Subject: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by davidf on Tue, 04 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz (mgs@io.harvard.edu) makes some excellent
points when he writes:

> let me just add a little note on seperate files vs.
> "libraries", because I don't think this is mentioned
> in the IDL manuals: It seems as if the automatic
> compilation of IDL always compiles up to point where
> it finds the procedure or function with the same name
> as the filename.

Martin is mostly right about this, although the information
is found in the manuals. (I will admit that its reference
is a bit obscure for such an important point, however.)

I would like to elaborate on this a moment because you simply
would not believe the number of people I run into who
do not understand how programs get compiled. Many of these
people try to run IDL in the compile-link-run fashion of
the good ol' Fortran days. :-)

If you use the IDL executive command .Compile to compile
a file, all the program modules in the file get compiled.
(The executive command .Run can also be used this way,
although it confuses new IDL users because nothing actually
gets *run*.) To run the compiled programs, just use their
names on the IDL command line or in an IDL program.

But one of the wonderful features of IDL is the ability
to have your programs compiled *automatically* when they
are needed. This makes it possible to write IDL code that
is not cluttered up with a bunch of INCLUDE statements, etc.
You simply put all of your program files in some directory
that is on the IDL path and you don't have to worry whether
the files are pre-compiled or not when you write the next
IDL program that uses one of these commands.

But the key word here is "command". When you write a program
you are, in effect, creating another IDL command. The key rule
for getting your programs to compile automatically is to make
sure that the "command" module is the *last* program module
in a file that has the *same name* as the command you are
trying to build (with a ".pro" extension, of course).

For example, suppose I were trying to write an IDL program

or command named VIEW_IT. What I am trying to do is have
something happen when I type the name VIEW_IT at the IDL
command line or put that name in an IDL program. But VIEW_IT
might be a huge program with lots of subroutines, etc. I might
have routines called MAKE_BIG_WINDOW and GET_MY_DATA, etc, etc.
that I need to make my VIEW_IT program run.

What I should do is put the MAKE_BIG_WINDOW and GET_MY_DATA and
all the rest of the VIEW_IT utility routines in my file *in
front of* the VIEW_IT program module. In fact, VIEW_IT should
be the *last* module in the file. And I should name the file
"view_it.pro".

Now, when I type "view_it" at the IDL command line, IDL looks
for a file named "veiw_it.pro" (all lowercase letters on UNIX
file systems, no matter *how* you type it on the IDL command
line!). It compiles the program modules in that file *until*
it gets to the program module with the same name as the
file (which, don't you know, is the *last* program module :-),
at which point it stops compiling and *runs* that program
module automatically.

Now, it sometimes happens that a utility routine like
MAKE_BIG_WINDOW becomes a whole lot more useful to you
than it was before. In fact, you may want to use it
in a new program you are building called SEE_IT. If that
is the case, then take MAKE_BIG_WINDOW out of the VIEW_IT
code and put it in its own file called "make_big_window.pro".
It has now become another IDL command.

Automatic compilation doesn't solve *all* the problems
you might run into. (Hence, the Forward_Function command.)
But in my experience it solves a lot of them. And it sure
beats having to compile all of your program modules before
you can do anything in IDL. :-)

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by Martin Schultz on Thu, 06 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

This is a multi-part message in MIME format.

--------------446B794B15FB
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

David Fanning wrote:
>
> Martin Schultz (mgs@io.harvard.edu) writes:
>
> [...] It would
>> certainly be useful to have a tool which would look for all
>> the routines that may be called from a "command" (i.e. pro or function
>> identical to filename), and lists the files in which they are found. Of
>> course, this does depend on your installation (order of searchable
>> libraries). I guess it would come down to a real or pseudo compilation
>> and could possibly be achieved by
>> some tricky use of the journal output ??? Has anyone written something
>> like this ?
>
> It would be useful to have a tool like this.
[...]

--

Thanks for this tip (why can't they put a link to RESOLVE_ROUTINE in the
.COMPILE section of the manual ???

With the help of the RESOLVE_... routines and the ROUTINE_INFO function
I put together a small tool that does gather all the files one needs for
a program distribution (attached below). The drawback is that this list
also contains the routines that are only needed in order to find the
routines that are needed AND everything that had been compiled before
(e.g. the startup file). If it does not happen too frequently, one can
delete these files manually, but it would be nice if there were a better
way to restrict the output to only the routines associated with the one
in question.

Martin.

-- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
186 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
IDL-homepage: http://www-as.harvard.edu/people/staff/mgs/idl/
 ---------- -------

--------------446B794B15FB
Content-Type: text/plain; charset=us-ascii; name="distribute.pro"
Content-Transfer-Encoding: 7bit
Content-Disposition: inline; filename="distribute.pro"

```
pro distribute,routinename

; first compile the routine of interest

    resolve_routine,routinename

; and all the routines called therein

    resolve_all

; then obtain information on all the routines and functions
; that are currently compiled

    r1 = routine_info(/source)
    r2 = routine_info(/source,/functions)

; seperate path and name information

    path = [ r1.path, r2.path ]
    name = [ r1.name, r2.name ]


; get uniq path (i.e. single files)
    si = path(sort(path))
    upath = si(uniq(si))

; print out results (filenames and sorted routine names):
;   1. routines in local directory
    ind = where(strpos(upath,'/') lt 0)
    if (ind(0) ge 0) then $
      for i=0,n_elements(ind)-1 do print,upath(ind(i))
;   2. routines in directories that do not contain "lib" or "rsi"
    ind = where(strpos(upath,'lib') lt 0 AND strpos(upath,'rsi') lt 0 $
              AND strpos(upath,'/') ge 0)
```

```
    if (ind(0) ge 0) then $
      for i=0,n_elements(ind)-1 do print,upath(ind(i))
;   3. routines in directories that contain "lib" but not "rsi"
    ind = where(strpos(upath,'lib') ge 0 AND strpos(upath,'rsi') lt 0 $
              AND strpos(upath,'/') ge 0)
    if (ind(0) ge 0) then $
      for i=0,n_elements(ind)-1 do print,upath(ind(i))
;   4. routines in directories that contain "rsi"
    ind = where(strpos(upath,'rsi') ge 0 $
              AND strpos(upath,'/') ge 0)
    if (ind(0) ge 0) then $
      for i=0,n_elements(ind)-1 do print,upath(ind(i))

    si = sort(name)
    print,';'
    for i=0,n_elements(path)-1 do $
       print,'; ',name(si(i)),' : ',path(si(i))

return
end



----------446B794B15FB--
```

Subject: Re: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by davidf on Thu, 06 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz (mgs@io.harvard.edu) writes:

> With the help of the RESOLVE_... routines and the ROUTINE_INFO function
> I put together a small tool that does gather all the files one needs for
> a program distribution (attached below).

Nice program, Martin. Thanks. :-)

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by hegde on Thu, 06 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning (davidf@dfanning.com) wrote:

> so I could make a save file. I decided to construct a
> "make" file.
> It looked something like this:
> PRO COYOTE_MAKE
> Resolve_Routine, 'coyote1'
> ...
> Resolve_All
> Save, /Routines, File='coyote.sav'
> END

I do use something similar:

1) makefile is similar to above example except that I use .Run file1 ..... filen
2) 'idlmake' script (I use UNIX):
#! /bin/csh
source $IDL_DIR/bin/idl_setup
$IDL_DIR/bin/idl $1

3) 'idlmake coyote.pro' at shell prompt compiles everything.

4) To run the software from shell prompt, I use the script 'run':
#! /bin/csh
source $IDL_DIR/bin/idl_setup
$IDL_DIR/bin/idl $1

5) Now I put the lines
;
Restore, 'coyote.sav'
Call the first routine (in my case the main GUI).
exit
;
in file 'coyote' and type 'run coyote' from command line. To a user it
appears as a executable or a software running at shell prompt rather than IDL
command prompt. By setting and retreiving environment variables one can put
.sav or .pro files anywhere. Unless I am experimenting with a new command, I
don't even go to IDL prompt.

But the biggest headache I face is, my software is about 200+ (~ 15000 lines) IDL routines and
60+ C routines. Each time I add a new module, by the time
I compile in the above mentioned way, read the test data and run the new module
it will be 5-10 minutes. If it comes across a misspelled keyword,
I have to start all over again and I have seen this happen too many times.

What I would like to have atleast is a pre-processor/parser if not a *compiler*
(I guess RSI wants to make money selling Runtime IDL).

-M. Hegde

Subject: Re: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by Martin Schultz on Thu, 06 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan wrote:
>
> [..]
>
> But it would certainly be nice to have a few extra tools to manage
> large IDL software collections - e.g., checking for number of
> parameters,
> legality of keywords, etc..
>

Have you tried the ROUTINE_INFO function ? This has some keywords to
get the number of keywords or parameters and/or their names.

Martin.

---------- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
186 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
IDL-homepage: http://www-as.harvard.edu/people/staff/mgs/idl/
---------- -------
