Subject: BSQ to BIL
Posted by jtmcahill on Fri, 05 Feb 2010 19:46:43 GMT
View Forum Message <> Reply to Message

Hi,

I'm trying to convert a BSQ to BIL and then make an ISIS .cub file using some of the ENVI Doit functions and a write_isis command. Converting to BIL appears to work when I run the program below (meaning it is identified as a BIL when I open it in ENVI) but I'm having trouble making an ISIS .cub file out of the product. I realize my mistake may be in ISIS but I want to make sure I've converted it to BIL correctly first.

Second, the write_isis command isn't being recognized by idl.

I'd appreciate any help with either topic.

```
pro bsq2isis,filename

envi, /restore_base_save_files
envi_batch_init, log_file='batch.txt'
envi_open_file, filename, r_fid=fid
envi_file_query, fid, dims=dims, nb=nb
  pos  = lindgen(nb)
  out_name = out_name
envi_doit, 'convert_doit', $
    fid=fid, pos=pos, dims=dims, $
    o_interleave=1, out_name=filename + '_bil.img', $
    r_fid=r_fid
a=isis_write, r_fid,filename + '_bil.cub'

end
```

Subject: Re: BSQ to BIL
Posted by David Fanning on Mon, 08 Feb 2010 17:48:04 GMT
View Forum Message <> Reply to Message

Hawaiianite writes:

> Anyway, I'm more concerned with converting the file to BIL format
> first. I can worry about writing the .cub file later (which I've done
> before I'm just a bit rusty, obviously). What I don't understand is
> the above code (with the isis_write command commented out) makes a
> file that ENVI says is a BIL when opened and provides a small header
> file with basic information. How can verify that this is true

If ENVI is happy with it, I'm sure it's true. Here is an article that points to another that shows how I would convert the interleaving of files in IDL. If you did something like that, you are probably OK.

http://www.dfanning.com/ip_tips/where3.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: BSQ to BIL
Posted by Maxwell Peck on Mon, 08 Feb 2010 20:10:41 GMT
View Forum Message <> Reply to Message

> me. I've tried verifying this by loading it into ISIS with a RAW2ISIS
> procedure but it gives me a pretty messed up image so either it isn't
> writing out the header info correctly for me to enter into ISIS or it
> hasn't made the conversion at all. It looks beautiful in ENVI and

The easiest way to check the conversion is to load the converted file into ENVI (the image before the ISIS_WRITE). If the interleaving in the header is BIL and the image looks 'right' when you open it then it almost certainly is right, the code itself looks fine. If you want to convince yourself I guess you could edit the header and force it to BSQ then try and display it. ENVI uses the header to decide how to load and display the image, it's not just a "info" file so the interleaving has to be correct.

If the ISIS writing commands are IDL commands they almost certainly wont accept the fid keyword properly. It's an easy trap to fall in to when doing IDL/ENVI programming. It's possible it wont even error as fid is just a long integer scalar as far as IDL is concerned.

Max

Subject: Re: BSQ to BIL

Posted by jtmcahill on Mon, 08 Feb 2010 20:54:45 GMT
View Forum Message <> Reply to Message

On Feb 8, 3:10 pm, Maxwell Peck <maxjp...@gmail.com> wrote:
>> me. I've tried verifying this by loading it into ISIS with a RAW2ISIS
>> procedure but it gives me a pretty messed up image so either it isn't
>> writing out the header info correctly for me to enter into ISIS or it
>> hasn't made the conversion at all. It looks beautiful in ENVI and
>
> The easiest way to check the conversion is to load the converted file
> into ENVI (the image before the ISIS_WRITE). If the interleaving in
> the header is BIL and the image looks 'right' when you open it then it
> almost certainly is right, the code itself looks fine. If you want to
> convince yourself I guess you could edit the header and force it to
> BSQ then try and display it. ENVI uses the header to decide how to
> load and display the image, it's not just a "info" file so the
> interleaving has to be correct.
>
> If the ISIS writing commands are IDL commands they almost certainly
> wont accept the fid keyword properly. It's an easy trap to fall in to
> when doing IDL/ENVI programming. It's possible it wont even error as
> fid is just a long integer scalar as far as IDL is concerned.
>
> Max

Ok, thanks for the perspective. I'll look more into loading it into
ISIS appropriately now and get back to you.

