Subject: Re: file paths
Posted by Tim Patterson on Wed, 02 Apr 1997 08:00:00 GMT
View Forum Message <> Reply to Message

I have a similar problem. What I ended up doing was
to have subdirectories below my source code directory
and defining an environment variable $CASPER_SRC_DIR
to point at the top level, then set up a file
that would define my !path like so...

```
!path = !path+":"+$
  getenv('CASPER_SRC_DIR')+'modules/' + ':'+$
  getenv('CASPER_SRC_DIR')+'fov/' + ':'+$
  getenv('CASPER_SRC_DIR')+'kernels/' + ':'+$
  getenv('CASPER_SRC_DIR')+'plot/' + ':'+$
  getenv('CASPER_SRC_DIR')+'batch/' + ':'+$
  getenv('CASPER_SRC_DIR')+'utils/' + ':'+$
  getenv('CASPER_SRC_DIR')+'spice/' + ':'+$
[....]
  getenv('CASPER_SRC_DIR')+'target/' +':'+$
  getenv('CASPER_SRC_DIR')+'includes/'
```

Then I have a compilation routine, which executes this
at the very beginning and has a list of files to compile
as:

```
.size 65000 65000
@$CASPER_SRC_DIR/casper_path
.run casper_main
.run casper_makefiles
.run casper_widget
.run casper_toplevel
[...]
.run casper_final_routine
print, 'All compiled - starting Casper'
Casper_Main
```

and so on. I can move the files around as much as I want
as long as I make sure all the subdirectories are in my
path definition file and as I create new routines I
just tack them on the end of the compilation routine.
(Which also saves eveything as a precompiled binary, btw.
As I have over 200 routines it takes a fair while to
have to compile it ech time)

I tried a lot of fancier things, but this was the simplest

and easiest method I could come up with. Unfortunately
.run *.pro doesn't seem work which would have made my life
much easier, and you can't use EXECUTE to .RUN a file, and
then RESOLVE routines only work if you have one routine per
file.

If you find something better, let me know!

Tim

PS Yep, the tool is called Casper.

David wrote:
>
> hi,
>
> I want to add some code at the top of a fairly large project that will
> tell IDL where to look for all the subroutines it'll need. They are in
> various different folders, organized by their function. I want the code
> to compile everything automatically, even if I move the source around.
> (It would be messy to put all the subroutines in the same folder).
>
> Is there any way other than help, /source to show the path of the
> currently executing file? If I knew the path to the executing file, I
> could append it (with a +) to !path to make this work, right?
>
> Thanks in advance for any help!
>
> cheers!
> dave
>
> **
> David Katz
> Esquimalt Defence Research Detachment
> co-op student
>