
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,samples=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(systemtime(),/  
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.
