Subject: what to make of read_interfile.pro ?
Posted by Mike[2] on Wed, 29 Oct 2008 14:17:49 GMT
View Forum Message <> Reply to Message

New IDL "idiom"?

Interfile is an aged but quite straight forward image file format used in nuclear medicine. Turns out that there are some vestiges of it deep in the bowels of a certain manufacturer's reconstruction codes, so I decided it was time to come up with an interfile reader for IDL. I was pleased to see that there is a read_interfile routine that has been a part of IDL for many versions (since 1993). It doesn't work on the data I'm trying to read - in their wisdom, the people who wrote the code that wrote the data I've got called floats "float", rather than using the interfile specs which call for "short float" and "long float". I though it would be no problem to modify read_interfile to handle this, but then I became astonished by a new IDL idiom - new to me at least.

In the code, there is a structure that handles image headers. Each header has widget_base associated with it and the uvalue for that that base is used to store the header data value. There is another widget_base uvalue that is used to handle valid choices. This is in an entirely non-gui procedure by the way. At the end of the procedure, there is a loop that "widget_control, /destroy"s the widget_bases.

Anyone have an idea of why the author might have made that choice? Were there no pointers in 1993? Maybe the original author (Goldstein) just wanted to toy with us here in the future?

Anyway, out of respect/fear/queasiness, I think I may let this procedure rest in peace and write a header cleaner/fixer to stick in between reading the header and interpreting it.

Mike

Subject: Re: what to make of read_interfile.pro ?
Posted by Mike[2] on Fri, 31 Oct 2008 14:12:51 GMT
View Forum Message <> Reply to Message

Thanks for the history lesson gang. It is amazing how quickly I forget why we used to do the things we did. Of course 15 years may not be all that quick, but time does go faster as time goes on.

Robbie - Nice technique of packing results into keywords. This is where I'd use a python dictionary. My IDL version is an object with

two arrays: one containing keys and the other values. My get_value method looks for the value that matches the key that is passed in, pretty much like a python dictionary.

Actually, I don't have arrays in my object - they are pointers to arrays, which seemed a little bit easier for my mind to handle than uvalues ;-)

Happy Halloween!

Mike
