Subject: Re: Nice ways to compile
Posted by Ken Mankoff on Sun, 10 Apr 2005 15:45:28 GMT
View Forum Message <> Reply to Message

On Sun, 10 Apr 2005, Robert Barnett wrote:
> I'm looking for an easier way to compile my code into a .sav file.
Me too. I use the same process (roughly). The method seems to work,
but it also seems fragile.

> + Complile the package and depenancies to a save file and run in
> an IDL VM "sandbox" environment.

> I expect to be able to compile my IDL packages with a single UNIX
> script (step marked as a +). I also expect that my compilation
> depends solely on the contents of the filesystem rather than the
> current state of the IDL command prompt. This ensures that I can
> always return to a snapshot of my idl directory as it was when I
> compiled my code.

I think it is tricky to get it to rely on the contents of the
filesystem. There are known issues when it will not track down
dependencies that you or I would see reading the code (is this what
you mean by filesystem?). Hence your request for a compiler
directive.

> For example, I currently compile like the following:
I use IDL with other software that I build from emacs/CLI by calling
make on a standard *nix Makefile. So 1 command: "make". At some
point in the dependency chain I call

```
  idl make.pro
```

which is

```
; makefile for EVA
; produces eva.sav
.reset_session
.com my_map_set
.com eva
.com eva_earth__define
.com eva_ocean__define
.com eva_data__define
.com eva_image__define
;;; this part grows as my code library grows.
RESOLVE_ALL
spawn, 'echo $HOSTNAME', h
if h = 'random' then save, /routines, EMBEDDED=..., file='eva.sav' else $
  save, /routines, file='eva.sav' ; no license
```

```
$cp eva.sav ../IDL/bin/
exit
; is there any way to determine if I am a real interactive IDL
; session or a unix CLI "$ idl foo.pro" version? Because
; this exit should be conditional :)
```

> Perhaps I am looking at this problem completely wrong. I would
> expect there to be some way I can append a compiler directive
> which indicates that blah__define.pro requires
> 'idlgrlegend__define'. Compiler directives are not really in the
> IDL idiom, so I'm wondering what other choices I might have.

They are in the idiom. See ?COMPILE_OPT
But none does what you want that I know of.

I'm thinking there is some way (with a shell script and grep?) to
solve your dependencies automagically, until IDL does it correctly
itself.

-k.

--
http://spacebit.dyndns.org/