Thanks,
Josh

Subject: Re: BSQ to BIL
Posted by jtmcahill on Mon, 08 Feb 2010 20:59:27 GMT
View Forum Message <> Reply to Message

On Feb 8, 3:53 pm, pp <pp.pente...@gmail.com> wrote:
> On Feb 6, 7:00 pm, Maxwell Peck <maxjp...@gmail.com> wrote:
>
>>> a=isis_write, r_fid,filename + '_bil.cub'
>
>> Hi,
>
>> I'm not sure where ISIS_WRITE comes from (it's not an ENVI function as
>> far as i know), or what your actual error is, but it is unlikely it
>> will accept the r_fid keyword. You will probably need to use the
>> ENVI_GET_DATA function to put the band in an IDL variable and output
>> it from that. If there is a lot of data you may need to use tiling

>> (all in the ENVI Programmers guide).
>
>> Max
>
> This is not directly related to the original question, but if anyone
> is interested, I have a reader and writer for ISIS cubes. The main
> reasons I wrote them are the awkward interface of the ISIS routines,
> their lack of ability to do more complicated things than just reading
> or writing core bands, and to not need to install ISIS just to read
> and write cubes from IDL. However, my routines presently only support
> BSQ.

pp,

I'd be very interested to see what you have. I was under the
impression ISIS files had to be in BIL format. Are you saying they
don't necessarily need to?

Josh

Subject: Re: BSQ to BIL
Posted by penteado on Fri, 12 Feb 2010 04:13:17 GMT
View Forum Message <> Reply to Message

On Feb 8, 6:59 pm, Hawaiianite <jtmcah...@gmail.com> wrote:
> I'd be very interested to see what you have. I was under the
> impression ISIS files had to be in BIL format. Are you saying they
> don't necessarily need to?
>
> Josh

The class to read, edit and write cubes is called pp_editablecube. The
routines you need to use it are documented at:

http://www.ppenteado.net/idl/pp_editablecube__define.html
http://www.ppenteado.net/idl/pp_readcube__define.html
http://www.ppenteado.net/idl/pp_setcubeheadervalue.html
http://www.ppenteado.net/idl/pp_extractfields.html
http://www.ppenteado.net/idl/pp_getcubeheadervalue.html
http://www.ppenteado.net/idl/pp_buffered_vector__define.html

and the .pro files are at the same location, with the corresponding
file names.

The documentation on the ISIS cube format is scarce, so those are
partly based on examining the contents of Cassini VIMS cubes and of
ISIS' source code. It may be the case that some changes may be

necessary to handle other cubes. Let me know if you have problems.

Subject: Re: BSQ to BIL
Posted by jtmcahill on Mon, 15 Feb 2010 16:42:21 GMT
View Forum Message <> Reply to Message

On Feb 11, 11:13 pm, pp <pp.pente...@gmail.com> wrote:
> On Feb 8, 6:59 pm, Hawaiianite <jtmcah...@gmail.com> wrote:
>
>> I'd be very interested to see what you have. I was under the
>> impression ISIS files had to be in BIL format. Are you saying they
>> don't necessarily need to?
>
>> Josh
>
> The class to read, edit and write cubes is called pp_editablecube. The
> routines you need to use it are documented at:
>
> http://www.ppenteado.net/idl/pp_editablecube__define.htmlhtt
p://www.ppenteado.net/idl/pp_readcube__define.htmlhttp://www
.ppenteado.net/idl/pp_setcubeheadervalue.htmlhttp://www.ppen
teado.net/idl/pp_extractfields.htmlhttp://www.ppenteado.net/
idl/pp_getcubeheadervalue.htmlhttp://www.ppenteado.net/idl/p p_buffered_vector__define.html
>
> and the .pro files are at the same location, with the corresponding
> file names.
>
> The documentation on the ISIS cube format is scarce, so those are
> partly based on examining the contents of Cassini VIMS cubes and of
> ISIS' source code. It may be the case that some changes may be
> necessary to handle other cubes. Let me know if you have problems.

I'm having trouble getting your code to run. Why are the .pro files
labeled "_define.pro"? Do I need to remove that and put them in the
IDL library to run. Because right now it isn't recognizing them and
keeps giving me a syntax error. IDL appears to not like the "::"
before init before I try to run pp_readcube.pro.

Consider me a novice. I find I've been using an older version of IDL
for some time and am not familiar with a lot of what you are trying to
offer.

Subject: Re: BSQ to BIL
Posted by penteado on Mon, 15 Feb 2010 17:04:04 GMT
View Forum Message <> Reply to Message

On Feb 15, 2:42 pm, Hawaiianite <jtmcah...@gmail.com> wrote:
> I'm having trouble getting your code to run. Why are the .pro files
> labeled "_define.pro"? Do I need to remove that and put them in the
> IDL library to run. Because right now it isn't recognizing them and
> keeps giving me a syntax error. IDL appears to not like the "::"
> before init before I try to run pp_readcube.pro.

