Subject: Re: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by davidf on Wed, 05 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

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

> Hmmm... This sounds to me like you are fond of make
> files etc. I actually like IDL's way of automatic
> compilation, EXCEPT if one wants to distribute
> a program package and has to find out which files are needed. 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. But having
just gone through this exercise for a client, I'll tell you
what I did. I wanted to make a run-time application. The
application has 10-12 modules, each of which has calls
to any number of other library routines.

I've been using automatic compilation all through the
development cycle, but I needed something now that would
get all the library routines gathered up in one place
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_Routine, 'coyote2', /Is_Function
Resolve_Routine, 'coyote3'
...
...
Resolve_All
Save, /Routines, File='coyote.sav'
END
```

I just put all my program modules at the beginning to
make sure they are compiled (this part only took 4-5
iterations :-), the Resolve_All routine at the end
catches all the library routines I am using, and I
am set. Now, as I add modules to the application, I

just include the name in here and I can make a save
file whenever I need it by typing "coyote_make" at the
IDL command line. Pretty slick, I think.

> Another useful option of such a program would be
> to filter out dead or duplicate routines which linger
> around from old versions of a code.

Don't need it. Anytime I want to find bad code I just
explain it to a novice IDL user. I've found that
the prouder I am of my hot-shot code the more dumb
mistakes I find in it. :-)

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/