

---

Subject: Re: Add SCOPE\_VARFETCH output, other tags into PostScript files  
Posted by [Michael Galloy](#) on Thu, 19 Nov 2009 21:19:04 GMT

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

---

Ed Hyer wrote:

> Hello IDL wizards,  
>  
> What I am imagining is an IDL routine I can call at the end of any  
> routine that generates a PostScript output, that will add comment tags  
> to the .eps file specifying some information that I can then 'grep'  
> for.  
>  
> What I can't quite figure is how exactly to plant these tags in the  
> PostScript file. When I look at the actual .eps files generated by  
> IDL, they have this at the top:  
>  
> %!PS-Adobe-3.0 EPSF-3.0  
> %%BoundingBox: 0 0 767 767  
> %%Title: Graphics produced by IDL  
> %%For: hyer@cal, /users/hyer/idlpro/flambe  
> %%Creator: IDL Version 7.0.8 (linux x86\_64 m64)  
> %%CreationDate: Tue Nov 10 11:55:12 2009  
>  
> What's the best way to:  
>  
> 1) Replace that 'title' tag with one of my own choosing;  
> 2) Add tags: 'GenScript:', 'CallingScripts:', 'YYYYMMDDHHMM' <- (a  
> more parseable date);  
>  
> Suggestions, please. I'll post the script I come up with to the  
> thread-- I'm really looking forward to being able to look at a PS file  
> and immediately locate the script that created it, and to grep for  
> graphs created with a particular script!  
>  
> (P.S. I am a Unix-only IDL guy, but if you can think of a cross-  
> platform solution, I'd be happy to code it up. If you think of a Unix-  
> only solution, though, don't be shy!)  
>  
> Thanks in advance,  
>  
> --Edward H.  
>  
> P.S. I was hoping that Imagemagick 'convert -comment' would be the  
> answer, but it doesn't work (I do 'convert test.ps -comment "comment" -  
> test2.ps', then look for 'comment' in test2.ps', and get nothing).

Check out the OUTPUT keyword to the DEVICE routine. It will directly  
write a string to a PostScript file. This won't change the title, but I

think it should allow you to add other tags. Will that work?

Mike

--

www.michaelgalloy.com  
Research Mathematician  
Tech-X Corporation

---

---

Subject: Re: Add SCOPE\_VARFETCH output, other tags into PostScript files  
Posted by [Maarten\[1\]](#) on Fri, 20 Nov 2009 10:28:34 GMT

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

---

On Nov 19, 9:19 pm, Ed Hyer <ejh...@gmail.com> wrote:

> What I am imagining is an IDL routine I can call at the end of any  
> routine that generates a PostScript output, that will add comment tags  
> to the .eps file specifying some information that I can then 'grep'  
> for.  
>  
> What I can't quite figure is how exactly to plant these tags in the  
> PostScript file. When I look at the actual .eps files generated by  
> IDL, they have this at the top:  
>  
> %!PS-Adobe-3.0 EPSF-3.0  
> %%BoundingBox: 0 0 767 767  
> %%Title: Graphics produced by IDL  
> %%For: hyer@cal, /users/hyer/idlpro/flambe  
> %%Creator: IDL Version 7.0.8 (linux x86\_64 m64)  
> %%CreationDate: Tue Nov 10 11:55:12 2009  
>  
> What's the best way to:  
>  
> 1) Replace that 'title' tag with one of my own choosing;

Unix only, use sed:

```
pro set_title_on_idl_eps_output, graph, title, view=view, pdf=pdf  
  compile_opt defint32, strictarr, strictarrsubs
```

```
  if !version.OS_FAMILY eq "unix" && n_elements(graph)*n_elements  
(title) gt 0 then begin
```

```
    ;; remove backslashes and dollars (TeX math escapes)  
    str = strjoin(strsplit(title, "$", /extract), "")  
    str = strjoin(strsplit(str, "\", /extract), " ")  
    str = strjoin(strsplit(str, "'", /extract), "'")
```

```
    ;; escape slashes
```

```

str = strjoin(strsplit(str, "/", /extract), "V")

;; temporary file name
ID = getenv("IDL_PROC_ID")
if strlen(ID) eq 0 then begin
  spawn, "echo $$", out, err
  ID = getenv("LOGNAME") + "." + out
endif
tmp_file = strcompress("/tmp/SET_TITLE_ON_IDL_EPS_OUTPUT."+ID
+ ".eps", /remove_all)

command = "sed 's/^%%Title: .*$/%%Title: " + str + "/" < "" +
graph + $
"" > " + tmp_file + " && " + $
"mv -f " + tmp_file + " "" + graph + """

spawn, command

if keyword_set(pdf) then begin
  command = "epstopdf " + graph
  spawn, command
endif

if keyword_set(view) then begin
  if !version.OS_NAME ne 'Mac OS X' then $
    spawn, "gv "" + graph + "" & " $
  else $
    spawn, "open "" + graph + ""
  endif
endif
end

> 2) Add tags: 'GenScript:', 'CallingScripts:', 'YYYYMMDDHHMM' <- (a
> more parseable date);

```

See suggestion by Mike (output keyword for device routine).

Maarten

---