
Subject: Re: Where is my function called
Posted by [David Fanning](#) on Mon, 25 Aug 2008 12:31:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wox writes:

> Does anyone know how I can find out where a specific
> function/procedure is called in a project. I'm having build order
> problems: it's a mess. My idea was to use ROUTINE_INFO to get the
> names and files of the (user) routines in my project (2000 routines in
> 35 files) and then try to visualize the connections so I can order
> things.

Good luck! I think it would be easier to just name your
files correctly. :-)

<http://www.dfanning.com/tips/namefiles.html>

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: Where is my function called
Posted by [Juggernaut](#) on Mon, 25 Aug 2008 16:23:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Aug 25, 8:31 am, David Fanning <n...@dfanning.com> wrote:

> Wox writes:
>> Does anyone know how I can find out where a specific
>> function/procedure is called in a project. I'm having build order
>> problems: it's a mess. My idea was to use ROUTINE_INFO to get the
>> names and files of the (user) routines in my project (2000 routines in
>> 35 files) and then try to visualize the connections so I can order
>> things.
>
> Good luck! I think it would be easier to just name your
> files correctly. :-)
>
> <http://www.dfanning.com/tips/namefiles.html>
>
> 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.")

No idea if these may work for you but the `scope_traceback()` function seems to have some capability....whether or not it completely fits your needs I don't know.

Subject: Re: Where is my function called
Posted by [David Fanning](#) on Mon, 25 Aug 2008 17:05:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Bennett writes:

> No idea if these may work for you but the `scope_traceback()` function
> seems to have some capability....whether or not it completely fits
> your needs I don't know.

Now that I think about it, this problem probably does result in the build order of the project. I've given up on building projects a long time ago, which is why the problem didn't ring a bell with me at first.

Rather, I start a fresh IDL session. Run my program. Then do a `RESOLVE_ALL`. I *do* have to provide a list of all the objects my project is going to need using this method, because `RESOLVE_ALL` usually doesn't do any such thing. But, the advantage of the method is that with well-named programs, I don't have to constantly fight build order problems, which are **ESPECIALLY** pernicious when you are trying to get someone else to compile your damn programs. A customer, for example.

I usually provide a "make" program to "build" the project for my customer. It's pretty simple, as I say. The only tricky part is including all the objects the program needs in the `CLASS` keyword to `RESOLVE_ALL`:

```
;; MakeProjectScript
.RESET_FULL_SESSION
myprojectProgram
RESOLVE_ALL, CLASS=['FSC_PSCONFIG', 'PROGRESSBAR']
Save, /ALL, FILE='myprojectprogram.sav'
END
```

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: Where is my function called

Posted by [Wox](#) on Tue, 26 Aug 2008 08:46:07 GMT

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

On Mon, 25 Aug 2008 11:05:08 -0600, David Fanning <news@dfanning.com> wrote:

<snip>

```
> I usually provide a "make" program to "build" the project
> for my customer. It's pretty simple, as I say. The only
> tricky part is including all the objects the program
> needs in the CLASS keyword to RESOLVE_ALL:
>
> ;; MakeProjectScript
> .RESET_FULL_SESSION
> myprojectProgram
> RESOLVE_ALL, CLASS=['FSC_PSCONFIG', 'PROGRESSBAR']
> Save, /ALL, FILE='myprojectprogram.sav'
> END
>
> Cheers,
>
> David
```

What I was doing for now was having a start.pro file:

```
-----
@filex
@filey
...
@myprojectProgram

pro start
myprojectProgram
end
-----
```

And then make a batchfile compile.pro:

```
-----  
.RESET_FULL_SESSION  
.Compile start.pro  
Resolve_All,class=['Trackball',...]  
-----
```

"IDL> @compile" would compile the project and "IDL> start" would run it. The files filex,filey,... contain many routines. The crucial point is ordering the @filex,@filey,...(35 files for now) in the start.pro file and deviding the ~2000 routines I have over a manageable amount of files (since each file would need a line in start.pro).

I was reading your naming rules. I put too many routines in one file. In order to resolve this problem without looking at the routines manually, I was looking for something like the scope_traceback function (thx for the suggestion) but for this I have to call all the routines and procedures manually and for each build a calling tree (with scope_traceback) and afterwards merge all this calling trees. Is there a way of automating this?
