Subject: compiling many routines at once
Posted by kostis on Fri, 01 Jun 2007 13:12:36 GMT
View Forum Message <> Reply to Message

I use a main program main.pro that includes many other programs and
functions.
Each time i recompile amin.pro i have to compile all included routines
seperately...
Functions are compiled automatically with the command FORWARD_FUNCTION
but that doesnt work with routines...
How can i compile everything at once with .r main.pro ?

Subject: Re: compiling many routines at once
Posted by Maarten[1] on Fri, 01 Jun 2007 13:58:26 GMT
View Forum Message <> Reply to Message

On Jun 1, 1:12 pm, kostis <kostis...@gmail.com> wrote:

> I use a main program main.pro that includes many other programs and
> functions.
> Each time i recompile amin.pro i have to compile all included routines
> seperately...
> Functions are compiled automatically with the command FORWARD_FUNCTION
> but that doesnt work with routines...
> How can i compile everything at once with .r main.pro ?

make sure your program files have the correct name, and are stored in
a location where IDL can find them. See http://www.dfanning.com/tips/namefiles.html

Maarten

Subject: Re: compiling many routines at once
Posted by Jo Klein on Fri, 01 Jun 2007 14:06:51 GMT
View Forum Message <> Reply to Message

Also, you could create a batch file with .comp statements, and then use
@mybatchfile to run it.
Jo

Subject: Re: compiling many routines at once
Posted by KRDean on Fri, 01 Jun 2007 15:11:23 GMT
View Forum Message <> Reply to Message

Use IDL's Projects!

For years I would remove the the Project Window on the left side of IDLDE by unchecking the Preference -> Project. My style of programming was to place all the PRO and FUNCTION routines in the same file, making sure FUNCTION routines were at the top. Or use the Preference -
> Path to insert the path of my code.

However, recently, I started inheriting IDL code in my department that was written by other people that have left or are still around. I needed a way to manage all these snippets of code. I learn to use the IDLDE Project. Whenever, I start a new IDL application or given new code, I'll create an IDL Project.

IDL Project allow you to add and remove IDL files, have quick access to the routines if changes are require, search through all the files (helpful for debugging), and compiles all the routines with a click of the button.

All in all, IDL projects has made me an efficient application developer.

Kelly Dean
Fort Collins, CO

On Jun 1, 7:12 am, kostis <kostis...@gmail.com> wrote:

> I use a main program main.pro that includes many other programs and
> functions.
> Each time i recompile amin.pro i have to compile all included routines
> seperately...
> Functions are compiled automatically with the command FORWARD_FUNCTION
> but that doesnt work with routines...
> How can i compile everything at once with .r main.pro ?

Subject: Re: compiling many routines at once
Posted by Vince Hradil on Fri, 01 Jun 2007 15:19:44 GMT

View Forum Message <> Reply to Message

On Jun 1, 8:12 am, kostis <kostis...@gmail.com> wrote:

> I use a main program main.pro that includes many other programs and
> functions.
> Each time i recompile amin.pro i have to compile all included routines
> seperately...
> Functions are compiled automatically with the command FORWARD_FUNCTION
> but that doesnt work with routines...
> How can i compile everything at once with .r main.pro ?

resolve_all?

Subject: Re: compiling many routines at once
Posted by Robbie on Fri, 01 Jun 2007 22:42:34 GMT

View Forum Message <> Reply to Message

The way that you compile in IDL is ultimately about personal
preference and coding style. Things like 'resolve_all' will pick up
most things, but will miss objects and functions called using execute/
call_procedure/call_function

You can simply run your program and pray that you covered every line
of code. Resolve all and then save, /ROUTINE.

A project is toublesome if you are importing a large amount of
external code. I have had difficulties when I wanted to 'merge' or
update an external project. Particularly when there are more than
just .pro files.

My personal favourite is to compile all .pro files within a given
pathspec. After all, this is the same method that IDL uses to locate
routines. Why not use it to compile them as well? I prefer this method
because I can incorporate it into my WIX (Windows Installer XML) build
scripts.

Robbie

Example usage:

```
nmacomplile, 'itools.sav',['lib/itools','lib/igamma.pro'], ROOT_DIR=!
DIR

pro nmacompile, target, source, ROOT_DIR=root_dir,
ITRESOLVE=itresolve, NO_SAVE=no_save

if (keyword_set(itresolve)) then ITRESOLVE

ThisRoutine = Routine_Info('nmacompile', /Source)

if (n_elements(root_dir) eq 0) then $

 root_dir = FILE_DIRNAME(FILE_DIRNAME(FILE_DIRNAME(ThisRoutine.path)))

for i=0,n_elements(source)-1 do begin

 path = FILEPATH(source[i],ROOT_DIR=root_dir)

 if (file_test(path,/DIRECTORY)) then $

  programs = FILE_BASENAME(FILE_SEARCH(path, "*.pro"),'.pro') $
 else $
  programs = FILE_BASENAME(path,'.pro')
 help, CALLS=calls
 for j=0,n_elements(calls)-1 do calls[j] = (strsplit(calls[j], ' ',/
EXTRACT))[0]
 for j=0,n_elements(programs) -1 do begin
  inds = where(strupcase(programs[j]) eq calls,count)
  if (count gt 0) then continue

  catch, errorvalue
  if (errorvalue ne 0) then begin
   catch, /CANCEL
  endif else begin
   RESOLVE_ROUTINE, programs[j], /EITHER
  endelse
  catch, /cancel
 endfor
endfor
RESOLVE_ALL, /CONTINUE_ON_ERROR
catch, /cancel
if (~ keyword_set(no_save)) then $
 save, FILENAME=target, /ROUTINES
end
```