Subject: Re: Need advice on building a project
Posted by Paul Van Delst[1] on Fri, 02 Jul 2004 14:29:52 GMT
View Forum Message <> Reply to Message

Robert Barnett wrote:
>
> For my first major project using IDL, I used version 5.4 and the idlde.
> I'm about to move onto a second project and I think that even idlde for
> version 6.0 is not terribly useful.
>
> I want to know if anyone has tried this before:
>
> Today I quickly wrote up a (perl) script that allows one to specify the
> build order in a heirarchical fashion.

Forgive my denseness, but what do you mean by "build order"? Are you saying that your setup has problems finding other dependent .pro files that aren't in your current directory? If so, then all you should need to do is stick your entire IDL code heirarchy into you IDL_PATH env variable.

Apologies if you already know this, but the way I do it is to put *all* my IDL code in an /idl subdirectory off my home directory. Then I do a

```
export IDL_PATH=${IDL_PATH}:+\${HOME}/idl
```

in my .bashrc and every single file in every subdirectory is "visible" to IDL when you compile your .pro file of the moment.

> Each directory contains a BuildOrder file which indicates what programs
> need to be built but also can contain dependancies on other BuildOrder
> files.
>
> The aim of this is to package my programs so that any given group of
> programs can be compiled and tested as indepenantly as possible. Also a
> package (including it's build order) can be imported and exported
> without too much trouble.

Now, the export issue is a different one (I reckon). I solve this using CVS. When I want to export some code (e.g. to distribute to users) I tag (cvs tag) all of the required files with a string (e.g., "Ootsma_Plotter_2-13" to indicate this is release 2.13 of my IDL application "oostma_plotter.pro") I associate with that application. Then you simply do something like

```
cvs export -r Ootsma_Plotter_2-13 -d./ ${CVS_REPOSITORY}
```

to extract all the required .pro files into the current directory. And you can do this anywhere since cvs works across networks.

The only thing that can be a pain in the rear is tagging all the required files in the first place. But if you do it as you write your app (just like the documentation :o) then it's no biggie. Still, a perl script to do this task would be handy.

I'm not sure if this addresses exactly what you asked about. At any rate, I recommend CVS to cut down on versionitis -- although I'm always amazed at the resistance people have to using it... It's quite peculiar. Especially when I see people with multiple copies of the same file in different directories, each one possibly subtlely modified in a way that may or not be noticed depending on which one is first in the IDL path search list.. Anyway...

paulv
