

---

Subject: Re: what to make of read\_interfile.pro ?

Posted by [David Fanning](#) on Wed, 29 Oct 2008 14:37:01 GMT

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

---

Mike writes:

- > 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.

Ah, fond memories... :-)

Yes, before pointers there were "handles" in IDL, and before handles, there were user values of unrealized top-level base widgets. I can't remember now if I got this idea from Josh, or he got it from me. I probably got it from him, because he was an amazing programmer. But I know I was the one who started advertising it. And I remember David Stern coming to me one time after reading about "Using Pointers in IDL" on one of my IDL course brochures to find out just what the hell I was teaching people out there. The rest is history. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

---

Subject: Re: what to make of read\_interfile.pro ?

Posted by [Paul Van Delst\[1\]](#) on Wed, 29 Oct 2008 14:40:02 GMT

Mike wrote:

- > 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?

Pointers were introduced in v5 - and back in '93 I think something like v3.6 might have been current. But, didn't v3 and such have those things called "handles" that were sorta like pointers? Anyway....

- > 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.

I think respect is warranted - the author came up with a clever solution when no other methodology was available. But, regardless, read thee some books/articles on unit testing and refactoring techniques and you should be able to fix the code without any vestiges of fear and/or queasiness -- assuming, of course, your boss doesn't mind (or know :o)

cheers,

paulv

---

Subject: Re: what to make of read\_interfile.pro ?  
Posted by [Robbie](#) on Thu, 30 Oct 2008 00:41:34 GMT

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

---

I did come up with an incredibly simplistic way to read a "flat" interfile header.

I just use keywords to represent each entry.

; Example:

```
; interfile_readkey, '2001cntrate_13.s.hdr', STUDY_DATE_DD_MM_YYR=a
pro interfile_readkey, filename, _REF_EXTRA=ex
openr, lun, filename, /GET_LUN
line = ""
while (~ EOF(lun)) do begin
    readf, lun, line
```

```

if ((line ne "") && (STRMID(Line,0,1) ne ';')) then begin
  if (STRMID(Line,0,1) eq '!') then line = STRMID(Line,1)
  if (STRMID(Line,0,1) eq '%') then line = STRMID(Line,
1)
  KeyEnd = STRPOS(Line, ":=")
  if (KeyEnd ge 0) then begin
    count = 0
    Kwd = strjoin(strsplit(strupcase(strtrim(stRMID(Line,
0,KeyEnd),2)), '[]',=(/:'./EXTRACT),'_')
;    print, kwd
    if (n_elements(ex) gt 0) then $
      inds = where(kwd eq ex,count)
    if (count gt 0) then $
      (SCOPE_VARFETCH(kwd, /REF_EXTRA)) = strtrim(stRMID(Line,
KeyEnd+2),2)
    endif
  endif
endif
endwhile
free_lun, lun
end

```

---