_BIG_ENDIAN
>
> im4=images[4]
> Byteorder,im4, /SWAP_IF_BIG_ENDIAN
>
> im3=images[3]
> Byteorder,im3, /SWAP_IF_BIG_ENDIAN
>
> Nomr=(im5-im4)/(im3>threshold)
>
> This seems to be negating the syntactical efficiency of the ASSOC
function.
> I could wrap all of this into a function, of course, but it seems to me
that
> an easier way would be for ASSOC to have the capability of doing the
> byte-swapping on the fly, by having the same keywords as BYTEORDER. So,
for
> example, I would like to see:
>
>
> Images=ASSOC(lun,fltarr(nx,ny),/SWAP_IF_BIG_ENDIAN)
>
> If RSI could just then call byteorder internally to the ASSOC function
> before delivering the data to the user, it would result in much cleaner
code
> at my end. Does anyone see a way to do something like this already, or see

> any problems with this suggestion?

Why not try this?

openr, lun, /get_lun, 'images.bin', /swap_if_big_endian

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Assoc and byteorder keywords
Posted by Richard French on Thu, 28 Aug 2003 15:44:05 GMT

View Forum Message <> Reply to Message

On 8/28/03 7:45 AM, in article BB7363AB.C9D%rfrench@wellesley.edu, "Richard G. French" <rfrench@wellesley.edu> wrote:

> I have several large binary data sets (image cubes) that I access from
> multiple platforms, using IDL. Typically, I do something like:
>
> Openr,lun,/GET_LUN,'images.bin'
> Images=assoc(lun,fltarr(nx,ny))
>
> I might then want to do something like this:
>
> Norm=(images[5]-images[4])/(images[3]>threshold)
>
> Unfortunately, from some platforms, the images are not in the correct byte
> order. Because byteorder is a procedure and not a function, I am forced into
> locutions such as:
>
> im5=images[5]
> Byteorder,im5, /SWAP_IF_BIG_ENDIAN
>
> im4=images[4]
> Byteorder,im4, /SWAP_IF_BIG_ENDIAN
>
> im3=images[3]
> Byteorder,im3, /SWAP_IF_BIG_ENDIAN
>
> Nomr=(im5-im4)/(im3>threshold)
>
> This seems to be negating the syntactical efficiency of the ASSOC function.
> I could wrap all of this into a function, of course, but it seems to me that
> an easier way would be for ASSOC to have the capability of doing the
> byte-swapping on the fly, by having the same keywords as BYTEORDER. So, for

> example, I would like to see:
>
>
> Images=ASSOC(lun,fltarr(nx,ny),/SWAP_IF_BIG_ENDIAN)
>
> If RSI could just then call byteorder internally to the ASSOC function
> before delivering the data to the user, it would result in much cleaner code
> at my end. Does anyone see a way to do something like this already, or see
> any problems with this suggestion?
>
> Dick French
>

The fine folks at RSI wrote to me directly to let me know that there is such
a keyword to the OPENR command, which does exactly what I am after. Not sure
how I missed this one - probably was introduced a few versions ago and I did
not pay attention to the change. Thought others might want to know.

Dick French

Subject: Re: Assoc and byteorder keywords
Posted by Richard French on Thu, 28 Aug 2003 23:25:58 GMT

View Forum Message <> Reply to Message

>
> Why not try this?
>
> openr, lun, /get_lun, 'images.bin', /swap_if_big_endian
>
> Cheers,
> Liam.
> Practical IDL Programming
> http://www.gumley.com/
>
>

Well, I had hoped that that would work, but for my floating point images, I
seem to need to use the /LSWAP option of BYTEORDER in order to get things to
work - just doing /SWAP_IF_BIG_ENDIAN fails to work. To be specific, I want
the data to be in 'native' format on my Dec Alpha (oops - Compaq -ooopps,
HP), but to be readable/writable from a Sun workstation. If, on the Sun, I
do:

```
Read,1,array
Byteorder,/LSWAP,/SWAP_IF_BIG_ENDIAN,array
```

Then everything works

Dick