No, do not change the file names. Those with __define in their name
are files that define classes. The routines with :: in their names are
methods of those classes, and that is the way they should be.

First, make sure you have in your IDL path all the source code files:

http://www.ppenteado.net/idl/pp_editablecube__define.pro
http://www.ppenteado.net/idl/pp_readcube__define.pro
http://www.ppenteado.net/idl/pp_setcubeheadervalue.pro
http://www.ppenteado.net/idl/pp_extractfields.pro
http://www.ppenteado.net/idl/pp_getcubeheadervalue.pro
http://www.ppenteado.net/idl/pp_buffered_vector__define.pro

Then, what exactly were you trying to do?

This (copied from the file pp_editablecube__define.html) is one
example of how to use it to read, edit and write a cube:

To initialize from the cube CM_1553510065_1_ir.cub:

a=obj_new('pp_editablecube',file='CM_1553510065_1_ir.cub')

To add a dummy backplane:

a-
> getproperty,backplanes=back,backnames=bnames,lines=lines,sam ples=samples
backplanes=[[[backplanes]],[[findgen(lines,samples)]]]
backnames=[backnames,'DUMMY']
a->setproperty,backplanes=back,backnames=bnames

To remove the first core band:

a->getproperty,core=core,wavelengths=wavs
core=core[*,*,1:*] & wavs=wavs[1:*]
a->setproperty,core=core,wavelengths=wavs

To add lines to the history part of the header:

app=['GROUP = testedit','date = '+strcompress(systime(),/
remove),'END_GROUP = testdate']
a->headerset,append=app

To write the edited cube to 'testedit.cub':
a->write,'testedit.cub'

Destroy the object when done with it:

obj_destroy,a

There are more examples, for reading thing from a cube, in the file
pp_readcube__define.html.

Subject: Re: BSQ to BIL
Posted by David Fanning on Mon, 15 Feb 2010 17:21:40 GMT
View Forum Message <> Reply to Message

pp writes:

> No, do not change the file names. Those with __define in their name
> are files that define classes. The routines with :: in their names are
> methods of those classes, and that is the way they should be.

Huh!? Are these anything like widgets? Don't use 'em. ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: BSQ to BIL
Posted by penteado on Mon, 15 Feb 2010 19:01:10 GMT
View Forum Message <> Reply to Message

On Feb 15, 3:21 pm, David Fanning <n...@dfanning.com> wrote:
>
> Huh!? Are these anything like widgets? Don't use 'em. ;-)

No, there are not widgets there. Only one the pp_editablecube class,
that provides an API for reading, editing and writing a cube. Which

seemed the most proper choice for editing, so that the setproperty
methods take care of keeping the cube valid when one thing is changed.

Subject: Re: BSQ to BIL
Posted by David Fanning on Mon, 15 Feb 2010 19:11:34 GMT
View Forum Message <> Reply to Message

pp writes:

> No, there are not widgets there. Only one the pp_editablecube class,
> that provides an API for reading, editing and writing a cube. Which
> seemed the most proper choice for editing, so that the setproperty
> methods take care of keeping the cube valid when one thing is changed.

Sorry. This was a joke. I think I went through the same thing
back in 1987 or whenever widgets were first introduced. It's
just damn hard to move astronomers(or scientists, generally)
off the dime. This is why most of my objects have procedural
interfaces. I don't have the strength to fight the good
fight a second time. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: BSQ to BIL
Posted by penteado on Mon, 15 Feb 2010 21:13:30 GMT
View Forum Message <> Reply to Message

On Feb 15, 5:11 pm, David Fanning <n...@dfanning.com> wrote:
> Sorry. This was a joke. I think I went through the same thing
> back in 1987 or whenever widgets were first introduced. It's
> just damn hard to move astronomers(or scientists, generally)
> off the dime. This is why most of my objects have procedural
> interfaces. I don't have the strength to fight the good
> fight a second time. :-)

I have noticed that difficulty, too. Even at much more basic things than widgets, such as good programming practices. It seems to me that there is a tendency to keep things frozen into naive, impractical, programming, because that is the way that each generation is thought to do it, and passes it on, unaltered, to the next. Just because they were never thought any better, and made to think that is all that there is to programming.

That particular class happened to be designed that way primarily to be used by much more complicated applications, some with a compound widget that examines a cube's contents (written using Catalyst), others that use it to process and store many cubes into a sav file, in a way that they can be easily used later, even allowing to recreate the ISIS cube files.
