
Subject: Re: Suppress print statement from nested procedures

Posted by [lecacheux.alain](#) on Fri, 15 Nov 2013 09:50:10 GMT

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

Le vendredi 15 novembre 2013 00:24:37 UTC+1, Sam a écrit :

> Hello all,

>

>

>

> I built a wrapper for a procedure (it calls many others, in turn), but the output that was important for the original program is no long important. In fact it's quite a nuisance. I'd rather not dig down into the subroutines to add the option to turn off printing -- this would be a major effort. Is there an IDL equivalent of redirecting to /dev/null (or a text file) when calling a procedure from within another procedure? In a perfect world, it would be:

>

>

>

> pro wrapper

>

> [snip]

>

> while i lt 1000 do begin

>

> [snip]

>

> main_procedure >& /dev/null

>

> [snip]

>

> i++

>

> endwhile

>

> [snip]

>

> end

>

>

>

>

>

> Thanks!

>

>

>

> Sam

I see two possible solutions to suppress PRINT output.

1) PRINT,text is equivalent to PRINTF,-1,text. It means that logical unit -1 is used for stdout by IDL (you can confirm that by looking at FSTAT(-1) output). Then, you might redirect the logical unit -1 to some disk file. The drawback is that there is no way (afaik) to go back again to normal stdout behavior.

2) you could run your program in a detached IDL session created from your main session (see the IDL_IDLbridge class). In this case your PRINT messages will be discarded or, if you want them, stored in some file (OUTPUT keyword). You will have to manage the way in which you can eventually transfer your program results from the detached session to the main IDL session.

By the way, for any informational messaging in IDL, it is better to use MESSAGE,/INFO rather than PRINT. In particular, you can use the !QUIET system variable to control the output at the main level.

alx.
