
Subject: Re: Call Tree Generator

Posted by [m218003](#) on Mon, 16 Aug 1999 07:00:00 GMT

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

In article <MPG.121bc33671cb20f1989887@news.frii.com>,

davidf@dfanning.com (David Fanning) writes:

> Calvin King (cking@sandia.gov) writes:

>

>> Is there a program that looks at an IDL procedure/function and generates

>> the call tree? I inherited a large number of IDL programs for a project.

>> I am going through the programs one at a time starting at the

>> top-on-down to make sure I have all of the necessary programs before

>> compilation and use. Just to make sure I am explaining myself clearly,

>> the top level program A calls programs B and C; B calls D, E, and F; and

>> C calls G, H, I, and J, and so on... Is there an automated way to make

>> sure I have all of the programs: A, B, C, D, E, F, G, H, I and so on?

>>

>> Thanks, in advance, for any pointers!

>

> Start a fresh IDL session. Compile your top-level program.

> Now type `Resolve_All`. If you get error messages, you don't

> have all the programs. :-)

>

> You will know which ones you need: the ones `Resolve_All`

> can't find.

>

A good method, but not perfect. It doesn't find any routine called with `EXECUTE`, `CALL_PROCEDURE`, or `CALL_FUNCTION`. But it should be possible to write a program that does the following:

- * `resolve_all`

- * get all compiled routines with `-- was it HELP,... ?`

- there is also some output of the filenames for these routines

- * loop through all files and look for `EXECUTE`, `CALL_...` statements

- and analyze their argument. If the argument is a simple

- string, you can assume there should be a file named `<string>.pro`

- if the argument is some sort of variable, it's harder. Maybe

- it's enough to simply print out a list of all `EXECUTE` and

- `CALL_...` statements. Of course you can do that with `grep`

- (one of the many many reasons why I will never switch from Unix to Windows - although I am told it exists for Windows which is

- to say for DOS)

just \$0.02, or 0.0212 Euro ;-)

Martin

> Cheers,

>

> David

>
