Subject: Re: A routine to annotate PS files
Posted by David Fanning on Sat, 09 Jan 2010 15:30:24 GMT
View Forum Message <> Reply to Message

Ed Hyer writes:

> So here is a routine you can use to personalize your PS files and make
> them easier to search. The most obvious application is to attach the
> IDL calling stack in the postscript file, so that you can determine
> what script exactly made what file and when. However, this procedure
> is extensible, and I suspect people will find more uses for it.

This is an interesting idea, and ties in quite nicely with the
notion of provenance, which everyone in the data business is
paying a lot of attention to these days.

But doesn't this add an extra, blank, page of output to
every PostScript file? You can't just open the file up in
GhostView and see the results. You have to go to the 2nd
page. This takes some of the shine off it for me. :-(

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: A routine to annotate PS files
Posted by MarioIncandenza on Sat, 09 Jan 2010 16:12:22 GMT
View Forum Message <> Reply to Message

> But doesn't this add an extra, blank, page of output to
> every PostScript file? You can't just open the file up in
> GhostView and see the results. You have to go to the 2nd
> page. This takes some of the shine off it for me. :-(

When I tested it, it did not do that. All it's adding are comment
lines to the PostScript, and it's smart enough not to add them before
the first line (not in front of it right now, don't remember exactly
where it does add them). Maybe it depends on when you call it? I am
calling it immediately before DEVICE,/CLOSE.

Subject: Re: A routine to annotate PS files
Posted by David Fanning on Sat, 09 Jan 2010 16:46:46 GMT
View Forum Message <> Reply to Message

Ed Hyer writes:

> When I tested it, it did not do that. All it's adding are comment
> lines to the PostScript, and it's smart enough not to add them before
> the first line (not in front of it right now, don't remember exactly
> where it does add them). Maybe it depends on when you call it? I am
> calling it immediately before DEVICE,/CLOSE.

OK, when I tried it *after* the plot, the lines got
inserted just before the last character in the PostScript
stream that describes the plot. It didn't appear to
do any harm there, but it seems a weird place for it
to go.

How do you use this information? Do you have routines
that can examine your PostScript files to retrieve
this information?

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: A routine to annotate PS files
Posted by R.G.Stockwell on Sat, 09 Jan 2010 22:44:47 GMT
View Forum Message <> Reply to Message

"Ed Hyer" <ejhyer@gmail.com> wrote in message
news:62a37f74-b522-4f96-a7cb-cec1761c65a5@m3g2000yqf.googleg roups.com...
> Happy New Year Everybody!
>
> I asked about this last year, and someone pointed me to the DEVICE,/
> OUTPUT and SCOPE_TRACEBACK commands, which turned out to hit the
> spot.
>
> So here is a routine you can use to personalize your PS files and make

> them easier to search. The most obvious application is to attach the
> IDL calling stack in the postscript file, so that you can determine
> what script exactly made what file and when. However, this procedure
> is extensible, and I suspect people will find more uses for it.

Thanks a lot Ed. I can see me using this routine a lot, to help me keep track of publication ready figures. I'll put in the info of what journal etc, and include the name of the idl program that created the file (which makes for easy editing). Very nice.

cheers,
bob

Subject: Re: A routine to annotate PS files
Posted by MarioIncandenza on Sun, 10 Jan 2010 17:57:19 GMT
View Forum Message <> Reply to Message

On Jan 9, 8:46 am, David Fanning <n...@dfanning.com> wrote:
> How do you use this information? Do you have routines
> that can examine your PostScript files to retrieve
> this information?

Those I don't have to write. There are several, but my main tool is

find . -name '*ps -exec grep 'String' {} \;

I can't say what would work for Windows people...

--Edward H.

Subject: Re: A routine to annotate PS files
Posted by DavidPS on Thu, 14 Jan 2010 22:50:08 GMT
View Forum Message <> Reply to Message

On Jan 9, 12:44 am, Ed Hyer <ejh...@gmail.com> wrote:
> Happy New Year Everybody!
>
> I asked about this last year, and someone pointed me to the DEVICE,/
> OUTPUT and SCOPE_TRACEBACK commands, which turned out to hit the
> spot.
>
> So here is a routine you can use to personalize your PS files and make
> them easier to search. The most obvious application is to attach the
> IDL calling stack in the postscript file, so that you can determine

> what script exactly made what file and when. However, this procedure
> is extensible, and I suspect people will find more uses for it.
>

Thanks a lot Ed! and Happy New year to you too! I was thinking time ago about this... One of the info I would like to have was the data used and/or some parameters used when plotting. In that way if there is someone who wants to compare his data with mine he could do it just from the PS file (and there's a lot of paper with their eps files available). Now I know it is possible, I'd been told that grace does that but I never tried.

Cheers,

David

Subject: Re: A routine to annotate PS files
Posted by MarioIncandenza on Wed, 24 Feb 2010 19:04:44 GMT
View Forum Message <> Reply to Message

Here is an updated version of the routine. Changes include:
- Now writes entire call stack, including $MAIN$ ($MAIN$ has a program file associated with it, and often that is information you need, it's now included);
- Includes a /QUIET option to suppress echoing of the tags to be written;
- Includes a (Unix-only, or maybe even more limited) option to /PARSE_FINGER, which calls 'finger `whoami`' and writes the 'Login:' and 'Name:' tags into the PS file. If it doesn't work on your system, just ignore it. It won't cause any problems.

```
 ------------------------------------------------------------ -
pro
 annotate_psfile,extra_tags=extra_tags,parse_finger=parse_fin ger,quiet=quiet
tagf='("%% ",a," : ",a)'        ; %% Tag : Value
trace=scope_traceback(/structure)
ntrace=n_elements(trace)
geni=ntrace-2                   ; index to proximate calling script
tag0=string('GenRoutine',trace[geni].routine,format=tagf)
tag1=string('GenFile',trace[geni].filename,format=tagf)
tags=[tag0,tag1]
if(geni gt 0) then begin
   for icall=0,geni-1 do begin
      tag0=string('CallRoutine',trace[icall].routine,format=tagf)
      tag1=string('CallFile',trace[icall].filename,format=tagf)
      tags=[tags,tag0,tag1]
   endfor
```

```
endif
 timecode=string(julday(),format='(C(CYI4.4,CMOI2.2,CDI2.2,CH I2.2,CMI2.2))')
tag0=string('TimeCode: ',timecode,format=tagf)
tags=[tags,tag0]
if n_elements(extra_tags) ne 0 then begin
    extranames=tag_names(extra_tags)
    for iextra=0l,n_elements(extranames)-1 do begin
        tag0=string(extranames[iextra],extra_tags.
(iextra),format=tagf)
        tags=[tags,tag0]
    endfor
endif
if n_elements(parse_finger) ne 0 then begin
    spawn,'finger `whoami`',result,error
    if(error eq '') then begin
         fingertags=stregex(result[0],'Login:(.*)Name:(.*)',/subexpr, /
extract)
        tag0=string("Login: ",strtrim(fingertags[1],2),format=tagf)
        tag1=string("Name: ",strtrim(fingertags[2],2),format=tagf)
        tags=[tags,tag0,tag1]
    endif
endif

if n_elements(quiet) eq 0 then print,tags,format='(A)'
ntags=n_elements(tags)
for itag=0l,ntags-1 do device,output=tags[itag]
return
end
```